

The Analyst's Red Book

03/04/2021– Version: draft

*Deeper and broader insights
paradigm shifts and innovative ideas
to*

***EMPOWER the
BUSINESS INFORMATION
SYSTEMS ANALYST***

Exploitation of the Company's Information
Conception of the Corporate Information Systems
Developing Information Capabilities
Innovation through Information

Audience: **All types of Systems Analysts, ISE Analysts**

FOREWORD

The Analyst's Red Book (TARB) aims to strengthen and to boost the discipline of engineering information solutions in corporate environments.

In modern larger companies, the IT component plays a significant and critical role. Companies seek to automate and optimise their operations. Innovation is as important, not among the least, to stay ahead of the competition. IT has also the potential to transform companies and to drive business activities to achieve greater goals. But critical conditions are still far from ideal.

1) IT initiatives evolved from the development of small applications meant to be used by a limited number of users and having a specific function to the development of much larger tentacular systems in complex heterogeneous environments sharing and reusing information across the company and beyond. The IT discipline had to develop itself under pressure and in turbulent circumstances. The problems corporate IT encounters today are symptoms of a lack of thoughtfulness and maturity of the IT discipline.

2) The present underlying idea driving IT initiatives is to respond to business demand and to develop the software applications demanded by the business community. Responding to this demand and meeting their expectations should lead to beneficial results and customer's satisfaction. This way of thinking revolves around the demand, the business stakeholders, the developers, programming and technologies. This idea has been present since half of a century.

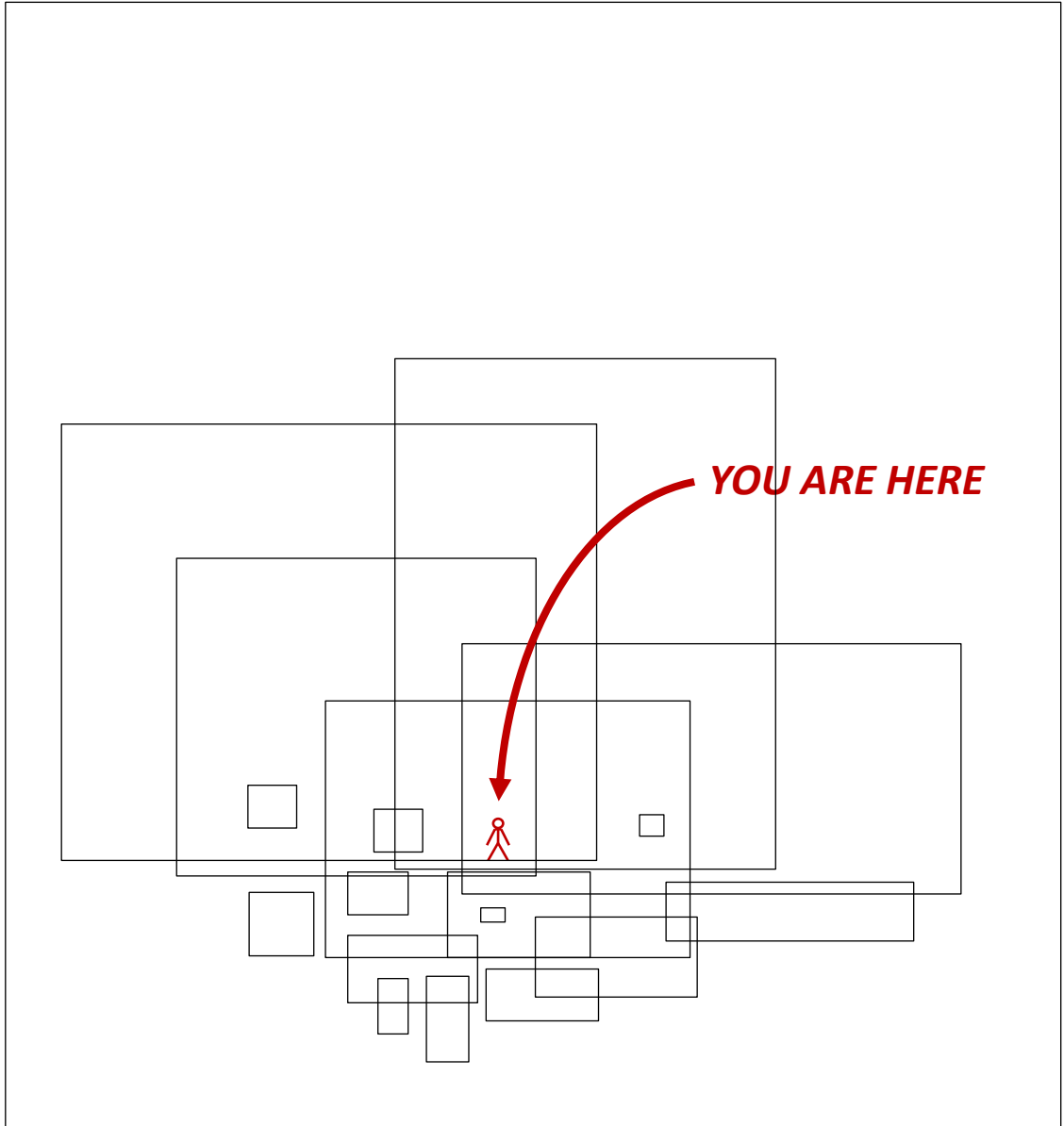
Lower goals lead only to lower results. Striving for higher goals leads to better and greater achievements. This requires a paradigm shift. A new belief system, a more solid thinking framework, a holistic view and approach and higher goals are essential.

There are two possible ways to evolve to another belief system. We can mature through repeating mistakes until, through crisis's, the right lessons are learned. This is a rather painful method and may take time. Or, we can evolve by thinking with an open, analytical and critical mind. TARB hopes to contribute to the latter.

It is hoped that TARB acts as a trigger and forms a basis for engaging in analysis, reflection and critical investigation to solve issues and obstacles and to renew and consolidate the discipline of Informatics, ISE, Systems Analysis, Business Analysis and other variants.

Axel Vanhooren

*Freelance Consultant Business Informatics
Belgium₂*



World of Systems

Special thanks go to Frédéric Naisse and Germelyn Javier for their valuable contribution.

Versions

LATEST VERSION @

<http://analystsredbook.net/Publications/The-Analysts-Red-Book--Axel-Vanhooren.pdf>

Older Versions

14/02/2020

10/01/2020

17/09/2019

15/04/2019

15/03/2019

15/02/2019

10/01/2019

18/12/2018

30/10/2018

Notes:

The dates are mentioned on the pages. The dates can be used to find the pages that are new or which have significantly changed. A search on the new date allows to find them easily.

Table of Contents

30/10/2018

1. [About the Analyst's Red Book](#)
2. [Corporate IT and Systems Analysis Today](#)
3. [Introduction](#)
4. [Foundation - Core Ideas](#)
5. [Fundamental Topics](#)
 1. [Systems](#)
 2. [Company](#)
 3. [Information](#)
 4. [Processes](#)
 5. [Problem Solving](#)
6. [Information Component](#)
7. [Business Knowledge and Expertise](#)
8. [History](#)
9. [General Subjects and Views](#)
10. [Business-IT Alignment](#)
11. [Information Exploitation](#)
12. [Enterprise Architecture](#)
13. [Information Systems](#)
14. [Engineering](#)
15. [Management](#)
16. [Work Environment](#)
17. [Organisation](#)
18. [Roles and Collaboration](#)
19. [The Analyst](#)
20. [Projects](#)
21. [Information Systems Development Framework](#)
22. [Analysis](#)
23. [Architecture](#)
24. [Design](#)
25. [Programming](#)
26. [Testing](#)
27. [Deployment](#)
28. [Techniques](#)



We live in a world of


- ORGANISATIONS
- STRUCTURES
- SYSTEMS
- PROCESSES

and in a world of INFORMATION

ranging from words, diagrams and pictures on paper or screen to thoughts and electrical nerve impulses.

War is a matter of **vital importance** to the state; a matter of life or death, the road either to survival or to ruin. Hence, it is imperative that it **be studied thoroughly**.

The Art of War, Sun Tzu



ABOUT THE ANALYST'S RED BOOK

“A wise man will make more opportunities than he finds.”

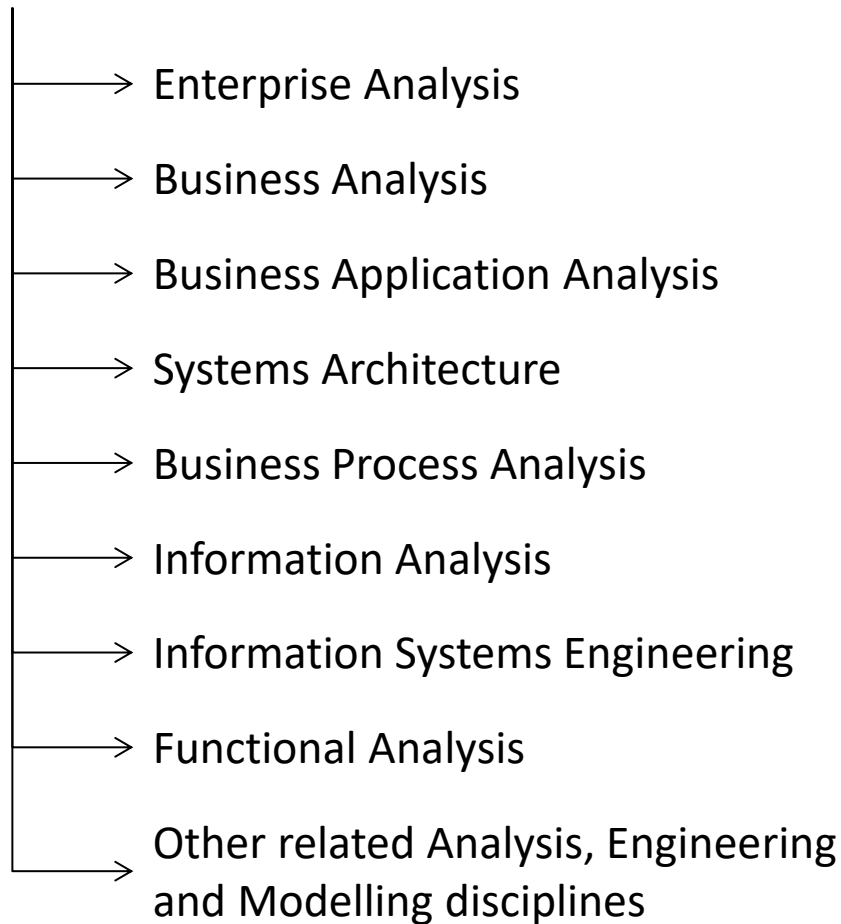
Francis Bacon

The ARB Document Organisation

18/12/2018

Introduction & Core Ideas about Systems Analysis		
Fundamental Knowledge Areas (Foundation and Building blocks)	Systems	Company / Organisation
	Process	Problem Solving
	Information	
Background Knowledge – Basic Topics		
Goal: The End in Mind	Information Exploitation	Information Systems
	Business-IT Alignment	Enterprise Architecture
Overall Organisation of the Informatics Engineering	Organisation	Work Environment
	The Analyst	Roles & Collaboration
	Projects	Engineering & Management
Informatics Initiatives (Projects, ...)	IS Development Framework	
	Activities	Techniques

Systems Analysis



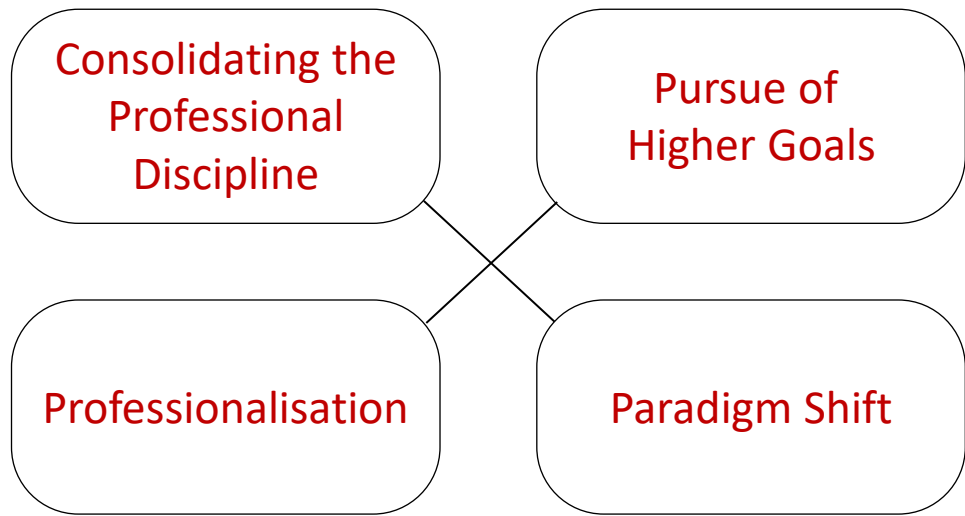
Notes:

In TARB, the term Systems Analysis is used in the most general and broader sense as an overarching discipline which includes all different levels and variants of analysis and design within (Business) Informatics.

Strictly spoken, the Business Analyst studies the business. Better titles can be “Business Information Systems Analyst” or “Corporate Information Systems Analyst”

The title “Business Analyst” also exists in financial world.

Systems Analysis also exists as a more specific discipline applied by the Systems Analyst inside projects. However, this is a more restrictive interpretation.



1. Consolidating the Professional Discipline

- Contributing to the development and progress of the discipline
- Strengthening its embedment into organisations

2. Pursuing Higher Goals

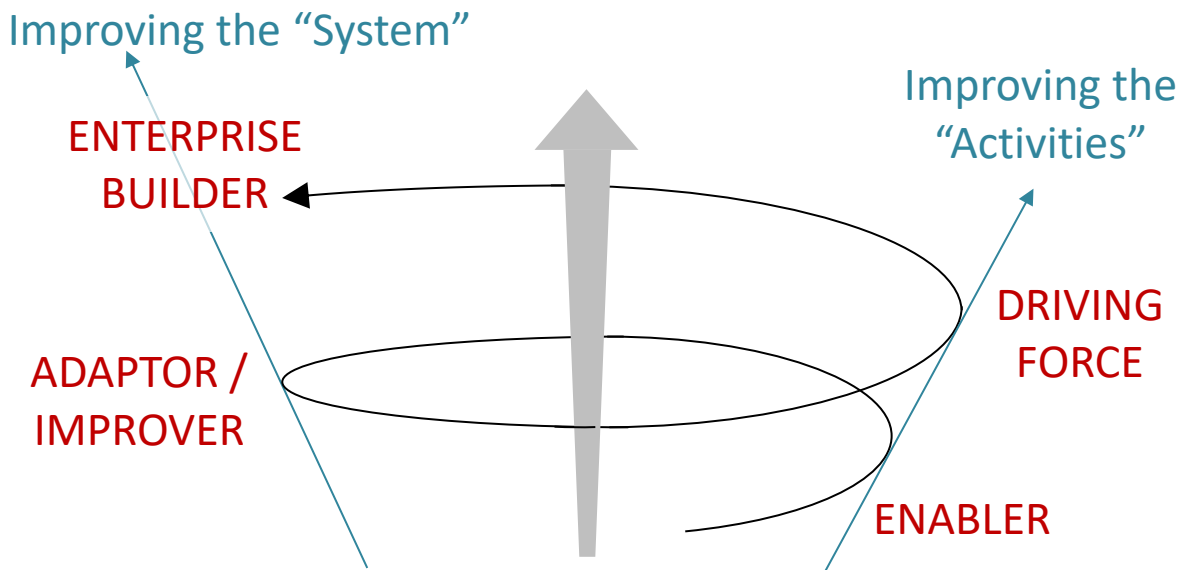
- Lower goals are easier. Lesser is learned. They require only basic competencies and produce limited results and smaller progress.
- Pursuing higher goals is more demanding, requires more learning, but allows greater progress and delivers better and greater results.

3. Professionalisation

- Sharing knowledge to reduce the gap between what is known of the discipline and what is (only) applied
- From HOW-TO to WHY (Deepening our knowledge)

4. Paradigm Shift

- A summary of shifts follows. These shifts are integrated in the content of TARB.



ENTERPRISE BUILDER

Engineering and building that part of the company that exploits information with the intention of supporting and strengthening the company and satisfying customer's information needs.

DRIVING FORCE

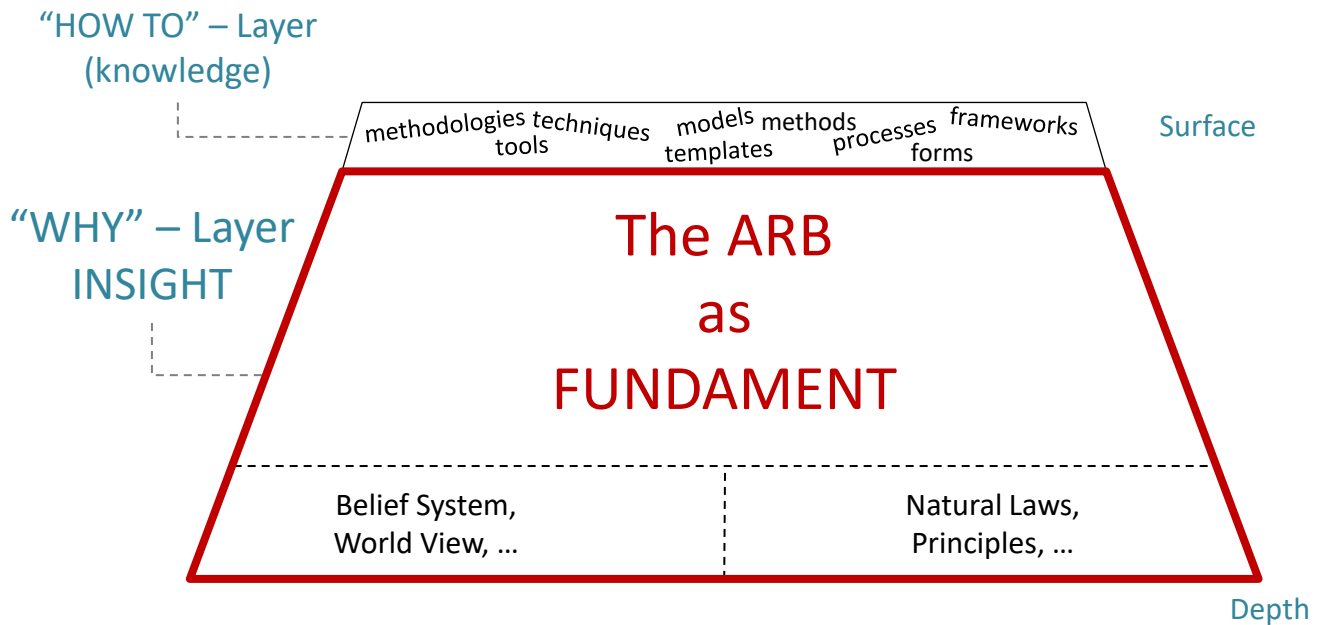
- Maximising resource utilisation
- Increasing production volumes and business results

ADAPTOR / IMPROVER

- Adapting to fit
- Reducing negative effects: limits, risks, costs
- Improving the system internally: controls, optimisation, structure, ...

ENABLER

- Putting technologies at work
- Enabling the business community to work better
- Automating traditional ways of working
- Introducing new technologies
- Modernisation
- Demand-driven software development



In Information Systems Analysis and Design,
no rule is absolute

Cooking by the book is a recipe for disaster



Understanding the FUNDAMENT is critical

Advantages:

- Ability to better identify and define initiatives correctly and to conceive better solutions
- Improved decision making
- Improved usage of frameworks, methodologies and methods and ability to adapt or conceive new ones suiting the initiatives.
- Ability to handle more complex projects and pursue higher goals

Proposes

- Principles
- Theories
- Concepts
- Relations
- Descriptions
- Solution Elements
- Building Blocks

and is

- adaptable
- improvable
- expandable

Reading the ARB:

- Understand how it fits in the overall picture of TARB
- Is the statement true?
- How important is it?
- What are the implications ?
- How can I use it ?

IMPORTANT NOTE:

The ARB does not take new domains like Artificial Intelligence, Brain–Computer interface (BCI) (Neural-Control Interface (NCI)) and similar new domains into account. Some statements have to be reinterpreted when dealing with these specific contexts.

Seven Key Values of the Analyst^{10/01/2020}



Truth Seeking & Understanding

The truth and insight is key for problem solving and building. Relentlessly, we try to discover the truth and gain understanding. Ignorance and assumptions are our enemy.

Building

This value is key to progress. It can be studying, learning, teaching, thinking, creating relations, optimising, improving, creating, sharing, or conceiving right solutions to be built.

Meaning & Purpose

Building meaningful solutions. Everything has a purpose. This must be respected. Only activities that are meaningful are performed. Do what makes sense. Meaningful solutions are based on necessity.

Passion & Perseverance

Do everything with passion. Be interested and motivated. Be kind. This is the driving force.

Respect

Respect for people, nature, reality, truth, principles, ...

Humility

The humility of being aware we may not know something, we might be wrong. Everybody might learn us something. Humility is a key quality to have an open mind.

Cooperation

Work together to achieve great things and to learn from each other. However, this does not rule out individual work and thinking.

These 7 values have **to be present all the time or most of the time**. Neither do they stand alone.
Example: Teach with passion and respect meaningful matter.

Smart Usage of TARB

18/12/2018

The ARB is not a methodology, process or structure.

The ARB is not a HOW-TO – Guide !!

It is not a solution that can be applied as is right away.

In general, solutions ready to apply, to be strictly followed are very unlikely in Systems Analysis.

Why? Every project, situation or problem is UNIQUE. There are no standard projects, standard situations or standard problems. So, there can not be a standard recipe.

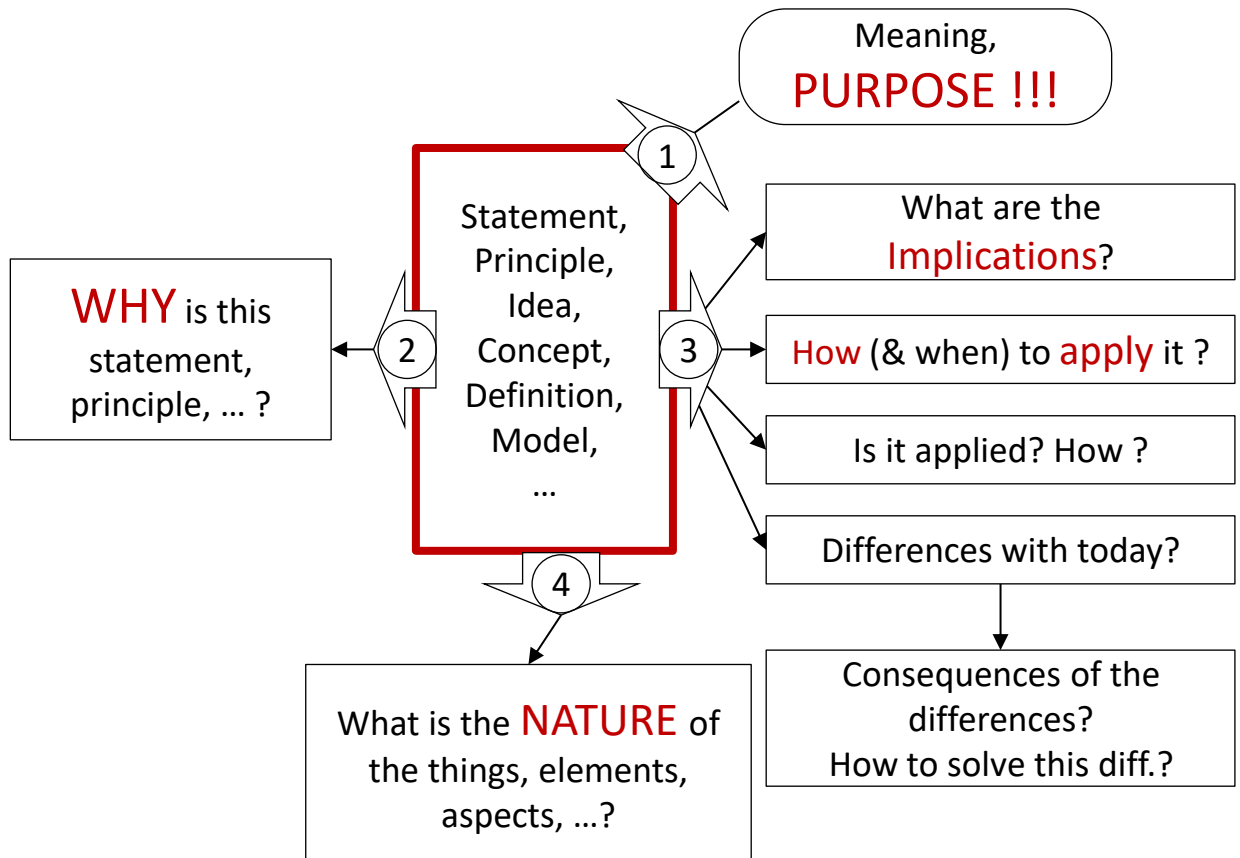
The Analyst has always to determine how to work, what preconditions must be met, what techniques to be used and how to adapt to the specific circumstances.

TARB (= The ARB)

- deepens the understanding
- organises knowledge
- provides patterns
- supports autonomous thinking
- incites further thinking and investigation

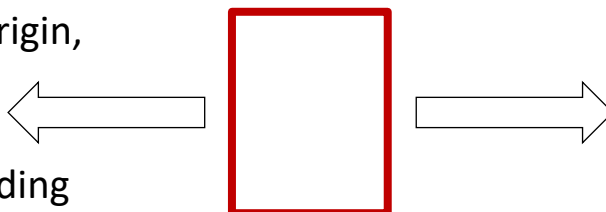
Uncritically following or applying TARB is the exact opposite of the intention of TARB

Key Questions:



Encouragement to always question
in these 4 directions

Towards origin,
deeper
understanding



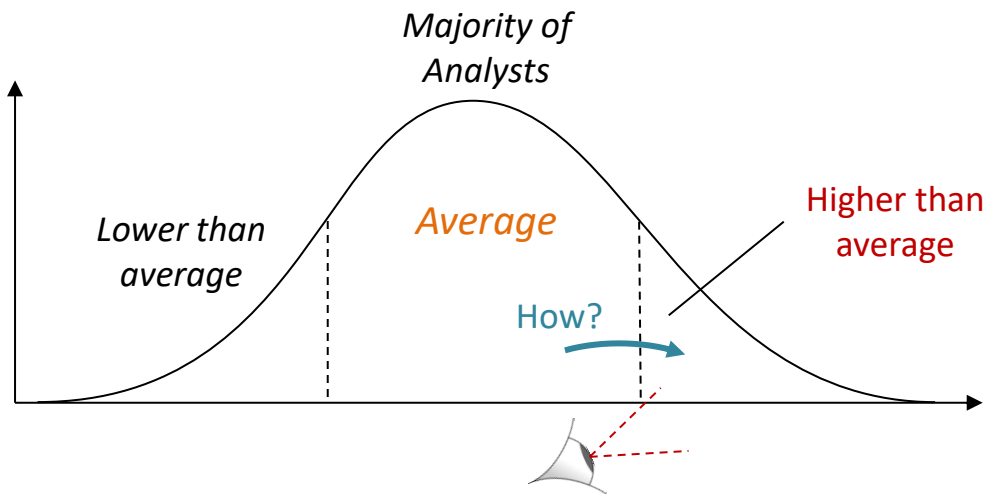
Towards integration
and improved
practical application

“Insanity is doing the same thing over and over again, but expecting different results.” (*)

Rita Mae Brown

Likewise

It is insane to do the same as everyone else and to think like everyone else while expecting superior results, results way above average.



Be careful by adopting the thinking and acting of the majority. The majority has, in the best case, an average understanding.

Follow the example of the top-level analysts.

Notes:

(*): quote is commonly misattributed to Einstein.

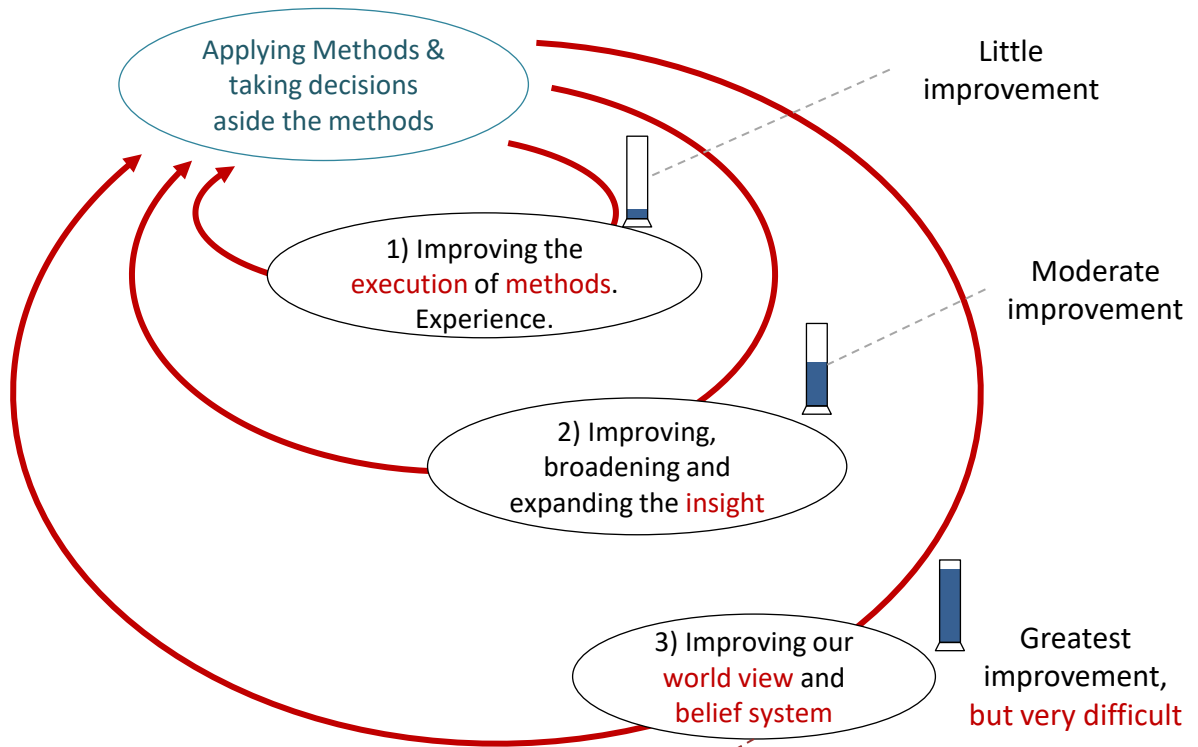
Set aside of intelligence or skills, what are the differences between lower than average analysts, average analysts and significantly better analysts?

It is easy to align the thoughts with that of the majority. But if the majority is wrong, it is not always the best thing to do.

Common practices are not necessarily “best practices”.

Improving the Discipline and Skills

Analysing Systems Analysis: Thinking about the Discipline



Our values, truths, beliefs, priorities, assumptions, foundational knowledge and holy principles should be made questionable. We have to be able to genuinely think about it with open mind. This is key to mature to a higher level.

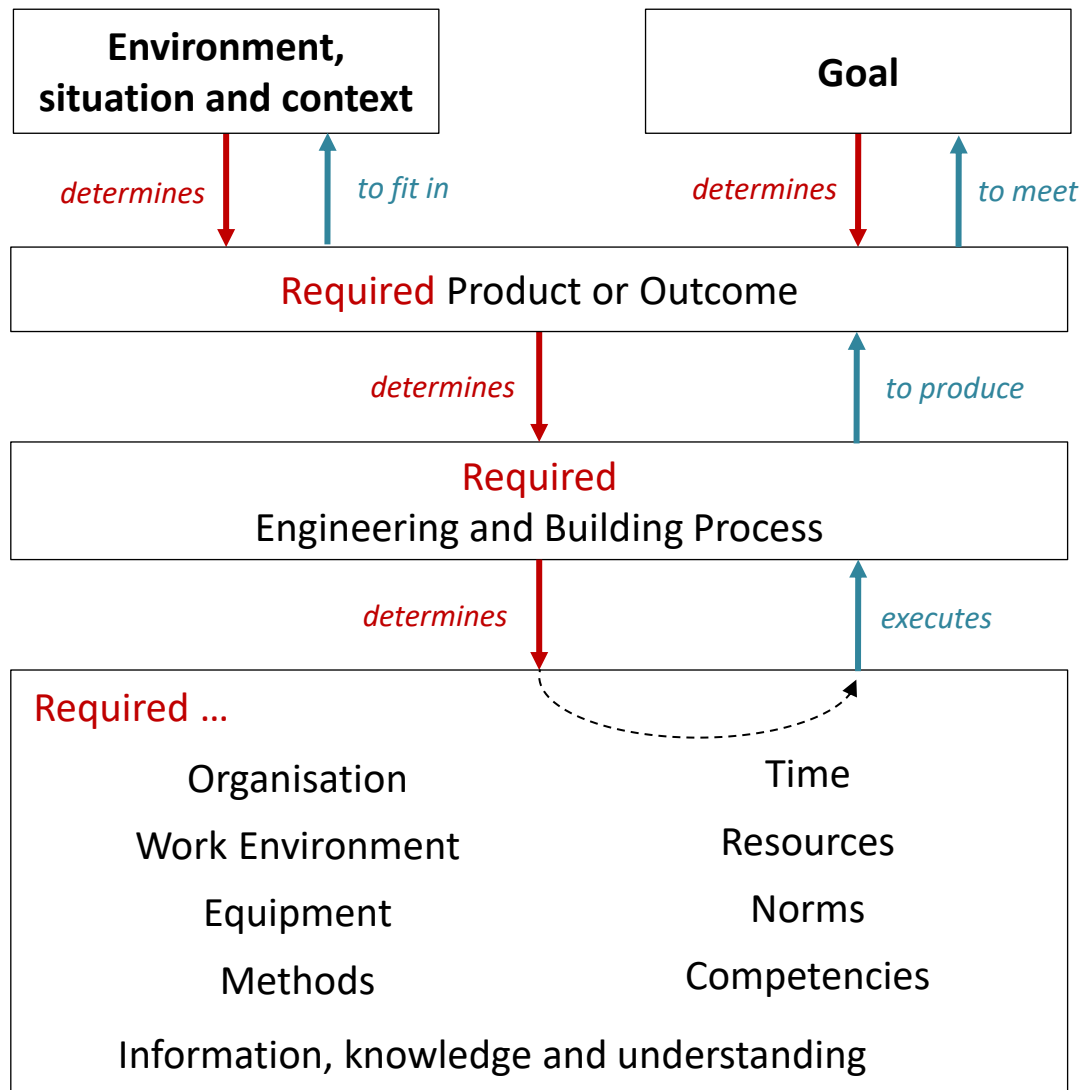
“Who cannot remember the past are condemned to repeat it.”;
George Santayana

Or, we are condemned to repeat our mistakes endlessly
until we learn the lessons we have to.

We keep running into the same kinds of struggles and problems over and over again, until we learn that lesson that allows us to progress.

“Begin with the End in Mind”

30/10/2018



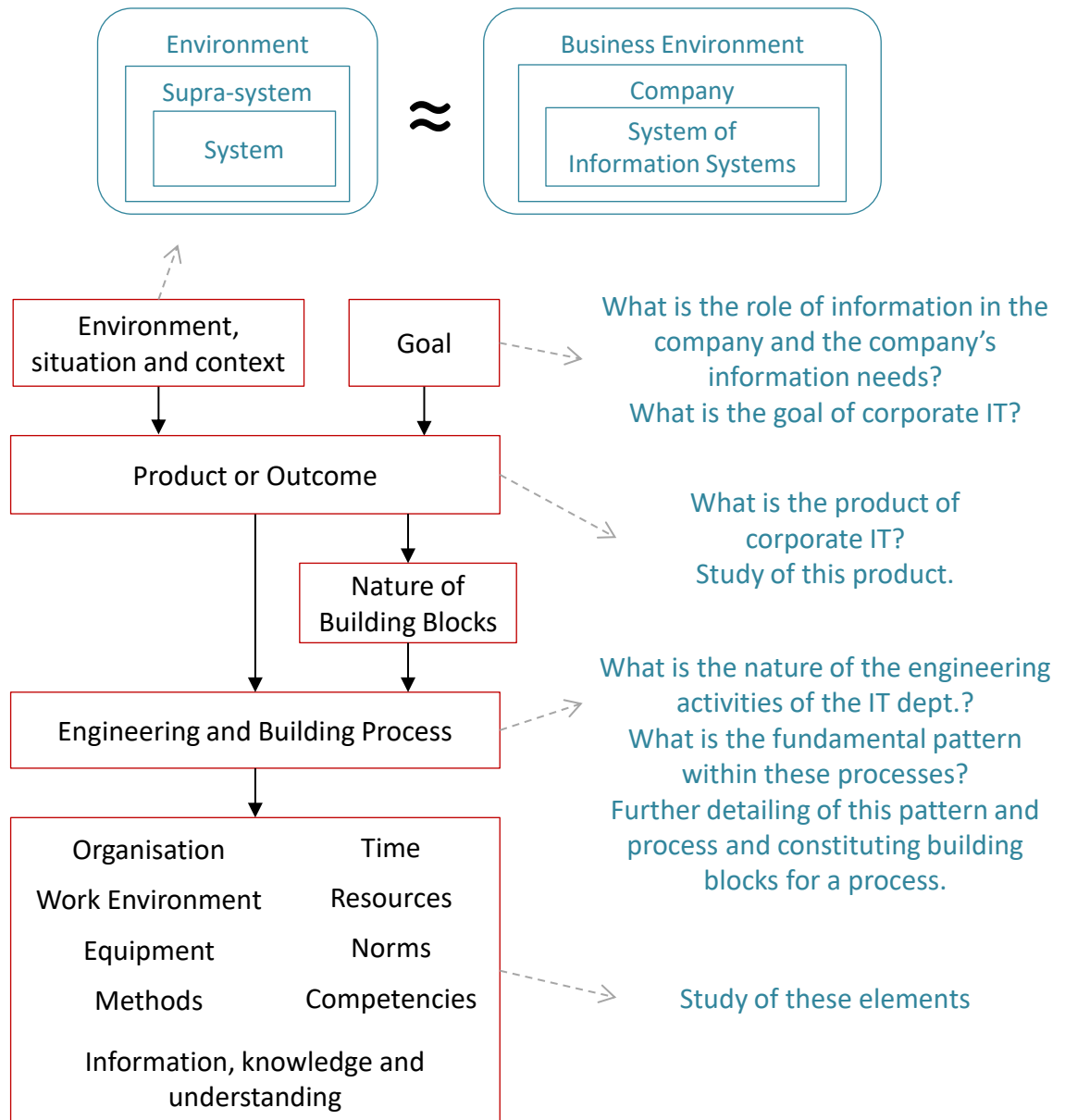
Our world view, beliefs and assumptions determine what we produce and deliver.

Though, it can hinder and even limit the ability to create value!

The environment, the situation, the context and the goal we seek to achieve will determine everything else.

This is thus the starting point.

Overall Approach of the Elaboration of TARB



Notes:

- Not all goals are expressed. Some are assumed. Or, we may not be aware of some goals and thus ignore them. Understanding the nature of goals (goal modelling techniques may help) is crucial.
- Many solution requirements will come from the study of environment, context, situation and goals.

KEY TO PROGRESS

“In times of turmoil,
the danger lies not in the turmoil
but in facing it
with yesterday’s logic.”

- Peter Drucker

“We cannot solve our problems with the same
thinking we used when we created them.”

- Albert Einstein

“There is no more neutrality in the world. You either
have to be part of the solution, or you're going to be
part of the problem.”

- Eldridge Cleaver.

“If you don’t understand it, don’t mess with it.”

- Louis Armstrong

“Nothing that results in human progress is achieved
with unanimous consent. Those that are enlightened
before the others are condemned to pursue that light
in spite of the others.”

- Christopher Columbus

“Science is a **way of thinking** much more than it is a body of knowledge.”

Carl Sagan

“As to methods, there may be a million and then some, but **principles** are few. The man who grasps principles can successfully select his own methods. The man who tries methods, ignoring principles, is sure to have trouble.”

Harrington Emerson

“I don’t know what’s the matter with people: they don’t learn by understanding; they learn by some other way—by rote or something. Their **knowledge is so fragile!**”

Richard Feynman

"People mistakenly believe that IT failures are due to a technical problem or a software problem, and in fact it has its roots into the culture, **how people work together**, how they share knowledge, the politics of an organisation. The worse the politics, the more likely the failure."

Michael Krigsman

Source:

The Worst IT Project Disasters of 2013 (by Chris Kanaracus, IDG News Service)


The 6 worst IT project disasters of 2013

http://www.cio.com/article/744455/The_Worst_IT_Project_Disasters_of_2013

[http://www.itworld.com/software/386675/worst-it-project-disasters-](http://www.itworld.com/software/386675/worst-it-project-disasters-2013?page=0,0&goback=.gmp_29008.gmr_29008.gde_29008_member_5829606858213588992)

[2013?page=0,0&goback=.gmp_29008.gmr_29008.gde_29008_member_5829606858213588992](http://www.itworld.com/software/386675/worst-it-project-disasters-2013?page=0,0&goback=.gmp_29008.gmr_29008.gde_29008_member_5829606858213588992)

<http://www.pcworld.com/article/2071520/the-worst-it-project-disasters-of-2013.html>



CORPORATE IT AND SYSTEMS ANALYSIS TODAY

Conventional Beliefs Driving Corporate IT

1. Responding to the business demands and delivering the software systems the business stakeholders wants or needs
2. The domain of IT is about technologies, applying these technologies and building software applications.
3. Building software applications which facilitate the execution of the business community's activities
4. Delivering accordingly to requirements within time, scope and budget
5. Business stakeholders satisfaction is obtained by delivering the desired/demanded solution: a software applications or features
6. Finding solutions in the products offered by the market and improving performance by implementing more powerful hardware
7. Progressing by following the market's tendencies and innovating by acquiring the latest technologies
8. If the business stakeholders are satisfied with the software system, then everything is fine.

Motivation: Obtaining end-users' or stakeholders' satisfaction by supporting business operations through the creation and delivery of software features and applications; accordingly to the demand, expectations, needs, wants and requirements; in order to facilitate the work performed by the business community.



Core questions in this line of thinking

- What is the client's demand?
- What are the client's needs?
- What does the client want? → Understanding what the client wants.
- What software do we need to build and to deliver to satisfy the client?
- What technologies can we use and what is the best way to use them to solve the client's problem or need?
- Is the client satisfied with the presented or delivered software application?

Conventional Beliefs: “the analyst”

30/10/2018

The analyst is

1. Someone who helps the business community to implement the systems or solutions they think they need and want, and which, sometimes, they already imagined.
2. An in-between, a bridge, an interface, a liaison between ‘business’ and ‘IT’
3. A facilitator, coach, guide, mediator resolving conflicting goals and incoherence’s or other issues.
4. A Requirements Collector/Recorder, Requirements Analyst, Requirements Manager
5. Someone who looks inwards in a given subject and study its parts, the relations between these parts and the nature of all this.
6. A Refiner: The analyst refines something that is given to him. He or she looks at the details, clarify what needs to be, fill the gaps and identifies and eliminates incoherencies and check for completeness.
7. A modeller (producing models)
8. An UML-specialist, BPMN-specialist, ...
9. A specification writer / translator (translating business demand into specifications for the developers)
10. Someone who responds to the business demand by finding out what the business community and stakeholders need and want, as well as what their expectations are.

Conventional Beliefs: “Analysis”

1. The Analysis consists of looking at what the business demand, what they need and want in order to create value and satisfy the business stakeholders.
2. Analysis is about documenting what the business stakeholders want / documenting what has to be built. (producing documentation)
3. It is about **looking inwards of something**, to **the parts and to their relations**. (≈ definition in dictionary) → the Analyst must be given something to analyse, to work with.
4. Looking at, verifying and **working out the details**.
5. **Translating** the business demand into artefacts useable **by developers**.
6. The analysis is an obligatory step between business stakeholders and developers.
7. Analysis is about gathering, verifying, detailing and managing **requirements**. Requirements skills are crucial.
8. Analysis is about producing **models** (UML, BPMN, ...) (representing the demanded solution) or specifications. UML or BPMN skills are crucial.
9. Everybody can do an Analysis or can become an Analyst.

Analysis is often seen as a natural and mandatory part in software development projects. Everybody does it. It's real and underlying purpose is to eliminate the doubts and worries and to appease the mind of the stakeholders (“Analysis has been done, so we are fine.”).

The analyst Today

The **analyst** is an “*expert*” who

- receives an objective and a mission
- is being told what problem has to be solved
- is being told what is expected
- is being told how to work and what techniques to use
- is being told what the desired solution is

The **analyst** delivers business stakeholders the solution they want and asked for, knowing it won't meet the expectations, won't solve the real problem and which sooner or later will get the business into trouble.

He or she doesn't tell them that better solutions are possible, let alone that the wrong problem is being solved

- because (s)he has not been properly trained for and may not even be aware of it,
- because it is not what has been asked for,
- because it is not expected,
- because (s)he is not allowed to,
- because it would be upset the business stakeholders
- and because if (s)he did, (s)he would end up being the bad or negative person slowing down or even undermining the project.

In many organisations, Analysis is considered as an activity or a role which has as **purpose** and **goal** to meet demands and satisfy people.

The **role** of the Analyst is often defined

- as “**bridge** between business and IT” (communication),
- as **facilitator** (collaboration)
- or as **coach** to support the business community to identify issues and finding solutions.

This implies that **Analysis** is perceived and applied as a

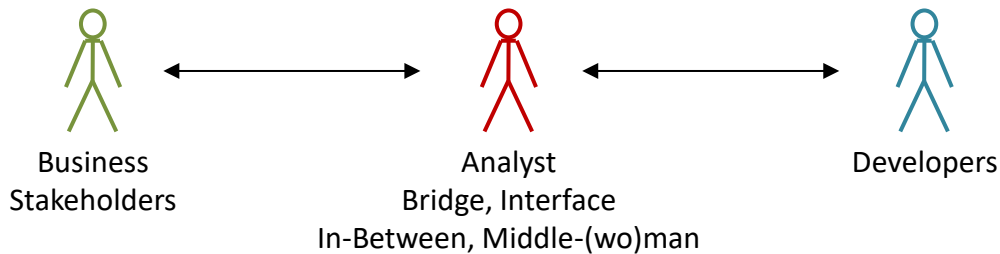
- **A collaboration & communication** discipline (soft skills)
- **A facilitation & coaching** discipline
- **A set of administrative tasks**

These considerations contribute in shaping the view, expectations and approach of business stakeholders.

This has a fundamental influence in the way Analysis is commonly employed and practised in companies.

Notes:

This should be kept in mind when the discipline of Systems Analysis is defined and when it's purpose is defined. Undeniably, communication, collaboration and facilitation are activities performed by the Analyst. This doesn't mean they are part of the essence of the Systems Analysis discipline.



The middle-(wo)man makes sense if this model works, no additional delays and costs and if the middle-(wo)man creates value.

- Responding to business demands
- Initiative and process is triggered, defined and guided by business stakeholders
- Refining and detailing their demand
- Implementing the solutions the business stakeholders and the business community wants
- Producing models and analysis artefacts
- Recording and managing business requirements and specifications

What's the true value that is created and what are the drawbacks? After the Analyst's has done her/his work, do the developers know what they have to develop and does it solve the problem and needs, meet the expectations, allow to reach the objectives?

Typical Recurrent Issues

Information

1. Information is not present or accessible when needed.
2. Unnecessary information being captured, while valuable information isn't.
3. The precise meaning of some information is unknown.
4. Information is duplicated and exists in various forms.
5. Information is incoherent and not updated everywhere.
6. Data is unorganised and hard to maintain.
7. Changing information at one place impacts other organisational units.
8. Poor data quality: outdated, unreliable, vague, insufficient, old and new data are mixed, various formats and encoding standards exist, ...
9. Lack of control on data; difficulty to ensure data consistency and timeliness.
10. The company's information pool is growing. A lot of information is unrecorded, unmanaged, uncontrolled and known by a very few ones. Chaos is looming.
11. Much of the information is not or can not be exploited.
12. Information glut, information swamp, information overload

Processes

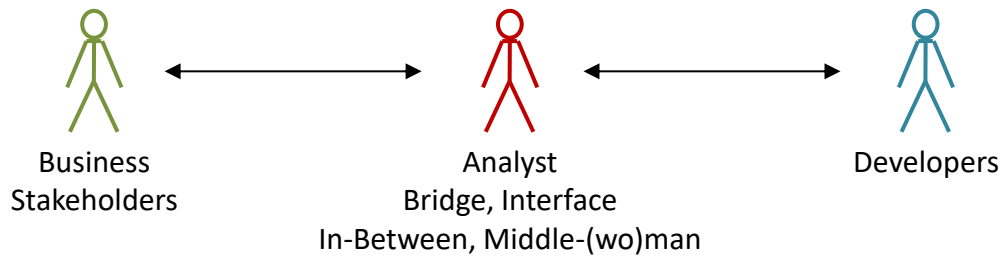
1. Cross-border business processes are slow, complex, awkward and inefficient.
2. Not all processes are inventoried. No overview (and management) of processes.
3. The whole processes aren't understood or only understood by very few.
4. Some steps are performed because they have always been done like that. It's assumed that it is of some value to another department.
5. Business processes may interfere with each other, resulting in unclear, unforeseen situations, exceptions and conflicts.
6. Not all business processes respect a same coherent logic.
7. Business processes are executed in different ways by different people. How can we then measure these processes in a coherent way?
8. No or irrelevant process metrics are collected, while valuable aspects aren't being measured. (not just about KPI's, but about all "PI's").
9. The metrics aren't fully exploited. Or, the right people don't receive these measures.
10. Production can't be planned, because of unknown processes, too much fuzziness, improvisation and variability in their execution.

Systems

1. Software solutions lack features or are awkward to use. Software features restrain the end-users activities.
2. Software applications contain too much unused features and logic.
3. The processes, the systems' architectures and the implemented concepts and the organisation of the data do not match with the real world.
4. Software systems are inflexible and not scalable.
5. Collaboration among software systems is limited and hard to achieve.
6. The software landscape and the technological landscape grew organically to an uncontrollable chaotic whole. Loss of insight in the implemented systems and in their interactions. A lot of corpses in the closets.
7. Adapting the systems became slow, hard and risky up to the point of paralysis.
8. Projects are late, cost more than estimated and, while matching the demand and requirements, they do not suit the needs and do not solve the problems. Often, they even create new problems.
9. Once operational, the systems underperform and do not meet the expectations.
10. Systems and projects do not create the intended business value.

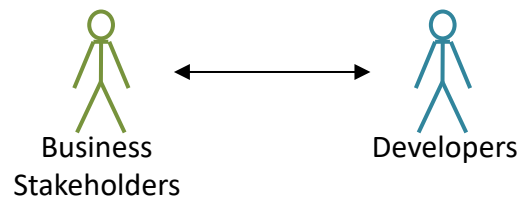
Optimising the Business-IT Collaboration

14/02/2020



If no value, then optimisation.
How?

Cutting out the middle-(wo)man



Elimination of the middle-(wo)man.
Direct talk and collaboration
between the business stakeholders and the developers

After more than half a century of “IT” ...

- Project failures ?
- Satisfied customers ?
- Quality of implemented IT?
- Opportunity gap between “what is” and “what is possible” ?
- Degree of Business-IT alignment?
- Degree of information exploitation?
- Quality of the Business-IT relation (trust, collaboration, ...) ?
- Still looking for better ways to develop software applications
- Newly created problems?
- Confusion about terms, concepts, roles, ...
- Coherence among the definitions
- In search for the self, journey of growth, ‘maturing’(?): unstable; huge evolutions; development of EA; moving from Waterfall to Agile; prone to simple but wrong messages and slogans; hypes diverting it from its path, ...
- Not building upon lessons from the past. Ignoring them.
- Introduction of overly of complex methods, waste, ... ?
- ...

Common Causes of Project Failure

- Unclear goals and objectives
- Loose scope definition
- Moving targets, changing objectives
- Conflicting interests of the stakeholders
- Changing decisions (until the right ones are taken, due to lack of insight)
 - Lack of active executive support
- Lack of SME (Subject Matter Expert) and end-user involvement
- Unrealistic expectations
- ❑ Poor planning or wrong or rigid usage of plans
- ❑ Unrealistic time and resource estimates
- Insufficient communication
- Tackling the wrong problem (consequences instead of causes)
- Bad quality of requirements
- Inappropriate skills
- Disappointing results
- Hidden wasted efforts, hidden costs, ...

- Analysis issue (mainly)
- ❑ Project Management issue (mainly)
 - Other

"God laughs at those who deplore the effects of which causes they cherish."

Jacques-Bénigne Bossuet

Causes of the "Causes"

- People, skills, collaboration, thinking, dependencies, assumptions, belief system, abuses, ... Not technical/technological issues.
- Commonly, early failure warnings/symptoms are not identified or are simply ignored.

Notes:

To reflect: Why (root cause)? And how can we resolve this?
(Some answers are provided by TAR.B.)

Some issues can be solved through a collaboration of Project Manager, Analysts, Architects, project team and some stakeholders.

The Grand Opportunity Gap

18/12/2018

A lot has changed a lot during the last half of a century. More powerful hardware and infrastructure, new technologies and tools. New software development approaches and new professions appeared.

1) Despite these changes, “IT” is still struggling with its old demons. Some underlying widespread beliefs, assumptions, reasonings and mind-set didn’t change very much.

2) The overall approach, the role of analyst in this approach and the analysis activities are inline with the beliefs, assumptions and reasonings. There is no other choice than to note that the **essence**, the **true purpose** and the **objective of Systems Analysis are completely absent**. They are simply **ignored**.

Consequently, Systems Analysis is **immensely underestimated** and **undervalued** and its application very limited. This leads to limited results, a lot of **problems**, **inefficiencies**, **waste**, **lost opportunities**, and so on. Systems Analysis should be able to solve most of IT’s issues. **If Systems Analysis is not applied in accordance with its purpose, with its spirit, the benefits can’t be reaped.**

3) The change in scale and nature of the IT initiatives demands more from IT and forces it to progress.

4) IT and Software development are still in their adolescence.

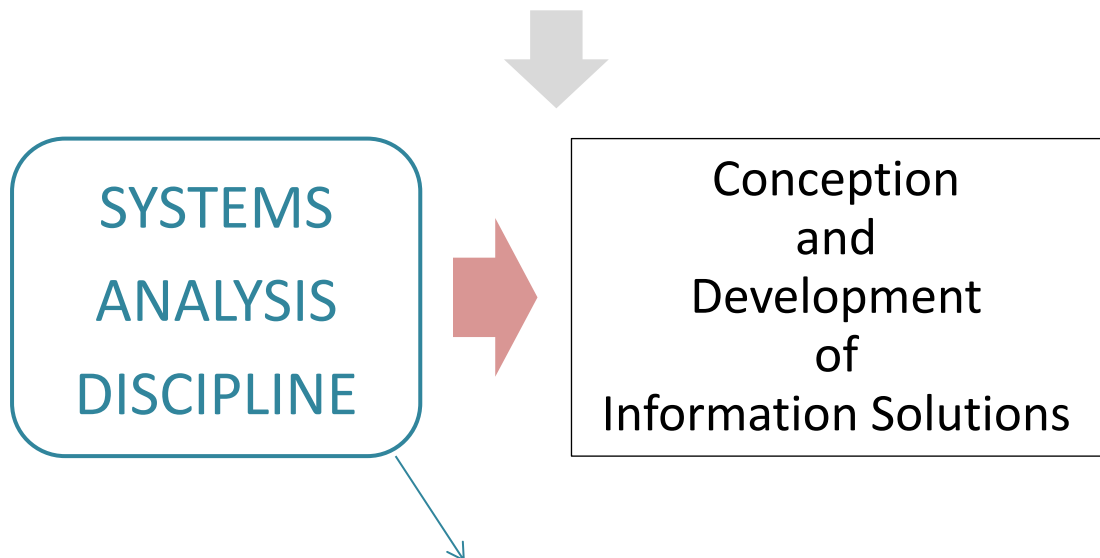
Expecting a more effective use of “IT” and a better exploitation of information is then unjustified.

An immense opportunity for improvement

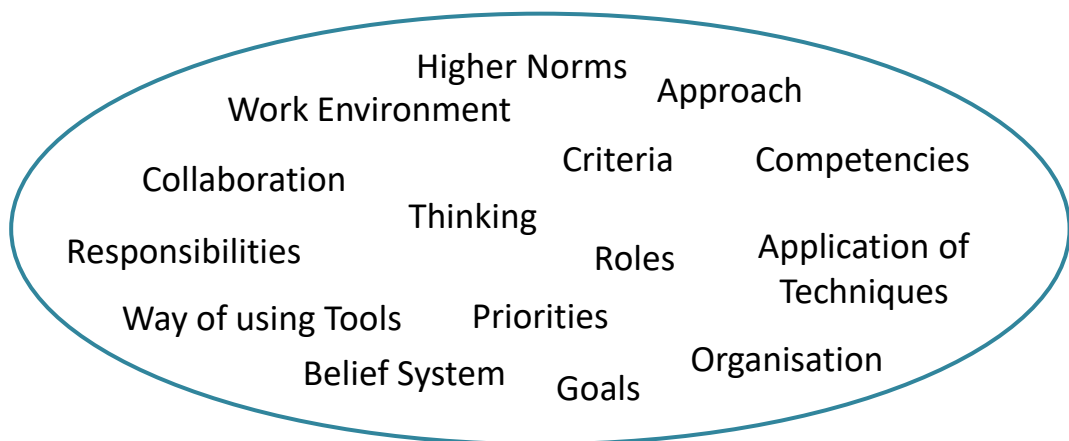
Rethinking Systems Analysis

31/12/2020

1. Pursue of Higher Goals
2. Paradigm Shift
3. Consolidation of the Professional Discipline
4. Professionalisation of Analysts



WE NEED TO RETHINK ...



Notes:

Progress is not achieved by not improving and not changing anything.

Every aspect needs to be in harmony with all the purpose, goal, level and with all other required aspects.

Most people welcome junk food and fast food.
It's easy to provide it to them.

However, it's much harder to determine what
nutrients are necessary and to provide them
in healthy meals.

Is a cook
who is only able to cook meals
by following recipes by the book
a real cook?

Is an analyst
that is only able to do what is told to him or her,
analyse what is given to him or her,
to draw models
and to follow methods by the book
a real analyst?

True Systems Analysis

won't always give you what you
want or what you asked for,

but

it will give you **what you need**,

what is **necessary**,

what is **the most useful**

to **reach** and maybe even to

exceed your **goal**,

even if a stakeholder isn't aware of it
or doesn't fully understand it.



Opening the Mind for New Beliefs

*“Don’t believe
everything you know”*

Intellectual Humility

- Humility of accepting not knowing everything, of accepting your limits and accepting the possibility of being.
- Maybe someone who might know more than we do, may know something we don't know or might be right.

Curiosity

- Curiosity of learning and of seeking what we don't know.

Empathy

- Trying to understand perspectives that differ from our own view and trying to understand why they do exist.

Courage and Honesty

- Having the courage and honesty to evaluate information that differs with the own knowledge and insight fairly and with an open mind.
- Not rejecting information conflicting with our own ideas too quickly.

Being Critical

- Not accepting any information at face value (even if it is something "everybody knows" or something "obvious").

Valuing Truth over Being Right

- Finding out to be wrong allows to eradicate a false "truth" from our knowledge and is an opportunity to progress towards the real truth.

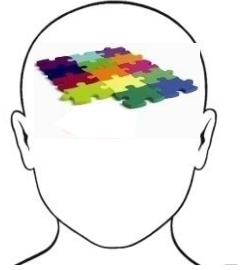
Mental Harmony Puzzle

30/10/2018

PROCESS

The mental picture is created like a jigsaw being solved. Each idea or information element is like a piece.

1. First, we have no mental image.
2. For each piece of information that is received, it is (or not) verified: Is it true? Believable? Does it fit with what I already know?
3. If the new piece of information doesn't conflict with what I know, I integrate it. If not, it is rejected.
4. Gradually, as new pieces are added, a picture is formed. This picture is the reference for all new incoming information. To obtain a harmonious picture, the new pieces of information have to be coherent with the global picture.



= “Mental Harmony Puzzle” (Dr. Chaesan So)

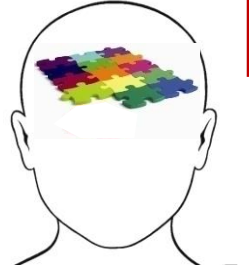


knowledge, map, mental picture,
vision, arguments, thinking
patterns, understanding, ...
already in our mind
*(read “current mental
picture we have about ‘IT’”)*

New idea, new piece of
information, new belief, ...
(read “TARB”)

Mental Harmony Puzzle

30/10/2018



Problems:

- New information that doesn't match our mental picture, creates a discomfort and is likely to be rejected. It is filtered and ignored. Contradicting information is more likely to pass unnoticed.
- Information matching our picture is more likely to be noticed, to be searched for and to be integrated.

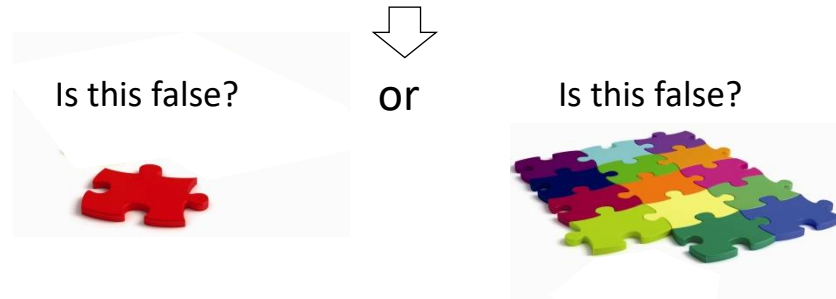
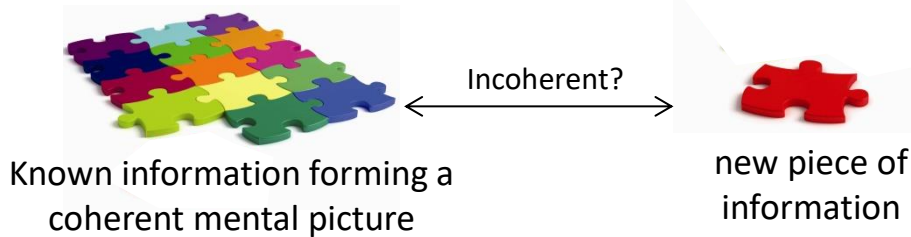
Example: we are more inclined to select sources which are in line with what we already know and want to hear and are more likely to listen to them.

- It is more important that information is in line with what we know and with what we want to hear, than whether it is true or false.
- It is easy to construct a coherent picture with very little information and to be self-confident about it because of its coherence. This self-confidence may exist even though the information and/or the coherent picture is unreliable or false.

It is hard to change our mind, to change our puzzle,
to change what we already know.

Mental Harmony Puzzle

30/10/2018



Ready to see information contradicting our puzzle. If there is an incoherence then we have to investigate what and why?

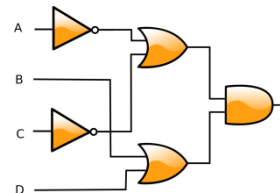
Need to be really ready, to be open-minded and to be willing to question our own knowledge, vision, understanding, belief system, ... as well (even if we don't like it)

Don't believe that everything you know is true

Two ways:



Looking at real world, measuring, experimentation, ..



By thinking, by logic, by reasoning

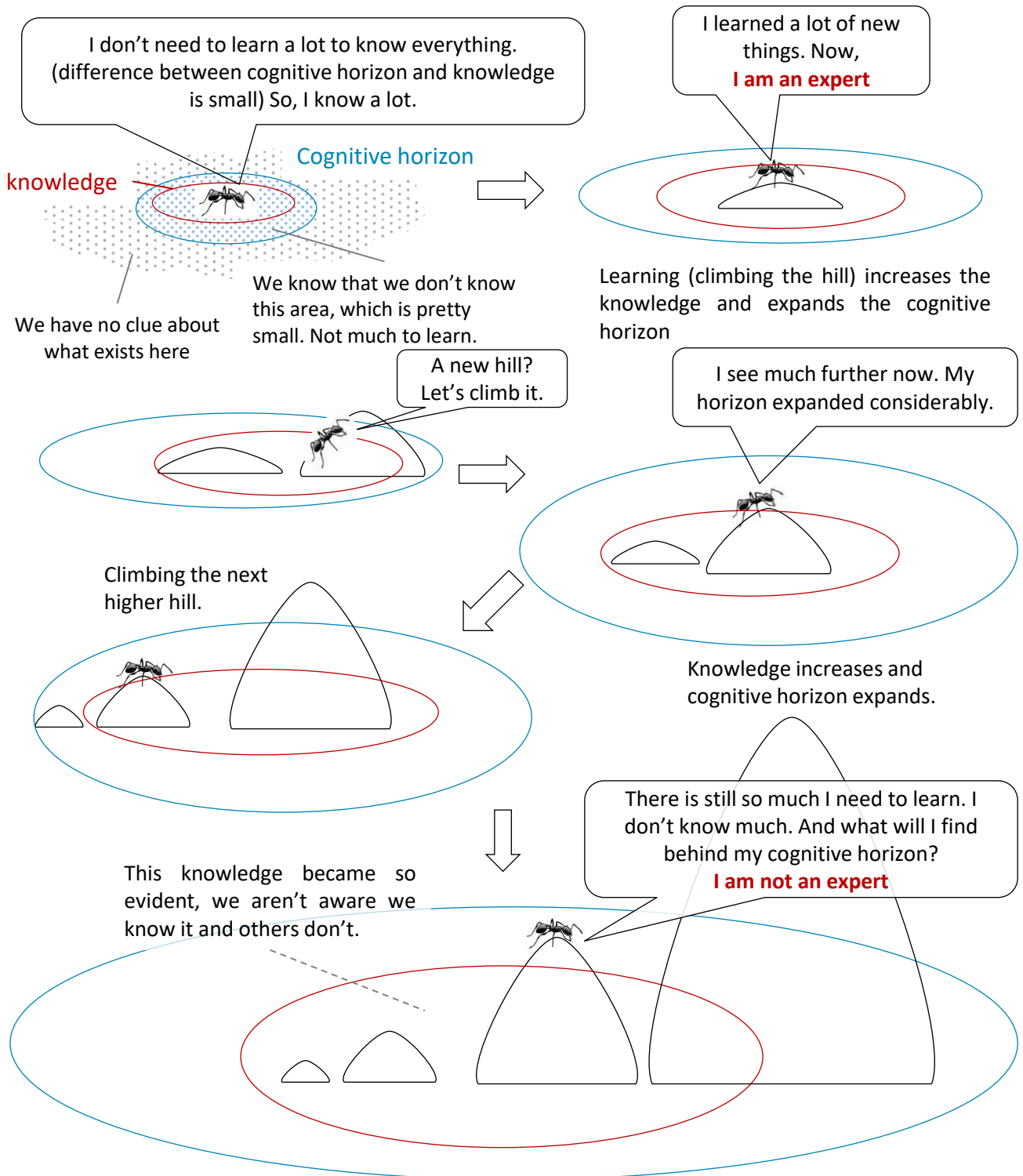
or a combination of both

Important to keep questioning, correcting, adjusting and improving our mental picture, our mental map, our beliefs.

Having a coherent mental picture is important. But, keeping a vision of the world, a logic, a belief system that doesn't reflect the reality, that is distorted, that has unjustified priorities, doesn't make sense. You can't build something lasting upon. The mental image and thinking patterns have to reflect the real world as much as possible. It is better to be a truth seekers, to truly try to know, rather than a protector of the own knowledge.

Knowledge & Cognitive Horizon

30/10/2018



Lacking of knowledge is a problem. But, there is worse: ignoring the very existence of all the knowledge we still can and should learn.

Dunning-Kruger effect: The one who knows little, thinks he knows a lot and is self-confident. And the opposite is true for the one who knows a lot.

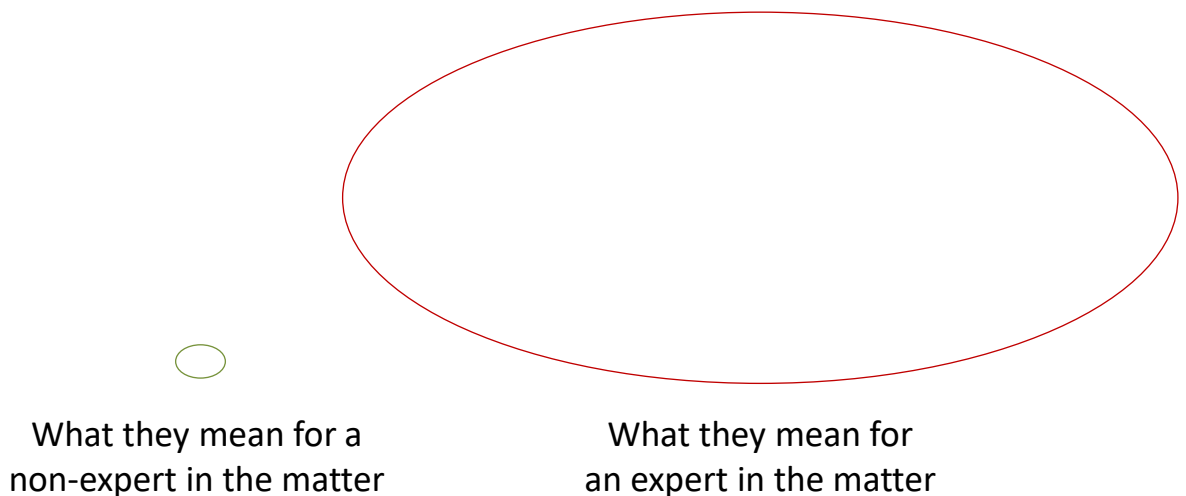
Understanding of Words

30/10/2018

Please, take the following into account:

Everybody understands words like 'analysis', 'design', 'information', 'system', 'solution', 'process', 'engineering', 'project' and 'methodology'.

Most people understand the meaning of these common words. Their knowledge about the subject can be contained in a paragraph, in a few phrases. This creates only the illusion of 'knowledge' or understanding and adds confusion.

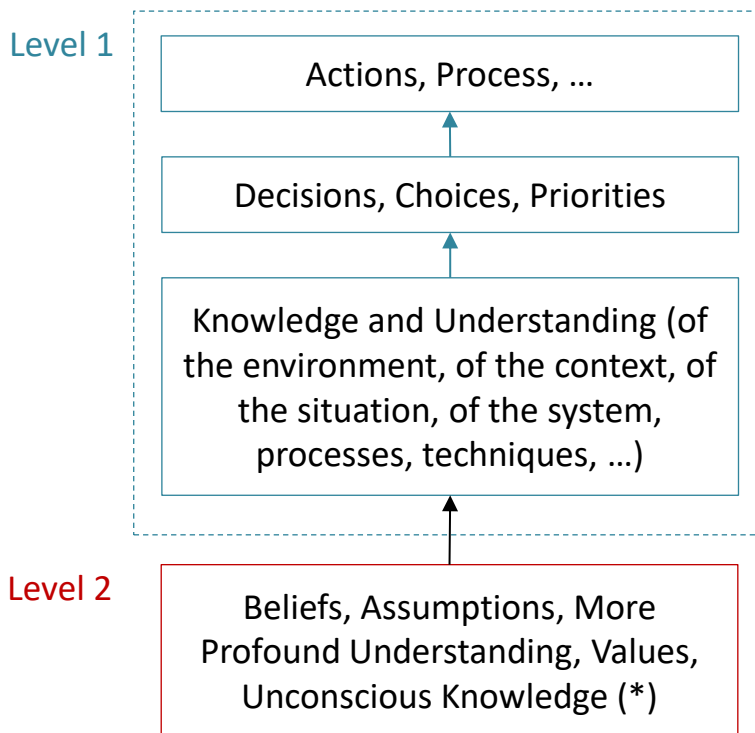


In the light of a **professional discipline**, such an explanation represents only an incredibly superficial understanding. Each of these terms is a subject that can fill books.

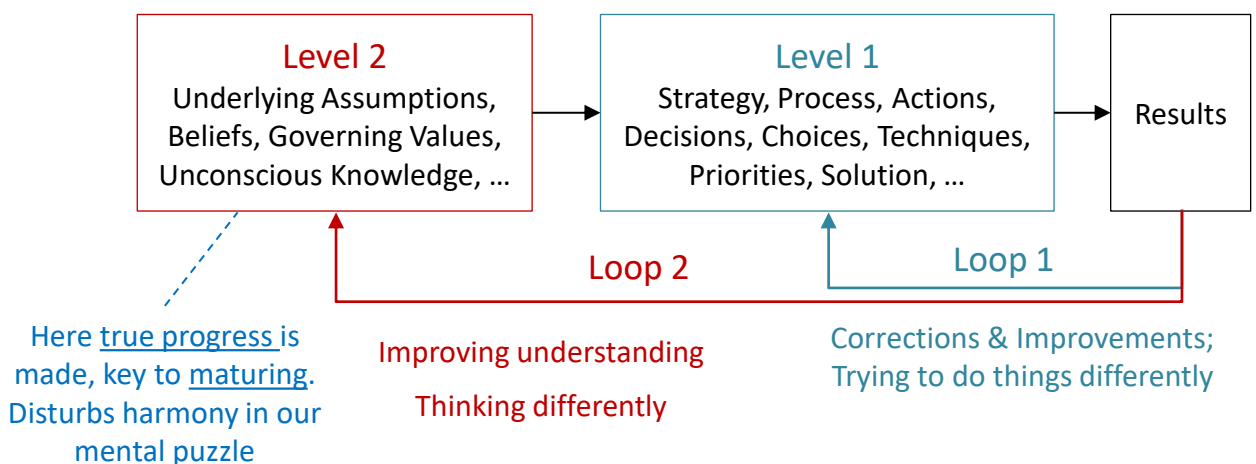
However, we are all on a learning path.

Double Loop Learning

30/10/2018



Double Loop Learning: Chris Argyris & Donald Schön

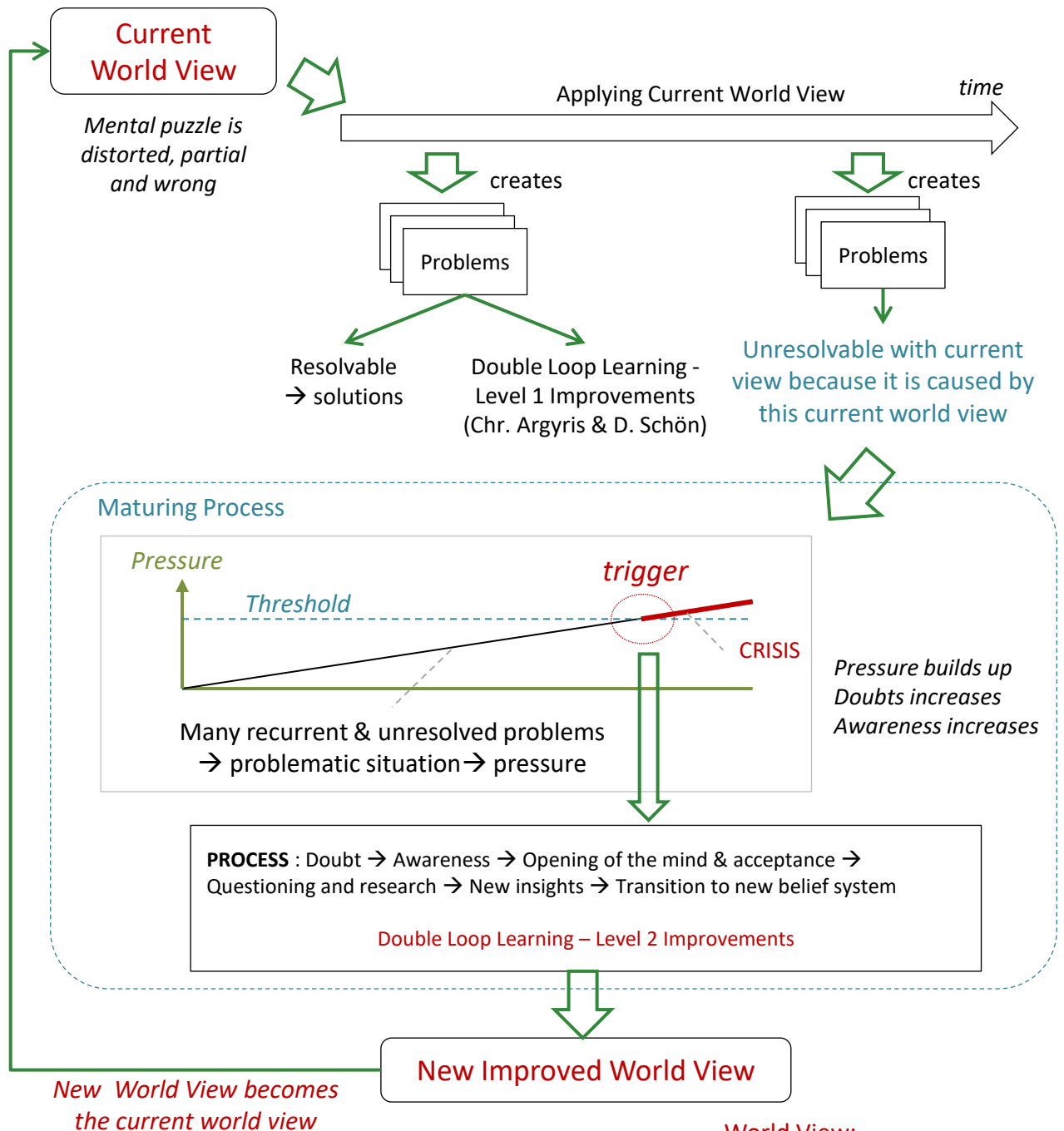


Notes:

"Unconscious knowledge": "we know, but we aren't aware we know it"
Without loop 2 learning, progress is possible, though limited.

Maturing Through Crisis

17/09/2019



Notes

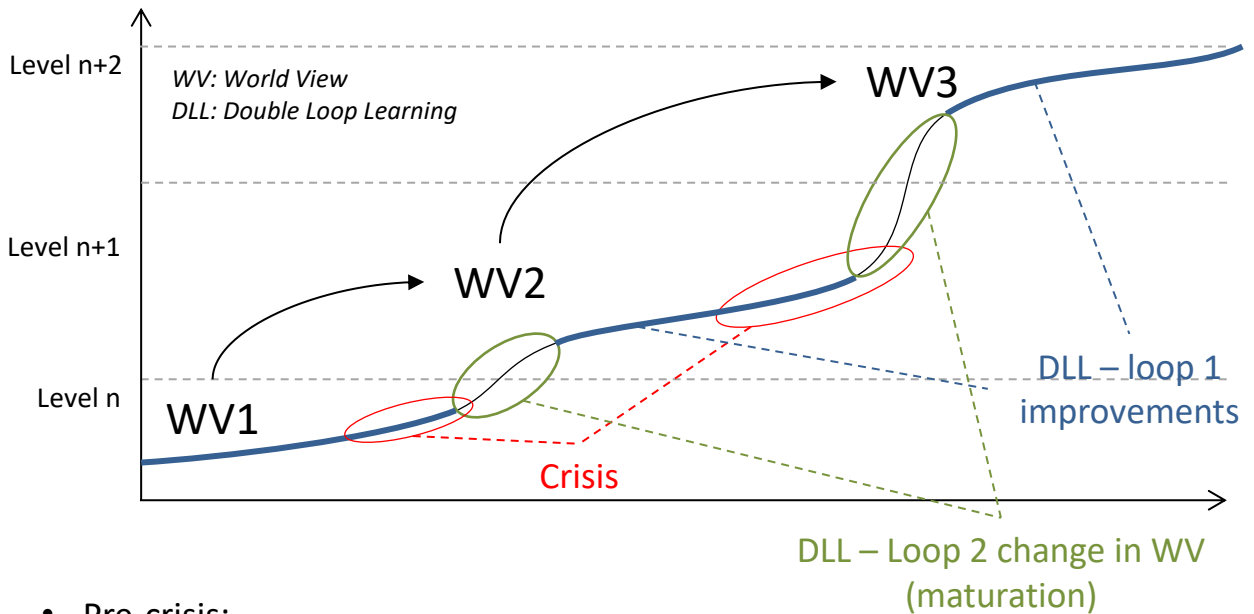
- The described process is a general process. In reality, it is somewhat messier and not necessarily nicely linear.
- Maturing may also happen through reading or any other source creating awareness and a greater understanding. (then not maturing through crisis)
- Possibly, no lesson or not the right lesson has been learned from a crisis. Remaining at same level of maturity.

World View:

- View and understanding
- Beliefs and assumptions
- Values, norms and priorities
- Thinking patterns
- = distorted and incomplete

Maturing Through Crisis

17/09/2019



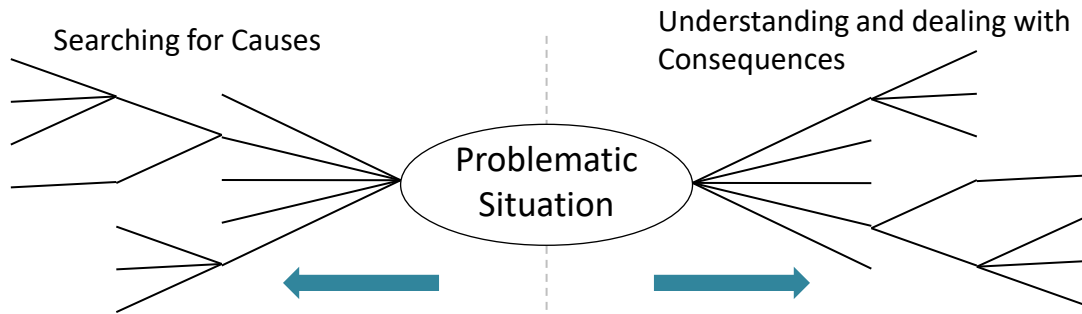
- Pre-crisis:
 - problems exists, inefficiencies, damage, loss. Minor improvements may happen.
- Crisis:
 - Painful, stress. Important pressure for learn and change is built up.
- Maturing World View (belief system, thinking patterns, ...):
 - Acquisition of a new improved, deeper and/or more complete insight
 - DLL Level 2 improvement = a change in the core (not superficial) allowing a kind of significant breakthrough
 - Not anyone succeeds (imitating higher skills without appropriate beliefs, knowledge and thinking → cargo cult)
 - Not any change in 'belief' corresponds with maturing. It isn't if the change in belief goes in the wrong way. (= Trap!)
- Level
 - Each level allows to solve types of problems that we weren't able to solve with the previous way of thinking

Notes

- The changes include changes in belief system, perspectives, thinking patterns, values, insight, understanding of importance, ... They all influence the way of thinking.
- Improvements, be they DLL Level-1 or Level 2, happen irregularly and unexpectedly.
- Crisis is not an obligatory path for maturation.
- The chances for these improvements can be decreased or increased by an (in)appropriate mind-set, environment, atmosphere and actions.
- The more the mind is closed and blind or unwilling to change, the more crisis's will drive maturation.
- Vigilant, open and curious minds have greater chance to mature gradually without crisis.

Causes and Consequences

17/09/2019



Structure of **CAUSES**

... Why ← Why ← Why ← Why
= how to identify and navigate through the structure

Branched structure of **CONSEQUENCES**

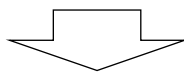
THIS IS THE PROBLEM TO BE SOLVED!

Why?

Annoyances, hindrances and limits

Most consequences are known or easy to know. They are experienced and/or observable. We want to get rid of these. Therefore, it gets the most attention and highest priority (unfortunately).

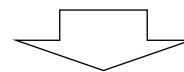
Deeper understanding



RIGHT

WAY TO EVOLVE OR
TO MATURE

Superficial understanding



WRONG

WAY TO EVOLVE OR
TO "MATURE"

Notes

"Why" gives an answer. Then question that answer with a new 'why'.

Other questions: What else? What if's? What if it was different? ...

Most of the time, we think we solve causes of problems, while actually we solve consequences (Symptomatic Problems Solving). This happens for most people and in most cases. It leads to CHAOS !!

- Partial or lacking understanding
- Ideas that do not correspond with the real nature of things and with the real situation
- Accepting seemingly obvious but unverified 'truths'
- Hidden assumptions
- Valuing trying and exploring over learning and thinking
- Doing, activity, action is more important than thinking
- Knowing = Understanding
- Pre-cooked instant answers, relying on common and widely accepted 'knowledge' (preconceived ideas),
- Accepting ideas, explanations, arguments, principles, etc. at face value
- Aligning own opinion with the mind of the majority or group
- Inability to question one's own view, ideas, beliefs, understanding, priorities, ...
- Accepting interpretations, opinions, guesses, assumptions over understanding
- Rejecting contradicting information without prior careful investigation, without true understanding
- Allowing own opinion to be more important than the truth
- Choosing preferences and dislikes over true necessity and righteousness
- Single perspectives, local considerations, simplistic ideas, simple (but wrong) answers, oversimplifications, considering one step at a time, ...
- Rejecting problems, complexity and everything one doesn't like to hear, often in favour of simple, understandable, easy and quick ones.
- ...

Notes:

These habits are contrary to the spirit of a true Analyst

A Few Words of Wisdom

30/10/2018

... describing the mind-set of TARB

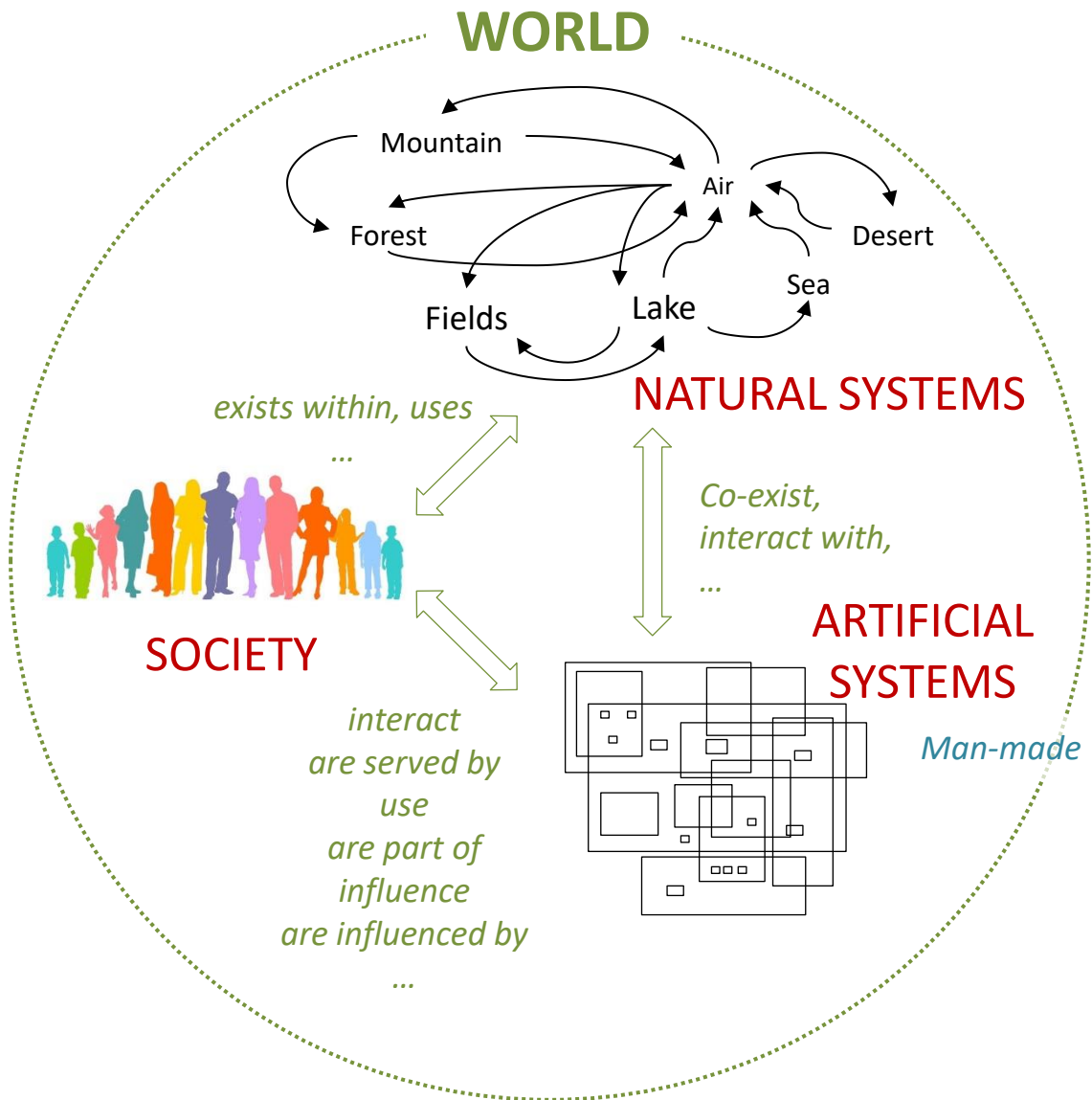
- Today's problems come from yesterday's solutions.
Peter Senge
- If you only read the books that everyone else is reading, you can only think what everyone else is thinking.
Haruki Murakami
- Mistakes are a great educator when one is honest enough to admit them and willing to learn from them.
Alexander Solzhenitsyn
- You don't drown by falling in the water; you drown by staying there.
Edwin Louis Cole
- Wrong does not cease to be wrong because the majority share in it.
Leo Tolstoy, A Confession
- There is nothing wrong with change, if it is in the right direction.
Winston Churchill
- Look deep into nature, and then you will understand everything better.
Albert Einstein
- I'm trying to free your mind, Neo. But I can only show you the door. You're the one that has to walk through it.
The Matrix
- There are no traffic jams along the extra mile.
Roger Staubach



INTRODUCTION

How Important are Systems

10/01/2020



Some systems are imposed upon us. And we can't live without.

Definition:

A system is a set of interrelated or interacting entities that form a unified whole.

Notes:

All living creates are systems as well.

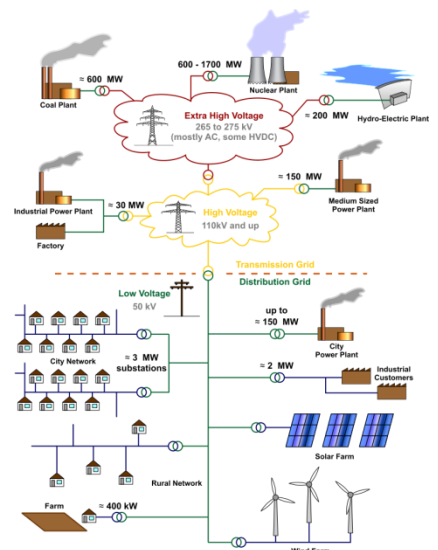
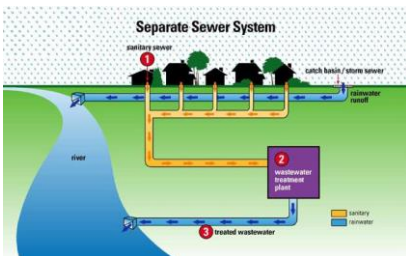
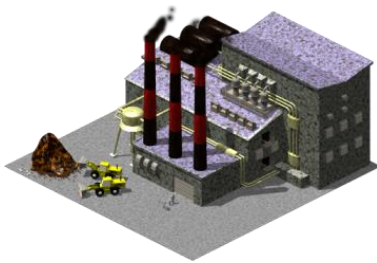
And the society is in itself a system and is organised in many different systems.

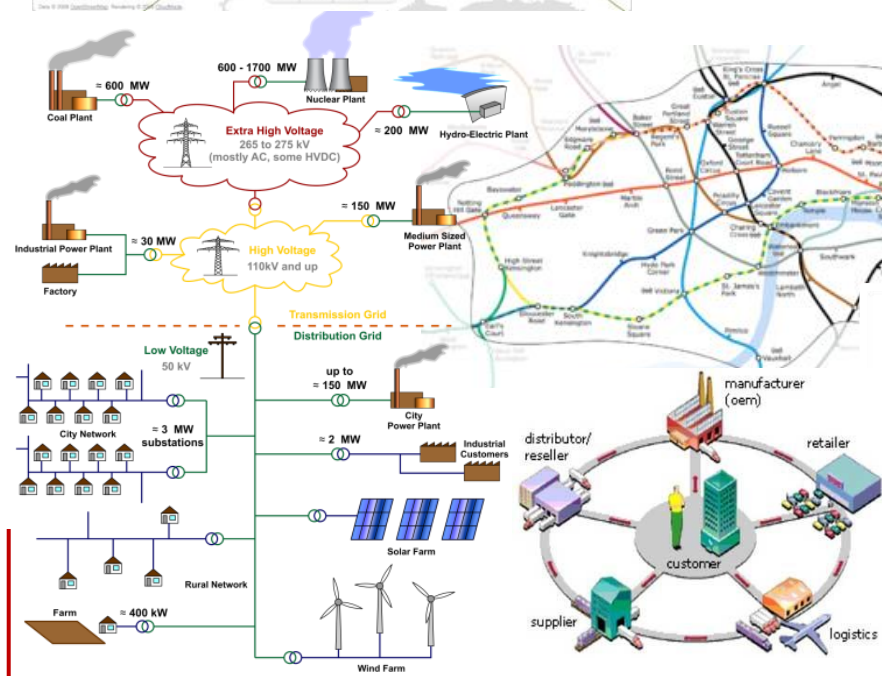
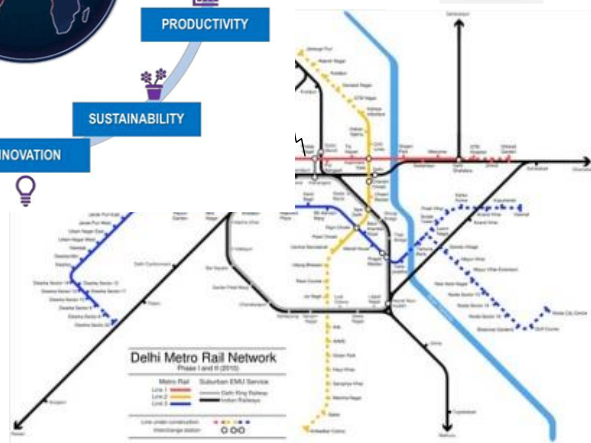
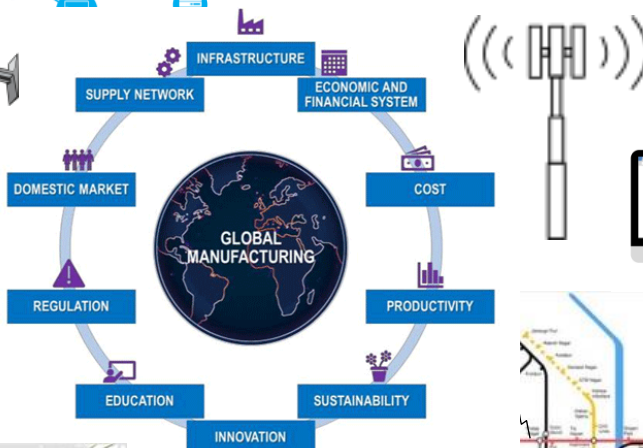
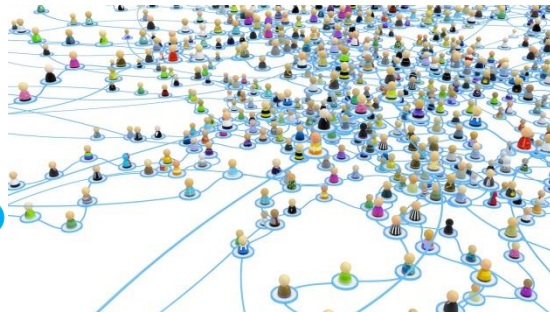


Governmental System

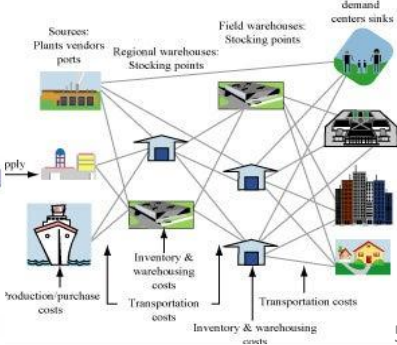
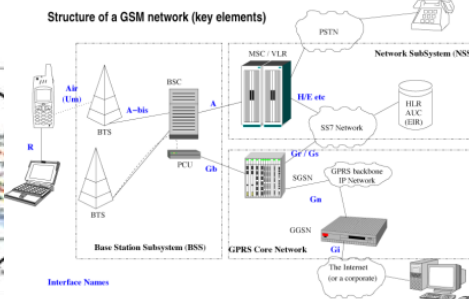


MODERN SOCIETY





Structure of a GSM network (key elements)



INFORMATION

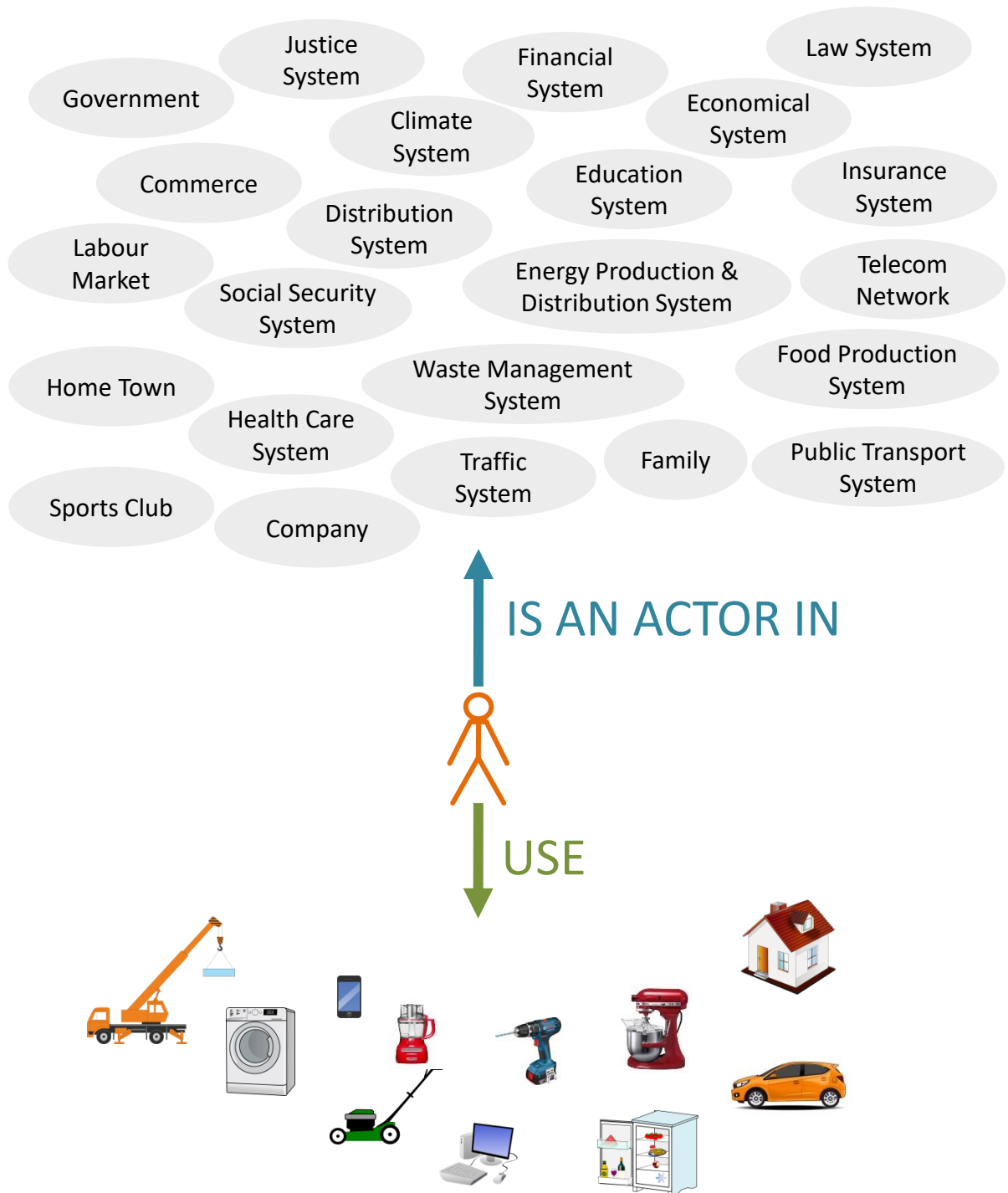
SYSTEMS

MODERN SOCIETY

HYPER-CONNECTEDNESS

UP

- ↑ Volume of Information
- ↑ Accessibility of Information
- ↑ Number of Systems
- ↑ Size of Systems
- ↑ Number of Actors and Systems
- ↑ Complexity of Systems
- ↑ Number of Connections
- ↑ Speed of Connections



Notes:

We tend to be more aware of the physical systems as they are visible and as we use them. We are lesser aware of the natural systems and of the larger systems we are part of. Artificial larger systems shape our society. Artificial (man-made) systems have been or can be engineered.

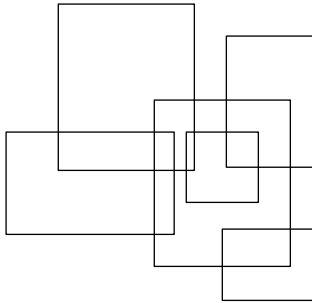
Why do we create systems for ?

For ...

- Organising and Governing the society
- Increase Speed, Efficiency and Effectiveness
- Standardisation, Homogenization
- Facilitation of human activities
- Exponentiation of Human Capabilities
- Control
- Increase of Quality
- To perform work people are incapable to do
- Limiting or transferring Risks to machines

Artificial Systems are man-made. Some are engineered.
Not all creation of systems can be called 'engineering'.

- Man-made systems have a purpose and an intended usage.
- Systems enhance human capabilities. They multiply the effect of our decisions and actions, as well the favourable ones as the undesirable and harmful ones.
- Systems behave in accordance with their own design and implementation. They follow their own logic. They also obey to higher laws of nature. They operate within their own limits. Their behaviour changes when limits are crossed.
- Systems tend to have more effects than only the intended ones, may remain unnoticed.
- We fail much more often than we think to diagnose the right problem. And often, consequences are solved instead.
- Many of the problems we face today result from our decisions in the past. We created them. And today, we plant already the seeds of the problems we will have to face tomorrow.
- Many problems are caused by present systems. These problems are embedded in the design of systems (in the “solution”). By improving a system, we may make things worse. Solving profound problems require a more profound change of the system/solution. Therefore, we need to change our thinking, our knowledge, our values and our beliefs.
- Systems should be designed in accordance with the intended purpose, respecting the laws of nature and suiting the environment.



World of Systems



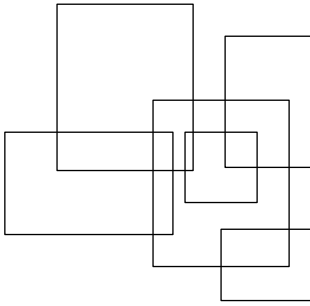
World of People

A Universe of Two Worlds

- People create some systems (man-made systems vs natural systems)
- **World of people is different from the world of systems.**
- People need to understand systems and to adapt their thinking when conceiving and dealing with systems.

Remark:

Everyone can create simple and small systems or create somewhat larger systems by a lot of trials and corrections. It's better to make use of true competencies.



World of Systems

The system runs by obeying only to its own rules. Changing those rules implies changing the system.

- Inviolable principles
- Structure
- Mechanisms
- Interactions
- Processes
- Influences
- Triggers
- Hard logic cause-consequence
- ...

The beliefs and way of thinking of used in daily life are totally inappropriate to design systems because the internals and principles governing systems differ completely from how we live and behave in our reality, in our daily life, in our visible world.



World of People

DAILY LIFE

How someone thinks, takes decisions and acts when meeting friends, lives at home, acts in the supermarket or at the sport club, ...

KNOWN BY ALL

- Goal seeking
- Intention-driven
- Emotional-driven
- Adjustments of priorities
- Capturing information continuously
- Ambiguity & Interpretation
- May use partial, distorted and partial information
- Emotions and feelings
- Choices, preferences and tastes
- Influenced and influencing
- Continuous exploration
- Continuous learning
- Experience-based
- Creative, spontaneous, volatility
- Negotiating
- Continuous adapting to new information (on-the-fly decisions)
- Highly unpredictable
- Unconscious mind, needs, ...
- ...

Example:

A person wakes up. She planned her day. She knows what she will do that day. She decides what she will have for breakfast and prepares it. Then she decides what she will wear when going outside. The person will visit a client, having a dinner at a restaurant. She drives to the client by car. Once she approaches the client's office, she looks for a free parking place. Suddenly, once parked, she realised she had to make a phone call. As she is earlier than planned, she makes the call. But she got bad news. Now she is disappointed. But she has to go to meeting the client. It went well. She goes to a restaurant for dinner. She discovers the dishes in the menu. She choses one and orders it. After her dinner, as she is very satisfied, she decides to take the afternoon off and to go shopping.

The person knew what she would do that day. She didn't know all the details. She has a goal. The path to reach the goal is not fully known in advance. She continuously capture information from the environment and takes decisions. She has emotions and those emotions influence the decision she makes. She improvises. Everyday, even if we perform the same task, like preparing a breakfast, it will be slightly different.

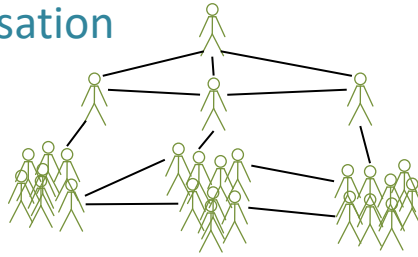
On the other hand, the car functions always the same way. Starting the car happens always the same way. Switching on the lights happens always by using the same switch. By turning the steering wheel to the left, the car will go to the left.

The car will never decide that today, for a change, turning the steering wheel to the left will make the car to turn to the right or to switch on the four direction indicators.

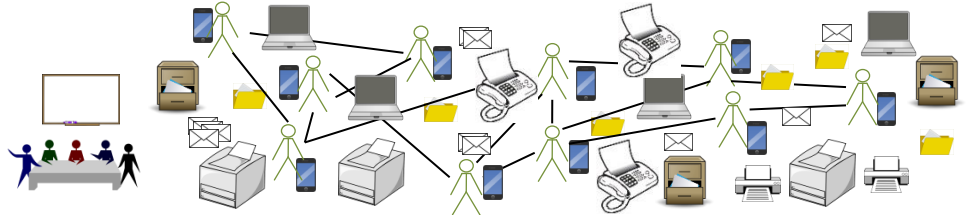
Company / Organisation

*functioning as a
single integrated
coherent system*

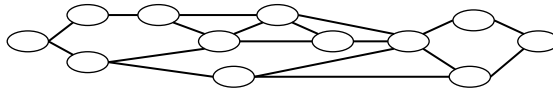
Human
organisation



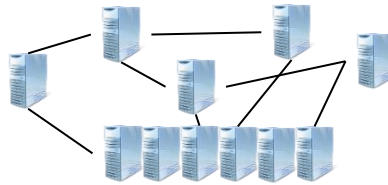
Free, ad hoc,
undefined or
semi-defined,
adaptive
activities



Formalised, structured and
automatable
information processes



Automated
information processes
and information storage



+ decision making processes
+ communication flows
...
Connected with the
environment and with other
organisations



- Pervaded with information – can't do without it
- Implemented in systems
- Organised in information systems, software applications, information architecture, ...

- Organisations, as a system, needs information to function.
- This information is captured, processed, stored, distributed and used.
- Some of these actions on information are or can be structured and formalised. Some are ad hoc, freely executes activities.
- Structured processes can be automated and executed by software and computers.
- Unstructured processes can only be supported with automated systems.
- Systems provide information when and where needed.

The **organisation**, the **company**, the **enterprise**, needs to **be able** to **exploit** all the information she needs for **functioning** (operations), for **management**, for **transformation**, for **guidance** and as **products** and **services**. Therefore, it has to be able to

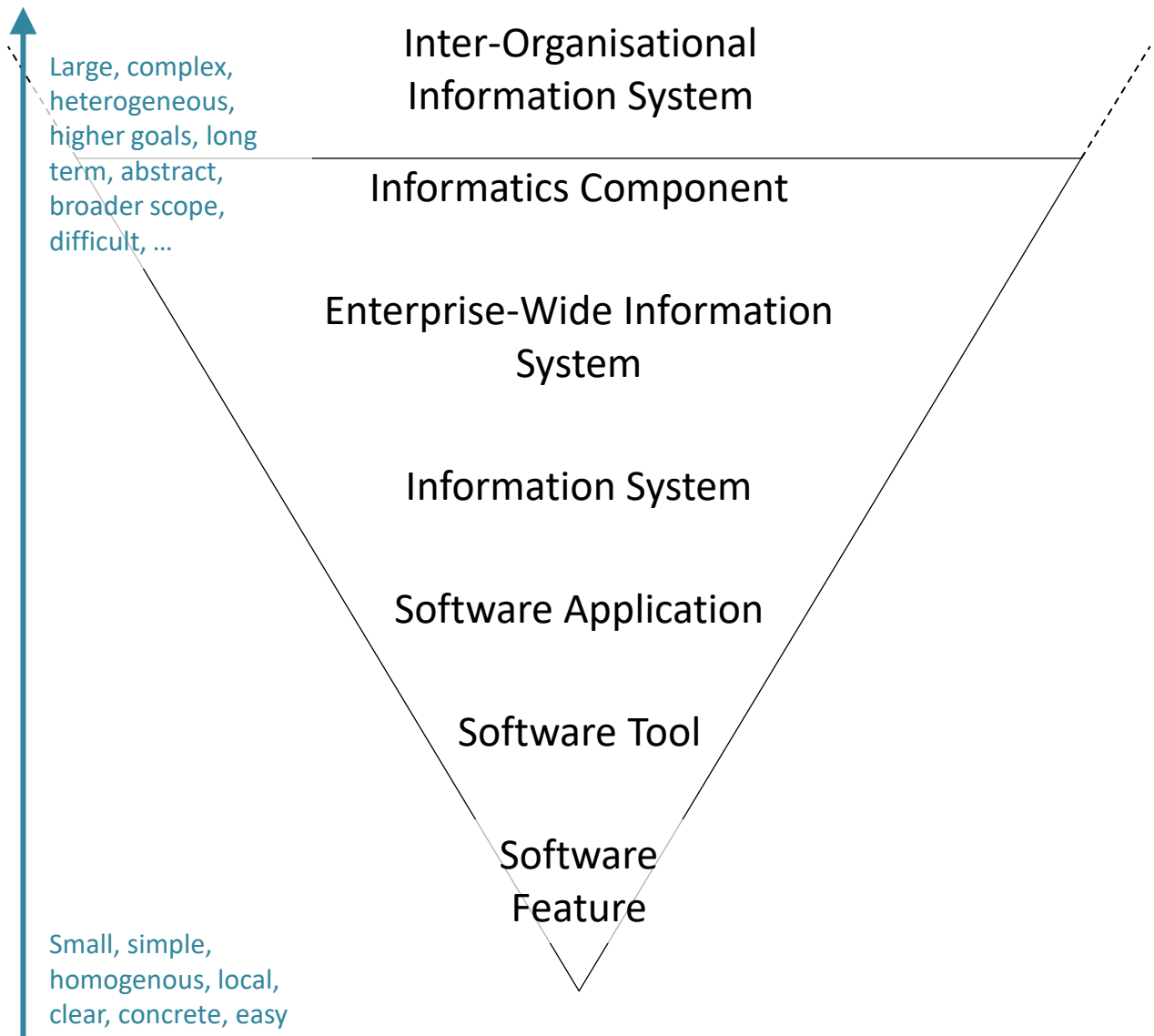
- to **capture, acquire** or **create relevant** information
- to **verify** this information if possible
- to **process** it coherently
- to **store** it in an organised way
- to **distribute** it where and when necessary
- to allow the **usage** of this information **in the most valuable ways** (ways useful to the organisation).
- to **maintain** the information
- to **maintain** or **increase the value** of the information in so far it is possible and beneficial, and **avoiding destroying** it.
- to **use** the information **correctly**

The organisation, company or enterprise **has to be equipped with coherent capabilities**. A lot of it implemented in a **system**, be it a **system of systems**. The organisation of information is crucial.

These structured activities should **preferably be automated**, while the more spontaneous variable activities can be **supported by automated systems**.

This involves much more than only the software-component of the organisation. We may not restrict it to the software-perspective.

Levels of Information Processing Systems



The approach, method and skills required for developing an enterprise-wide information system is not suitable to develop a few small software features.

Equipping an organisation with an enterprise-wide integrated information system is a large and demanding endeavour. We can not develop such a system with the level of thinking, beliefs, priorities, approach, norms and competencies just suitable to develop a software feature.

Small and **simple** endeavour
to solve a straightforward problem

- Easy to grasp
- Straightforward task
- To achieve lower goals
- Not risky
- Short term
- No broader vision, little view ahead
- Small changes

Requires :

No advanced techniques or advanced skills are required. Anyone can do it and obtain results.

Implications :

- Quickly executed
- **Easy success**
- **Only local understanding and insight**
- No broader vision, little view ahead
- Lesser value creation
- **Impossible to solve problems of a higher level of thinking**
- No or insignificant competitive advantage
- A company can't compete with small and easy software initiatives

Large systems exists. A large system is not simply an amount of small components put together. A true larger solution does not 'emerge' from a bunch of local 'individual solutions'.

Larger and **complex** solutions
in complex environment
to solve more complex problems
and/or pursuing higher goals

- Understanding is **harder** - **much lesser insight**
- More aspects to take into account
- More uncertainty, assumptions and contradictions
- Can not simply be executed based on superficial and easy to grasp knowledge
- Success is harder to achieve

Requires :

- a lot of investigation, thinking and planning
- more control, communication, organisation, collaboration, ...
- more and better approaches, methods and techniques
- higher norms
- **true expertise**: requires developed competencies and test these skills
- specific prerequisites and conditions
- may require additional disciplines
- ...

Implications :

- **Higher quality** of solutions
- Possibility to build **something that matters**
- Possibility to achieve **true competitive advantage**
- ...
- Development of **true capabilities and competencies**
- **Ability** of the company **to build and transform itself**.

Benefits of larger initiatives

- Greater efficiency and effectiveness
- Increased significance, things that really matter
- Greater amount of value created
- Realisation of a vision - Shaping the future
- Opportunity for significant innovation
- Greater competitive advantage
- More demanding → More learning & better practices

A company that is able to tackle only small and simple challenges is weak.

An organisation that is able to deal with larger and complex initiatives is strong. It is better able

- to transform itself,
- to innovate
- and to beat the competition.

Executing larger and more complex initiatives requires a higher belief system, holistic view, advanced thinking skills, higher norms, improved work environment, adapted approach. It also requires advanced competencies. → professionals

Notes:

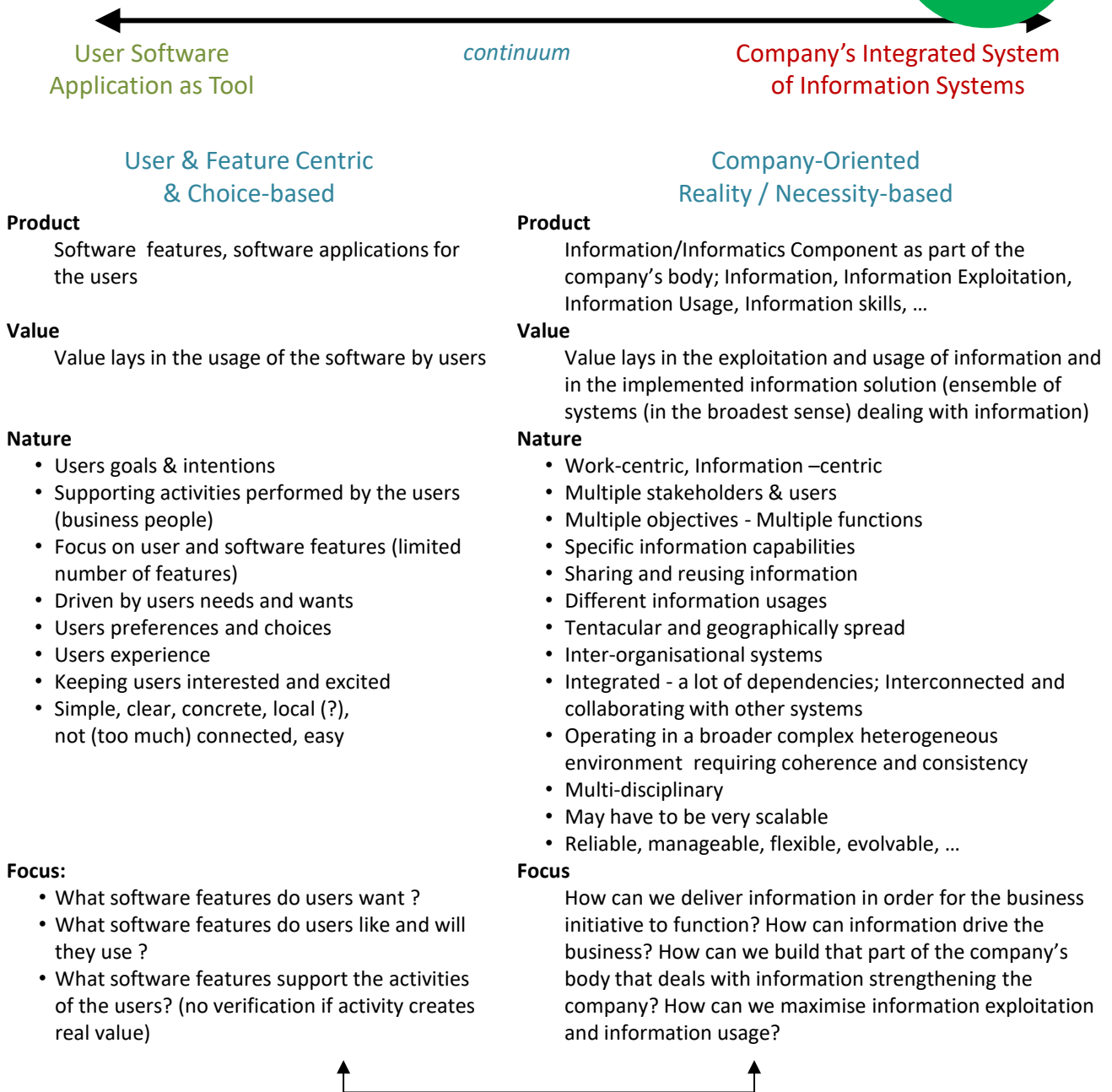
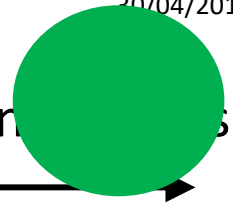
- The development of larger, heterogeneous and more integrated systems creates challenges which aren't present (or not to that extent) in the development of small and simple systems. Hence, the development of larger systems requires, among others, a different work environment, approach, skills, and even a different way of thinking.
- These initiatives are more risky. However, risks can be limited and mitigated. Dealing with risks is also a competency based on analytical skills, flexibility, resilience and creativity.

End-User Software Application

versus

Company's Integrated System of Information Systems

20/04/2019



Different endeavours requiring a different mind-set, a different approach, a different competencies of more advanced level

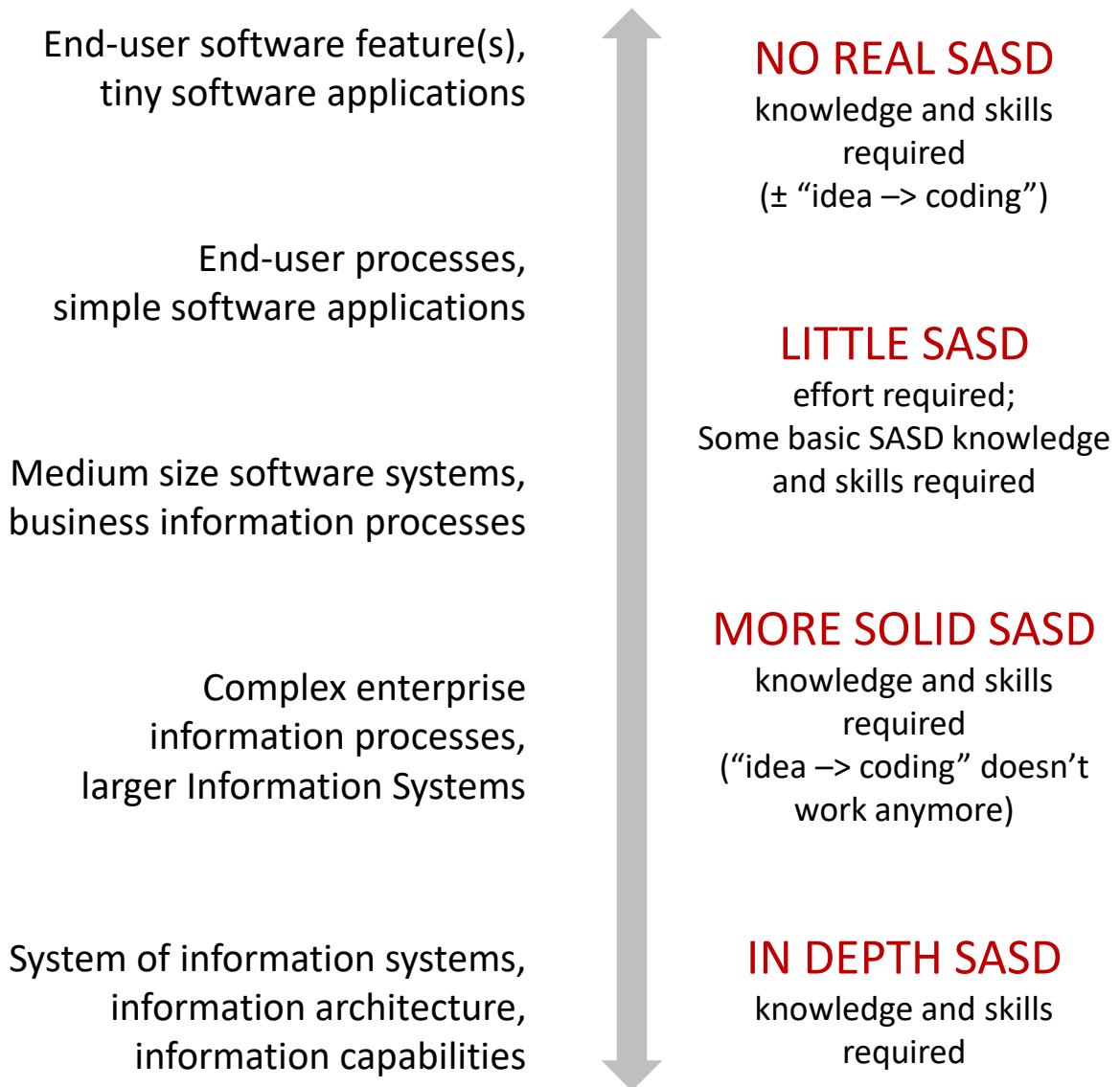
Notes :

- Software for a nuclear plant, an SCM-system or a company's website aren't built in the same way, with same mind-set.
- The final overall product or outcome determines the approach, required competencies, norms, etc. This is much broader than the project's mission and scope.
- The end-users aren't or shouldn't be ignored. But other aspects have a greater priority.

Need for Systems Analysis & Design

14/02/2020

The discipline of Systems Analysis (and variants) does NOT exist to solve the easy problems. Systems Analysis makes it possible to deal with the more challenging, more complex and larger challenges.



Notes:

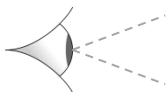
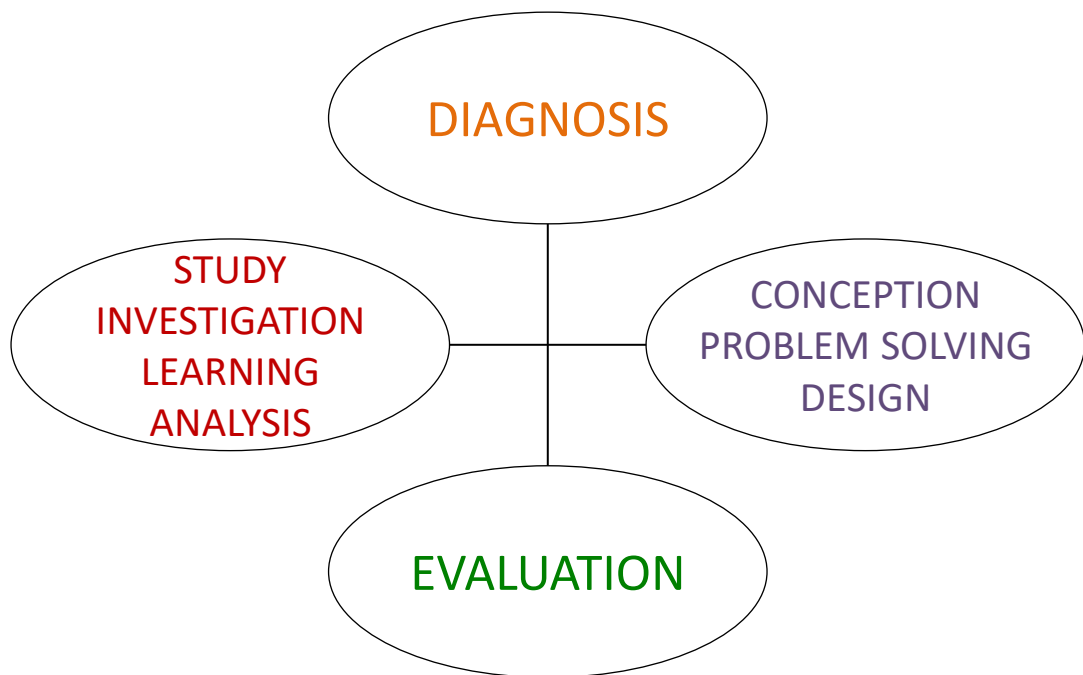
Systems Analysis includes, in its broader meaning, the design of the solution or system. It includes Systems Design (SD).

SYSTEMS ANALYSIS is

- the STUDY of SYSTEMS and their environment
- +
- the CONCEPTION (DESIGN) of SYSTEMS



Four Main Essential Components



TARB considers SA within the domain, context and perspective of Corporate IT / Informatics.

Systems Analysis

is an

Engineering Discipline

Conceiving systems and solutions
to solve more **complex** problems

(solving problems which would otherwise not be solvable)

Dealing with more larger and complex systems creates new challenges:

1. The knowledge fields **information science** and of **systemology** become critical.
2. Increased capability to **detect issues** and **opportunities** and to rightly **diagnose** the problems becomes necessary
3. Ability to **learn a lot and quickly, and to have an organised insight** in order to know what we do and to take better decisions
4. Getting **a grip** on the **complexity of problems** and **systems** and **their environment** is more challenging
5. Solving **information issues**, **systems design issues** and **process issues** (more than software feature issues) becomes harder
6. Greater support needed for **thinking activities**
7. Increased need for **communication** and **collaboration**
8. Avoiding to perform **unnecessary or inappropriate work (rework)** that could have been **avoided by learning and thinking**.
9. Limiting **risks**
10. Increasing the level of **certainty**
11. Increasing the level of **control**

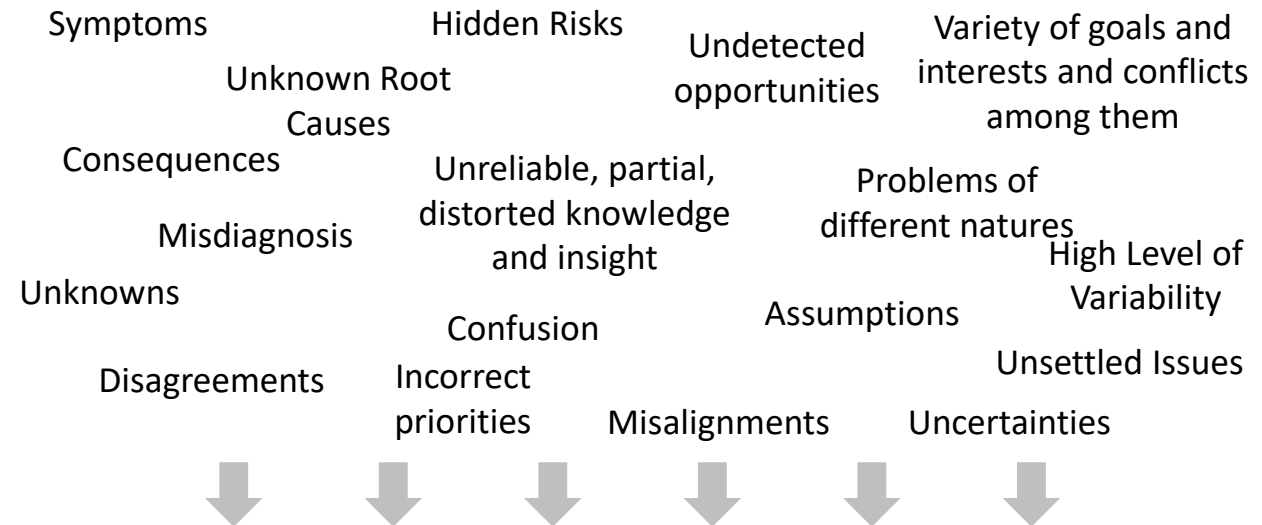
Notes:

Systems Analysis, as general field, existed before IT. This is about "SA for IT".

A sloppy analysis or an imitation of an analysis is usually flawed and unusable. It allows the project to carry on. But flaws are then discovered late. Risks materialise. The product doesn't solve the problem or creates new ones. Many problems ahead.

WHY SYSTEMS ANALYSIS

31/12/2020



SYSTEMS ANALYSIS

- Right Diagnosis
- Contextual Understanding ↑
- Identification of the problems, needs and opportunities ↑
- Understanding of the problems, needs and opportunities ↑
- Common understanding ↑
- Agreements ↑
- Reviewed and agreed priorities ↑
- Unknowns ↓
- Assumptions ↓ (more identified assumptions, lesser implicit)
- Uncertainty ↓
- Risk for changes ↓

*Beware of
cargo-cult Systems Analysis*



doesn't mean it's the final definitive
fully detailed solution !!!

PROPOSED SOLUTION

(better matching with the reality and with the environment,
more elegant, better exploiting the opportunity, ...)

A simple description

Systems Analysis is a professional discipline that studies systems and their environment, adapts them and creates new systems. The intention is to obtain well-functioning, suitable, effective, efficient and elegantly-designed systems that solve problems, facilitate work, create value and help reaching goals.

SA is especial needed for more complex initiatives.

Major Goals

- Detect issues and opportunities
- Improve systems
- Improve system's environments in order to allow a better functioning of a system
- Solve the right (root) problems and needs
- Obtain better systems designs

How to achieve the goals?

- Study goals, intentions, needs, plans, ...
- Study systems and environments
- Evaluate them
- Design solutions

Systems Analysis, in corporate IT, is primarily (mainly) concerned by the **information aspect**.

Diagnosing, evaluating and innovating are also part of the discipline.

Systems Analysis is a professional discipline that studies systems and their environment; identifies issues and opportunities for improvement; defines the adaptations for existing systems or for their environment or conceives new systems in order to obtain well-functioning, suitable, effective, efficient and elegantly-designed systems which achieve or contribute to a certain goal.

- **Professional discipline** → hard to master, steep learning curve, expert
- Studies processes, systems, environments and related aspects like structures, concepts, mechanisms, contexts and situations
- Identifies issues to solve and opportunities for improvement and expansion
- Devises adaptations to organisations, to structures, to processes, to systems and to their environment
- Conceives new structures, processes, systems, concepts, services, products, ...
- Evaluation of structures, processes, systems and their results
- In order to achieve or to contribute to goals → role, purpose

PRACTICALLY: SYSTEMS ANALYSIS spans

from

the detection and identification of issues and opportunities

to

the definition of the solution to be built.

Notes:

- SA deals with systems, processes, structures, concepts, mechanisms, relations, principles, nature of things, evolutions, forces, obstacles, knowledge, skills, ... anything that can be part of a system.
- Note that SA is not simply about looking inwards of a system, to its parts and their relations or only to something that exists (like analyse a document, a demand or given requirements).

Raison d'être - Main Purpose of Systems Analysis

10/01/2020

The core of the **purpose** is to **define the WHAT**

1. Determining problems, causes, gaps, issues and opportunities
2. Determining the solution

In reality, the **raison d'être**, the **essence of the purpose** of Systems Analysis (in Informatics) is precisely **to determine** the necessary information, **to diagnose** the needs and **to conceive the solution** that will solve the challenge, need or problem and how information can be used to drive business activities.

It is an approach, a discipline,
to solve more complex challenges !!!

Analysis is often perceived as and reduced to a skill and/or set of activities. It is NOT. It's much more than this.

The PHILOSOPHY of ANALYSIS^{30/10/2018}

The intent defining the philosophy of Systems Analysis:

**“to conceive the right solution, often a system,
to solve the right problem correctly
from the first time”**

This is an idealistic goal which is indeed often out of reach.

The idea is rather to converge as quickly as possible and as much as possible towards this goal.

It is a matter of real and genuine effort of a skilled Analyst, a matter of trade off, and a matter of true and justified confidence that the obtained solution is appropriate, elegant and rather definitive and that it will correctly solve the right problem.

BASED ON THE IDEA

- Better results can be obtained,
- the building process can go smoother and faster
- and risks can be reduced

by acquiring knowledge and insight and
by thinking before engaging in a building process.

LEARN & THINK FIRST, THEN ACT

Notes:

Doing work that doesn't respect the Analysis philosophy can't be called an Analysis.

There might be clarifications, refinements, (minor) corrections, adaptations, additions and changes after the first version of the solution design.

Learning and thinking never ends. Analysis and design never really ends.

The solution will live its life and will continue to evolve.

Analysis as Problem Solving

1. Detection of the Problem / Determining the Goal

The identification of the exact problem, as the **root cause** or **real objective**, **real need** or **opportunity** is **mandatory** before further actions are taken.

2. Understanding the Problem, Goal, Opportunity, Need

3. Get understanding of the existing systems, of the broader context and of the environment

It is impossible to conceive a submarine without a general understanding of submarines, of the systems the submarine is connected with, of warfare with submarines and without understanding of the ocean.

4. Conceive the solution

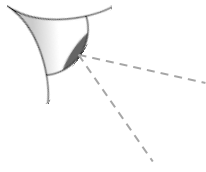
“If I had an hour to solve a problem, I’d spend 55 minutes thinking about the problem and 5 minutes thinking about solutions.”

- Albert Einstein

Notes:

A chapter is dedicated to problem solving.

Systems Analysis is not a synonym of “*Business Demand Analysis*”.



Information perspective

Solving the right
problem rightly

Implementing &
improving business
activities

Improving &
building the
company

Implementing
a strategy

Improving
products and
services

INVESTIGATING

DETECTING

DIAGNOSING

LEARNING

CONCEIVING

EVALUATING

ENVIRONMENTS

ORGANISATIONS

SYSTEMS

STRUCTURES

PROCESSES

RULES, CONCEPTS

.. and more

Studying Systems = “Looking inwards into the system” ?

Building a system while ignoring the environment



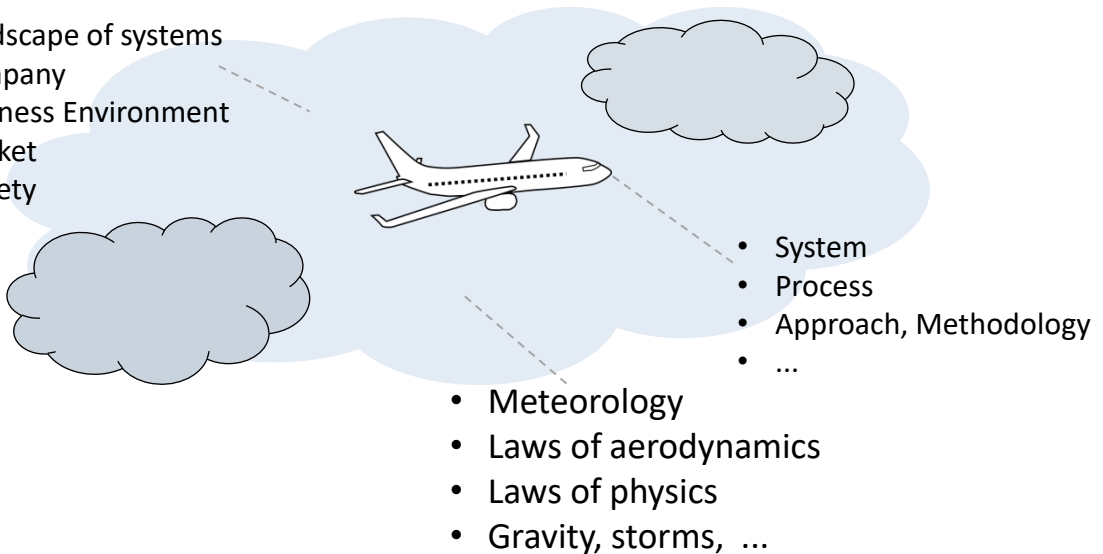
A submarine, even if it is fully functional and innovative, is completely useless in the desert.

Don't try flooding the desert to solve the problem.

If the solution doesn't fit within the broader environment, you built the wrong solution.

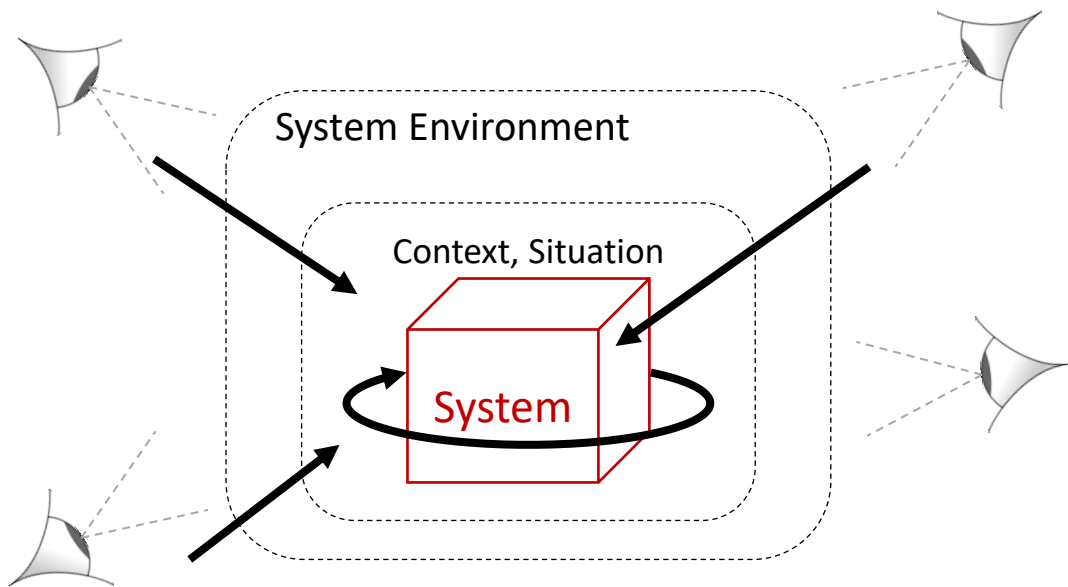
Don't adapt that environment, but rather built a solution fitting the environment.

- Landscape of systems
- Company
- Business Environment
- Market
- Society
- ...



- The **environment** must be studied and understood.
- The environment's **nature** has to be understood and respected.
- The environment's **laws** must be respected.

else you obtain a submarine in the desert



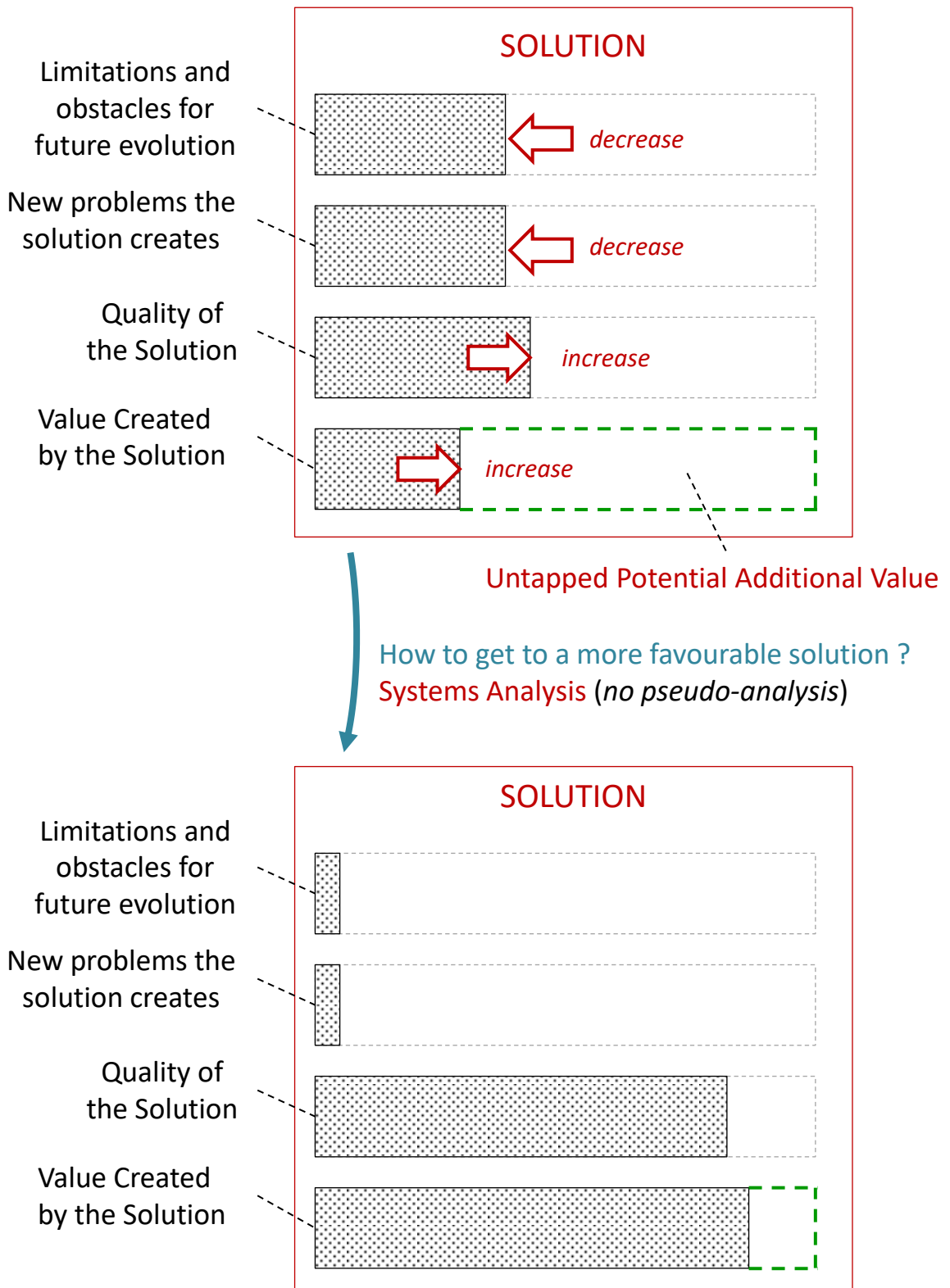
A system exists in a **supra-system** (example: the company). It exists in an **environment**. It operates in an **environment**, in a certain **context** and has to deal with **situations**.

If the system or solution doesn't exist yet, then the future supra-system(s), the future environment, the present and future context and aspects are studied.

The system depends of them. Therefore, this is crucial input for the design and must be studied. Different **aspects and perspectives** (including influences, forces, evolution, cost, ...) have also to be investigated.

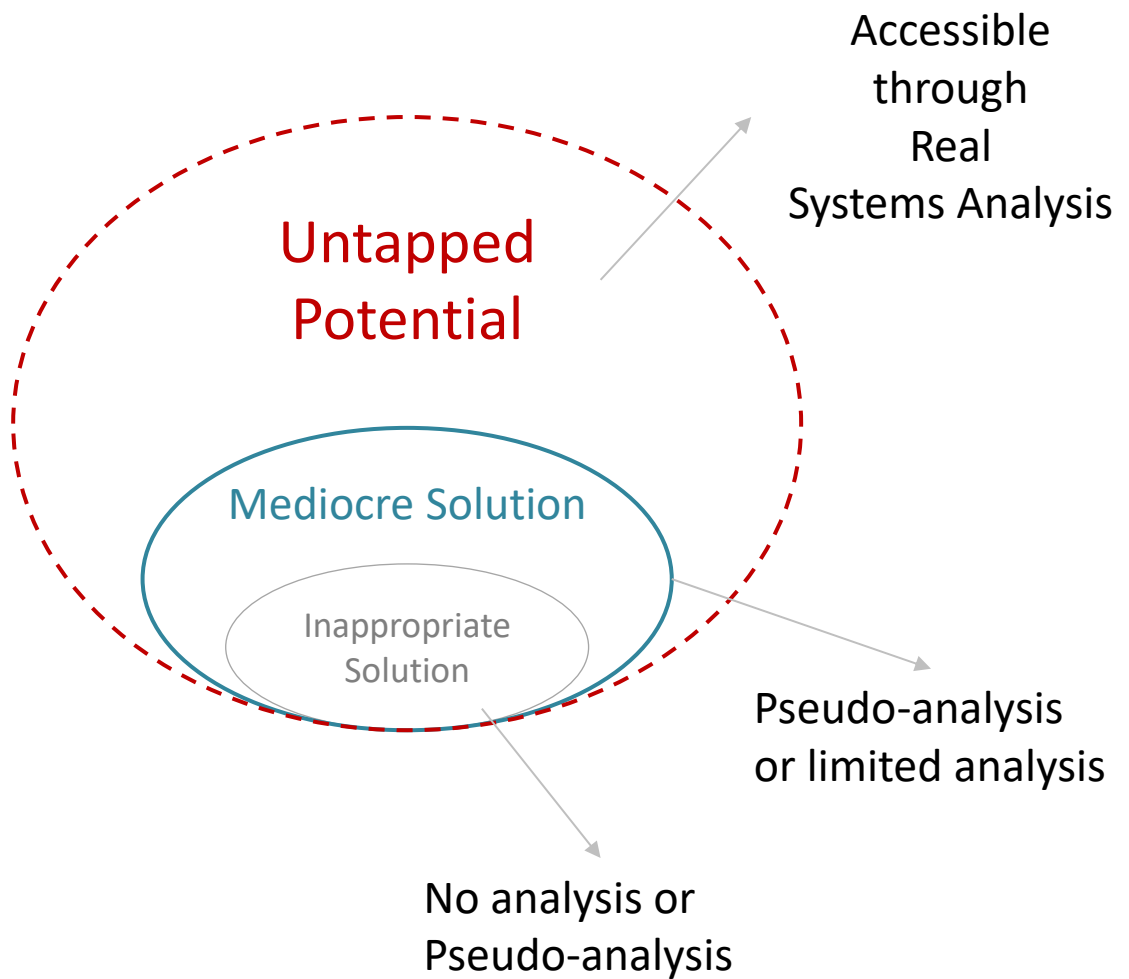
Systems Analysis **is NOT** and **CAN NOT** be reduced to a **REDUCTIONISTIC approach** (= looking inwards; studying the parts and their relations).

The design of systems operating in a company **requires a fully HOLISTIC approach**.



Appropriateness of a Solution

14/02/2020



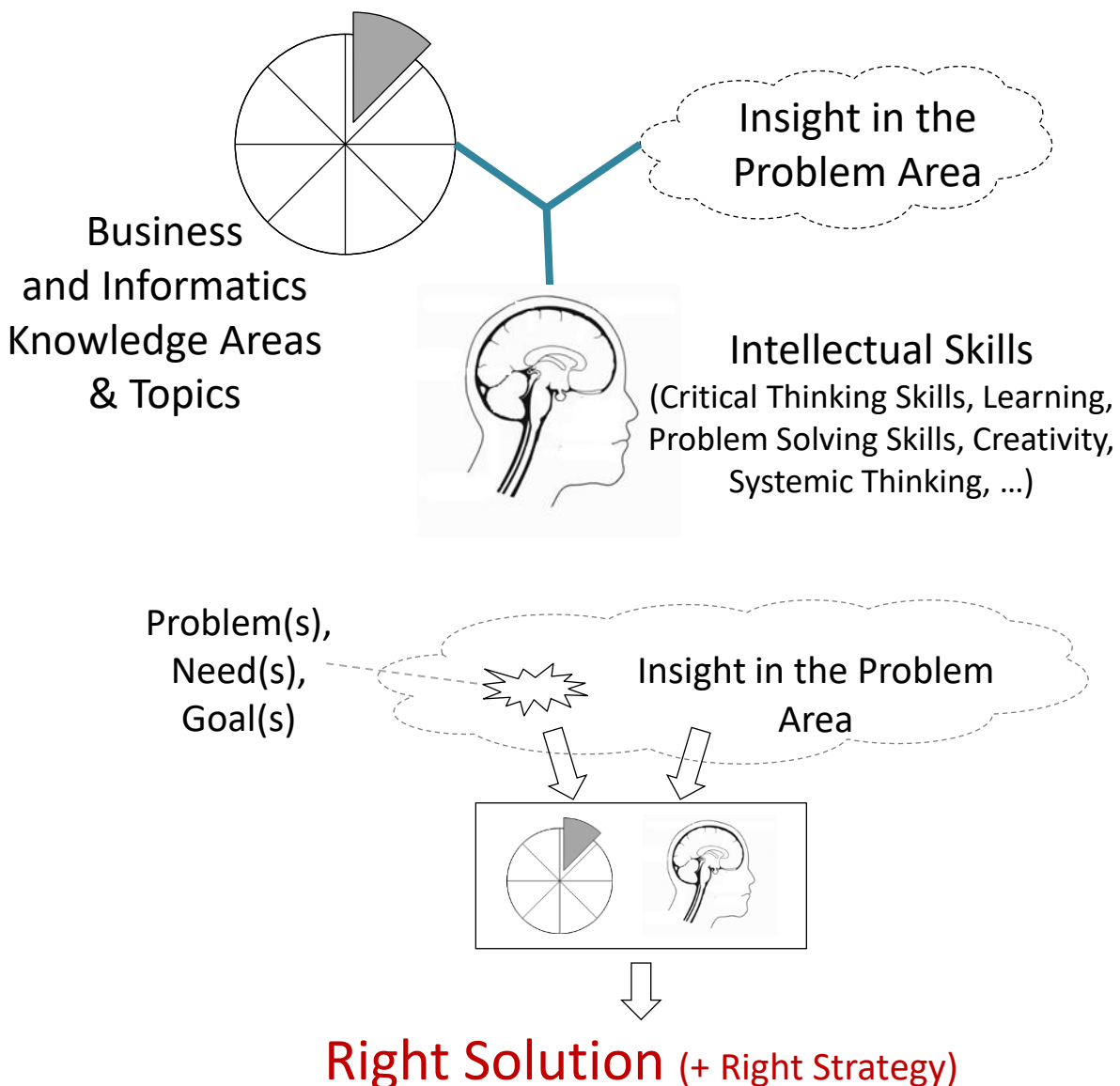
Core of Systems Analysis

10/01/2020

- **All** necessary **knowledge areas**,
- A correct identification of the **problem** (real cause(s)) and **goals**
- a correct and **profound insight** in the problem domain
- and advanced **intellectual** skills

are **critical**

- to **conceive** a right and well-designed solution
- to **evaluate** a solution
- and to be **aware of** and **grasping** the **possibilities** and the **(un)tapped potential**.

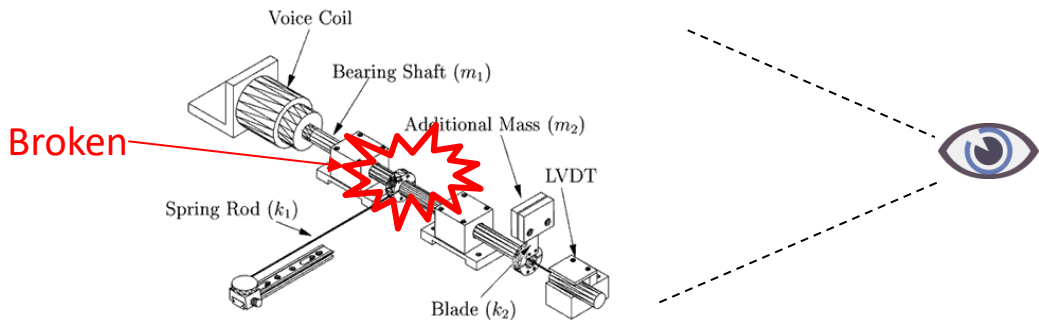


Concrete or Abstract Systems

31/12/2020

Mechanical System and Software Systems different in nature.

Mechanical Systems



A mechanical system is concrete and visible. A problem can often be resolved locally. It's often easy to estimate whether a solution or change on the system will have a local effect on the system or global effect.

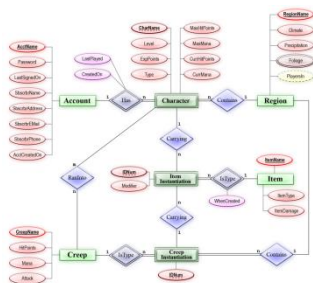
Software Systems



```
/**
 * Simple HelloButton() method.
 * @version 1.0
 * @author john doe <doe.j@example.com>
 */
HelloButton()
{
    JButton hello = new JButton("Hello, wor
    hello.addActionListener( new HelloBtnList

    // use the JFrame type until support for t
    // new component is finished
    JFrame frame = new JFrame("Hello Button"
    Container pane = frame.getContentPane();
    pane.add( hello );
    frame.pack();
    frame.show();
}
```

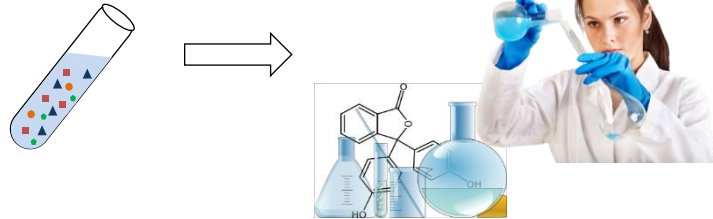
Software systems are abstract systems. They are represented by source code. The source code visible to a developer at a moment in time provide only a very limited view on the system. Human memory is rather volatile and unreliable. Therefore it is very difficult to know whether a logic is coherent with logic elsewhere, whether a logic on one place is necessary because of a need elsewhere, whether pieces of logic are reused or shared, etc.... A change at one place may cause problems elsewhere in the system. It's risky to limit one's focus, understanding and thinking to the area directly surrounding the problem.



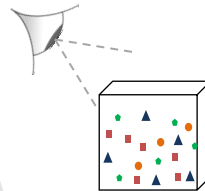
- It is risky to deal with software systems with the same philosophy, the same thinking and approach as with mechanical/concrete systems.
- Models significantly enlarge this view.

Major Widespread Misconception about Analysis

Analysis of sample



Looking inside, study the constituents, looking at details



The Analyst **must be told what** information, information solution, information system and/or software application the Business Stakeholders **want or need**.

Practically: the Analyst has to receive something to analyse. Business stakeholders have to give something to the analyst to analyse, usually this is what they want, what they need, what they ask, their requirements, ...

“The Analyst has to receive something to analyse. “

“Can you *analyse* my *Demand Requirements Document*”






NOT at all aligned with
the TRUE PURPOSE of Systems Analysis

Raison d'être - Main Purpose of Systems Analysis

10/01/2020

The Analyst may receive a mission, an objective. Or, Analysis may happen as a continuous activity.

The Analyst

- determines the environment, the area, the subject, the matter and the perspectives to be studied.
- determines the approach and what is required to perform the work.
- works as a radar scanning the environment;
 Radar
- as an investigator gathering information;
 Investigator
- as a physician making a diagnosis;
 Physician
- as a researcher finding solutions
 Researcher Innovator
- as an engineer conceiving those solution.
 Architect / Engineer

DEVELOPING SOFTWARE SYSTEMS

WITHOUT

(genuine) SYSTEMS ANALYSIS

30/10/2018

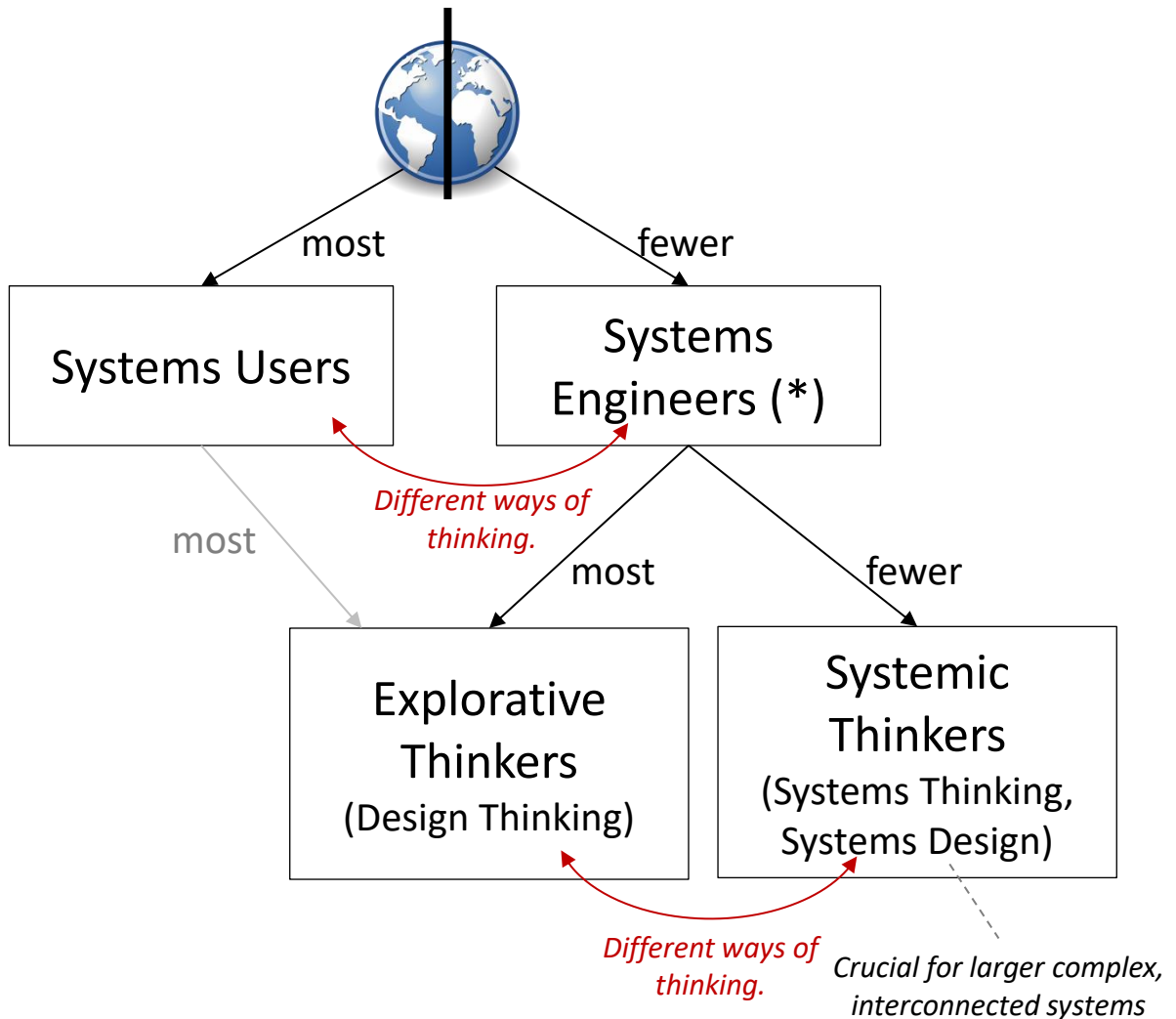
- Solving the wrong problem
- Solving partial problems or solving problems partially
- Building “solutions” solving nothing
- Symptoms and consequences are solved
- Conception of inefficient solutions
- Creation of new problems elsewhere in the organisation and/or in the longer term
- Waste of time and resources in figuring out and in building non-solutions, inappropriate solutions or in solutions that put a burden on future developments and on the future of the organisation
- Critical insight is acquired only after software applications have been built and/or implemented, leading to a lot of rework.
- Creation of little value, of no value or even systems detrimental for the company
- Little exploitation of the available resources, the possibilities and the capabilities
- Critical insight is not built
- ...

Notes:

If these complaints exist even when practising Systems Analysis, something is fundamentally wrong. (Double loop learning?)

Number of systems in our society increases dramatically.

- Increase in small, atomic connected systems (system-like components)
- Increase in integrated, connected more complex systems



Important for recruitment people, for assigning them on the right project, for collaboration and for evaluation.

Notes:

(*): Systems Engineer is an expert in systems. (S)He conceives systems. (This meaning differs from the job function known in IT environments).

Two Very Different Worlds

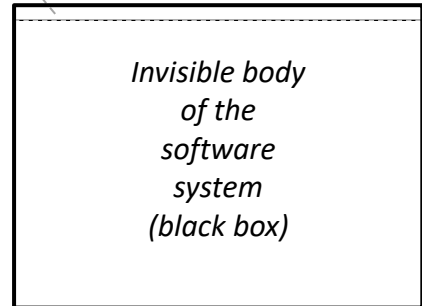
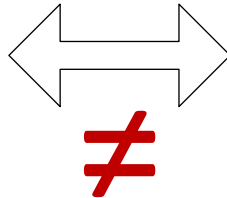
14/02/2020

Real World



*User Interface
(visible part)*

Software Systems



- “Known” by most
- The world we live in
- A lot is visible
- A part is concrete and touchable (material)

- Abstract - Immaterial
- Hyper-formal
- Hyper-detailed
- Complete
- Highly Logical
- Fixed (relatively)
- Invisible for users and business stakeholders (except UI & print-outs)

The two worlds are different in nature, in required competencies, in beliefs and in way of thinking.

A common mistake is to assume software systems can be understood and conceived with just habitual real-world knowledge, and without appropriate knowledge and insight in software.

You get what you created

Yesterday's solutions created today's problems

AND

Today's solutions will create tomorrow's problems.

“Good” is “not good enough” and “not good enough” is appalling.

A system works, behaves and delivers accordingly to its own capabilities, to its own logic and to its own characteristics.

Badly conceived systems deliver bad results and produce more adverse effects (limitations, difficulties, inefficiencies, other problems, ...).

AND

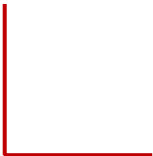
Well-designed systems dealing with the right issues and fitting in their environment deliver good results.

Solutions have to do so much more than solving only today's problems.



FOUNDATION

CORE IDEAS



The term “Analysis”

30/10/2018

ANALYSIS is:

- about **studying** the parts and their relations – looking **inwards** into something
- a project **phase**
- a branch, a group of tasks, in a **WBS**
- a **phase** in an System Development Lifecycle (SDLC)
- a specific **type** of activities
- a **process**
- a **method** or technique
- a **set** of activities (that anyone can perform)
- a set of **deliverables** (documents and models) resulting from analysis activities (“the analysis”)
- a **professional** discipline

All of these meanings are correct.
The meaning depends of the perspective, the context.

A single word having different meanings creates confusion.

What is “Business” ?

- The business community ?
- A part of the company or organisation ?
- Business knowledge, business logic, business expertise, field experience, ... ?
- Something under the authority and responsibility of the business community ?
- Anything emanating from the business community ?
- Anything the business community deals with or should deal with ?
- Everything that is not IT ?
 - A way to indicate matters and issues IT don't want to deal with (not wanting to take responsibility over it, non interested in it, time pressure, ...)



The Term “Business”

The label “**Business**” has two meanings

1. “Business” as **activities** deployed by the company directly related to a business domain.

2. “Business” **related to the Business Domain**

Business domain is the specific knowledge domain linked to the purpose or to the main functions of the company.

Examples: insurance, manufacturing, energy, accounting, marketing, production planning, ...

Beware ! This label is very confusing !

It does NOT mean “executed by”, “experienced by”, “emanating from” or “established by” or “managed by”

the Business Community (BC)

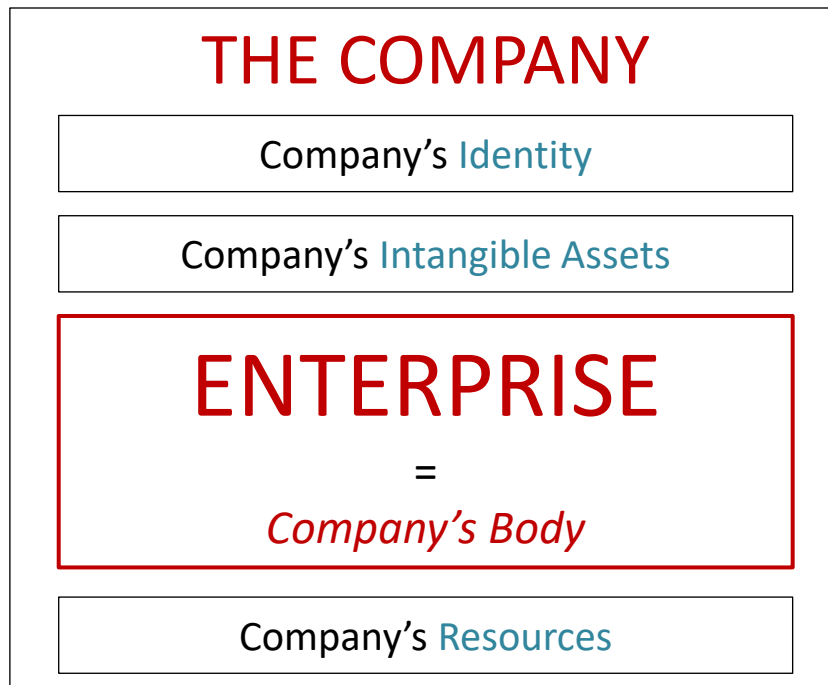
And does not imply ownership, responsibility or authority

Not all problems experienced by the BC are *business* problems

Not all processes executed by the BC are *business* processes

The Term “Enterprise”

30/10/2018



There is a need to differentiate different concepts:

Company: Commercial and non-commercial organisation

Organisation : Usually the human organisation (in ARB, the term ‘organisation’ may indicate a company or a non commercial organisation)

Business: the activities deployed by the company or a field of expertise

Enterprise: the body of the company, the system performing the company's activities essentially comprising of people, buildings, systems, machines, processes and infrastructure

This way of defining the terms allows to distinguish more easily different parts and concepts

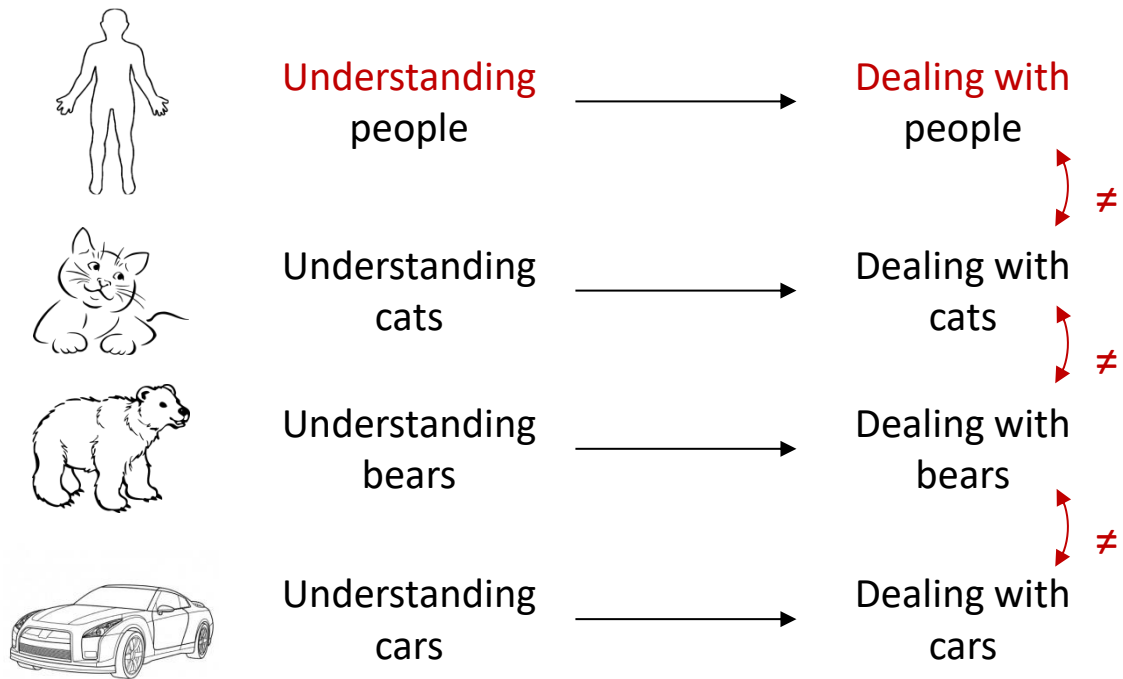
Notes:

(human) **organisation** is part of the **enterprise** is part of the **company**.

The company is the supra-system of the Enterprise.

Heterogeneous inter-organisational systems having the same nature as the company's body may also be defined as an 'Enterprise'. They require same knowledge and skills to be engineered.

Commonly: company = organisation = enterprise = business. In TARB, these terms are used for different concepts.



The way (the 'How') we **deal with a system** has to correspond with the **nature of that system**.

Understanding the nature of systems
is required
to deal correctly with systems.

AND ...the way we **deal with a system** has to correspond with the **nature of that type of systems and with the particular system**.

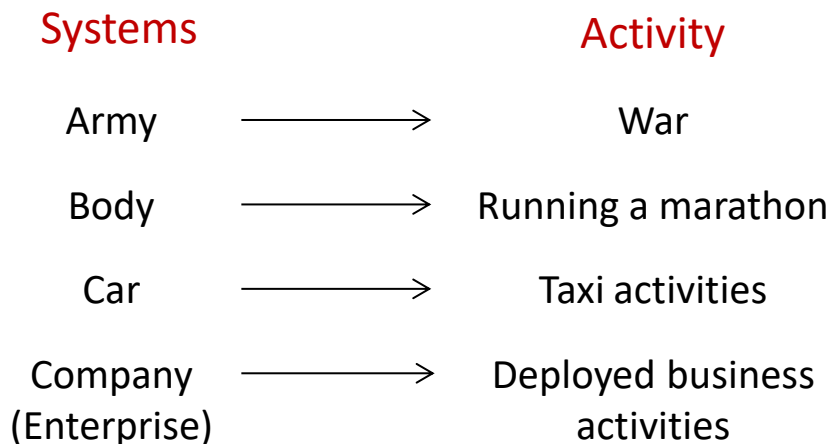
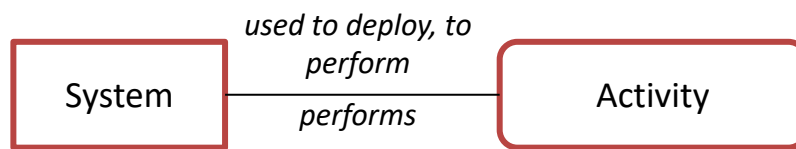
Notes

This is rather a profound principle worth more reflecting about it.

Understanding the "nature of systems" should not be confused with understanding a specific system (usage, behaviour, capabilities, logic and structure of a specific system)

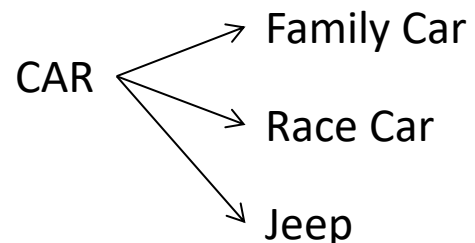
A **system** is a set of regularly interacting or interrelated entities that form a unified whole and which exists for a period of time.

Many man-made systems are created with a purpose in mind, to perform a function, to achieve a goal and used to perform/deploy activities.



“Same” system, but different ...

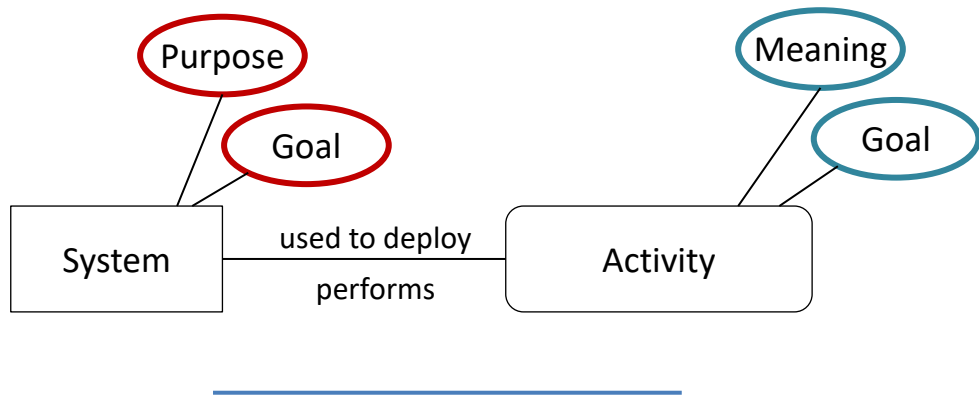
- **USAGES**
- **ENVIRONMENTS**
- **PURPOSES**



Not all natural systems have a function or goal. Human body has functions and goals (surviving). Climate system doesn't.

Differentiation between a System and its Usage

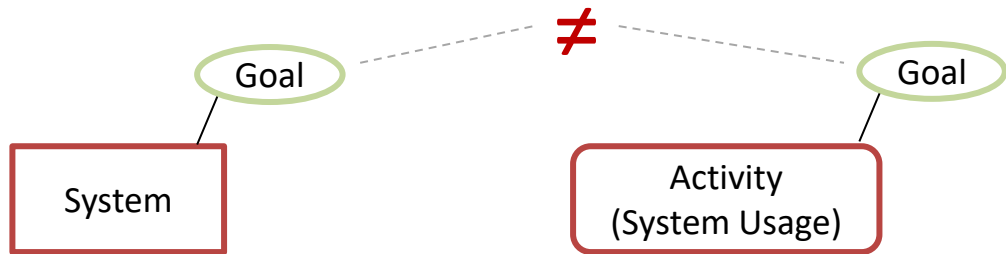
30/10/2018



- A system has a purpose and, through this purpose, it plays a role in an environment and in a situation.
 - The purpose of a car is to transport a limited amount of people and smaller goods over the road or over land. It should be able to drive on certain types of roads of different qualities, in different weather conditions, safely and for several years and for 150, 200 thousand kilometres or more.
- A system performs itself activities
 - Example: A car drives. Its engine runs.
- A system can be used to deploy activities
 - Example: A car (system) is used by the taxi driver to deploy taxi activities, to transport people to where they want.
 - Note: a car is a system; but a car + driver is also a system
- Activities have a meaning, a significance. They are performed with a certain intent, for a certain reason, to obtain an expected outcome, to reach a goal.
 - The taxi-company should make profit.

Goals of a Man-made System and Goals of its Usage

17/09/2019



A system has a goal:
"SELF-PRESERVATION"



Activities are executed with
an intend, to get a result,
to reach a goal

Long term

SURVIVAL
PROSPERITY

Growth
Continuing existence
Maintaining increasing the
system's qualities &
environmental fit

Short(er) term

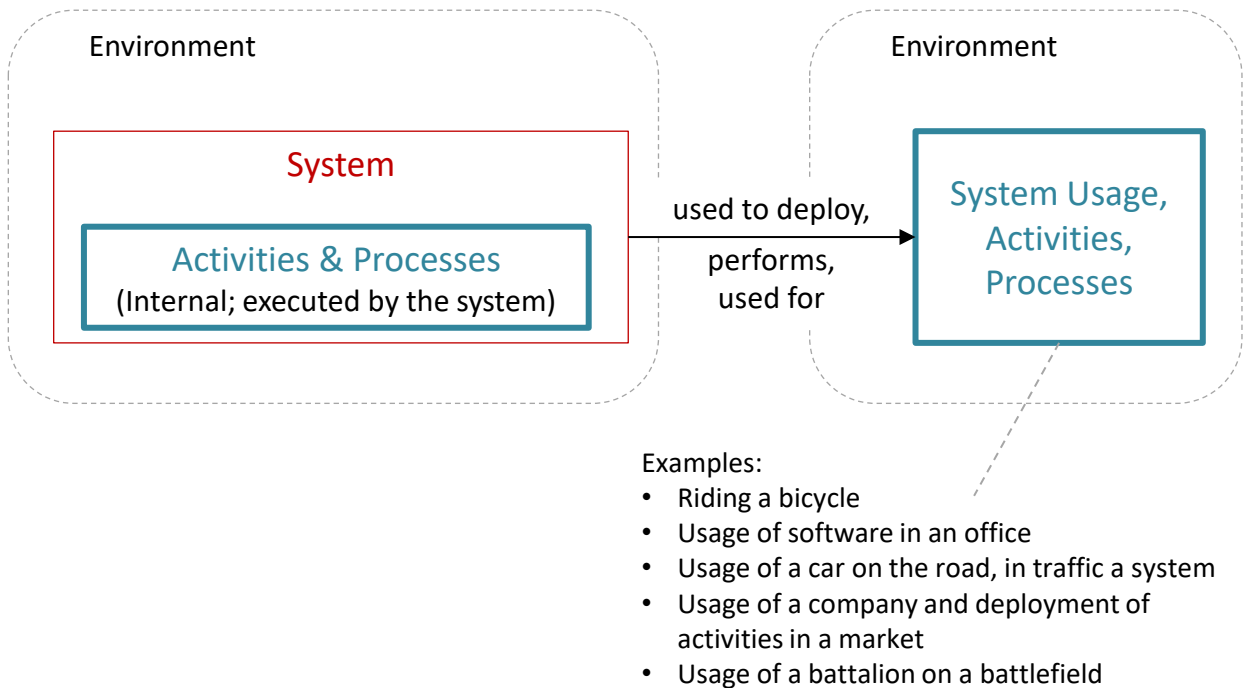
Successfully execute activities
Obtaining a specific result

If the long term has a low priority or is ignored, then
there is no solid foundation to build upon,
let alone to build something sustainable.

Anything threatening the long term systemic goals, undermines the
system's chances for survival and sustainable prosperity.

Reflection:

- Can a person deploy activities with a system, if that system is weak or broken?
- What will happen if the focus is only on activity results while the system's objectives are ignored?
- Who is interested in system's objectives and who's is much more focussed on the activity objectives?
- Which type of objectives is the most important? Why?



This is a basic model (simplified).

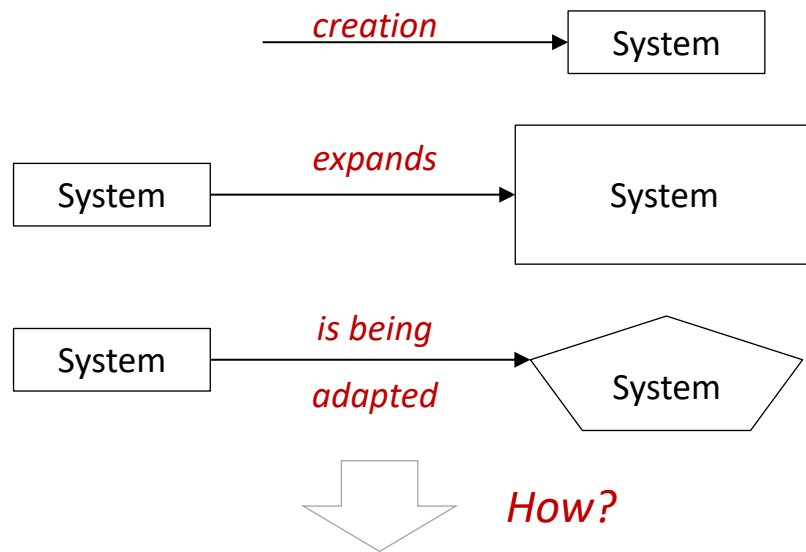
Later we'll see that

- Systems are often parts of supra-systems
- Systems often contain sub-systems
- System's environments may be common to different systems.
- The environment of a system is a subset of the environment of the supra-system.
- The system can be used by people. A system may contain people performing activities in, of and for the system.

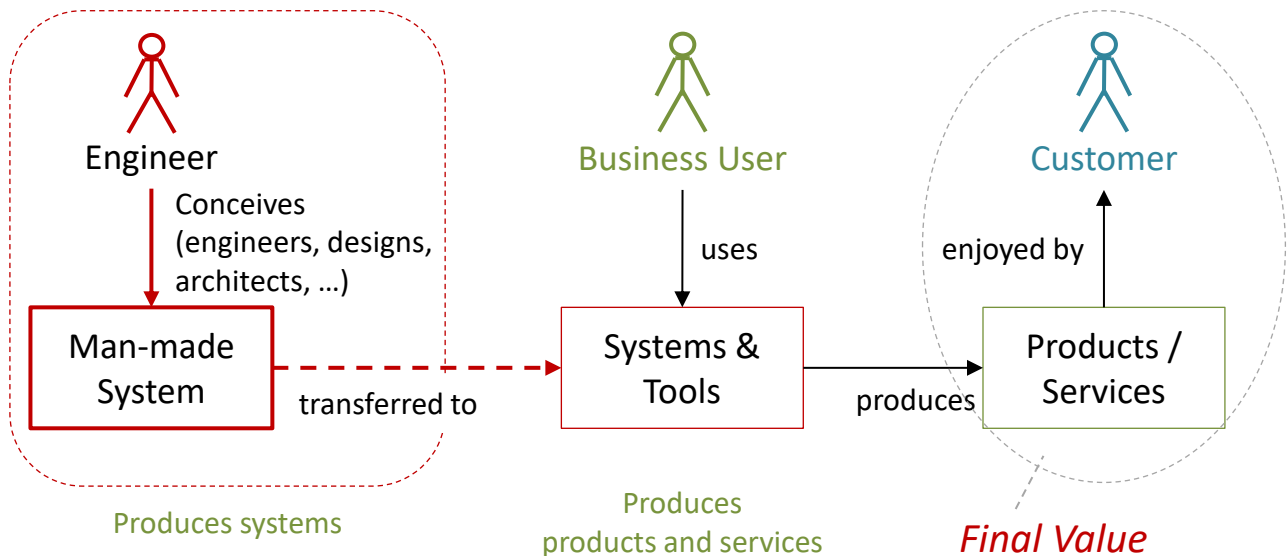
and much more.

Not all systems can be used in a same way as man-made systems and/or tools are used (users).

Example: eco-system, water cycle, climate system, tidal system, ...



1) Development of Production Tools



Final value is the final customer having his needs or desires to be satisfied.
Everything that precedes in the chain are means to achieve this.

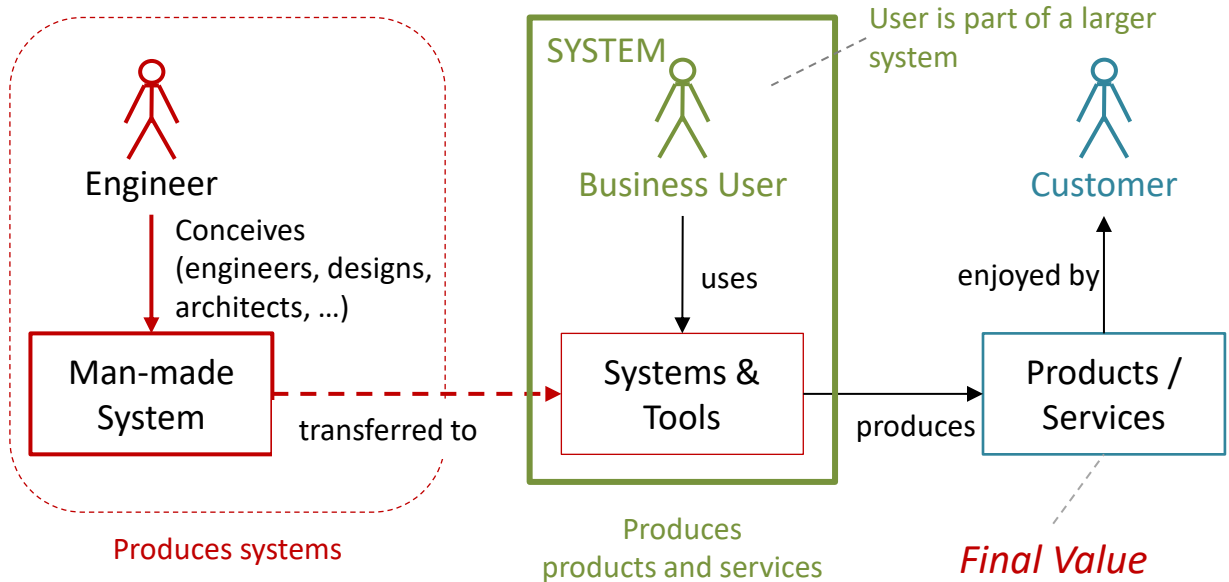
Notes:

Engineer, user and customer are roles. They may indicate a group of people or a person.
A person can have different roles.

This model is not a collaboration model. It doesn't represent the collaboration. This model doesn't exclude or prevent it. Collaboration is critical and advised.

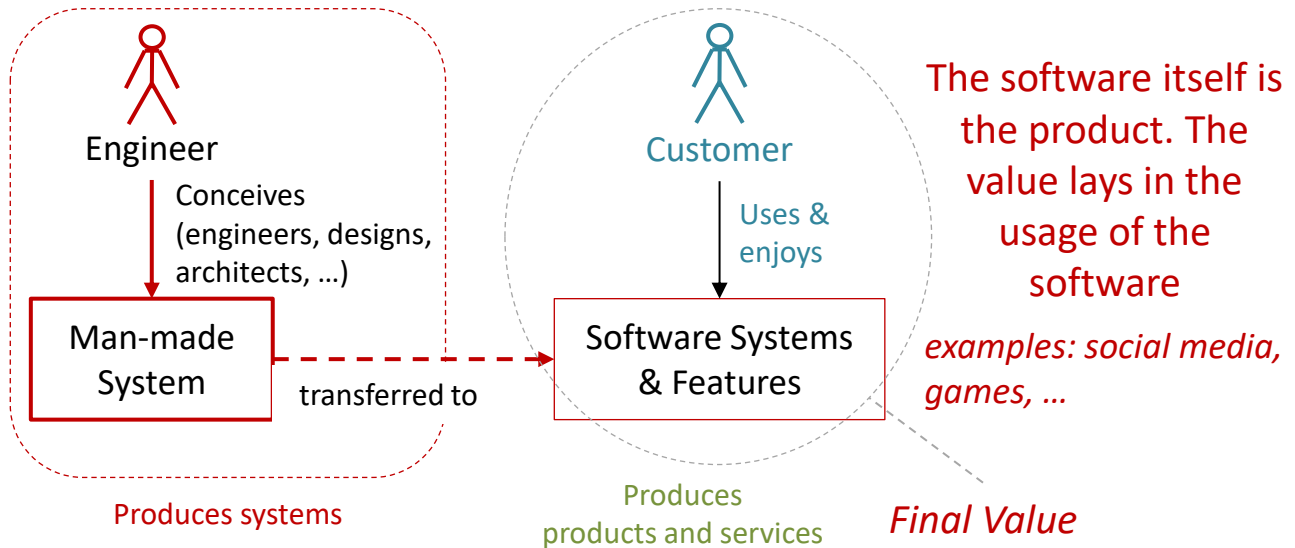
2) Development of a Production System (Enterprise, Company)

Considering not just the business user, but also the system (s)he is part of



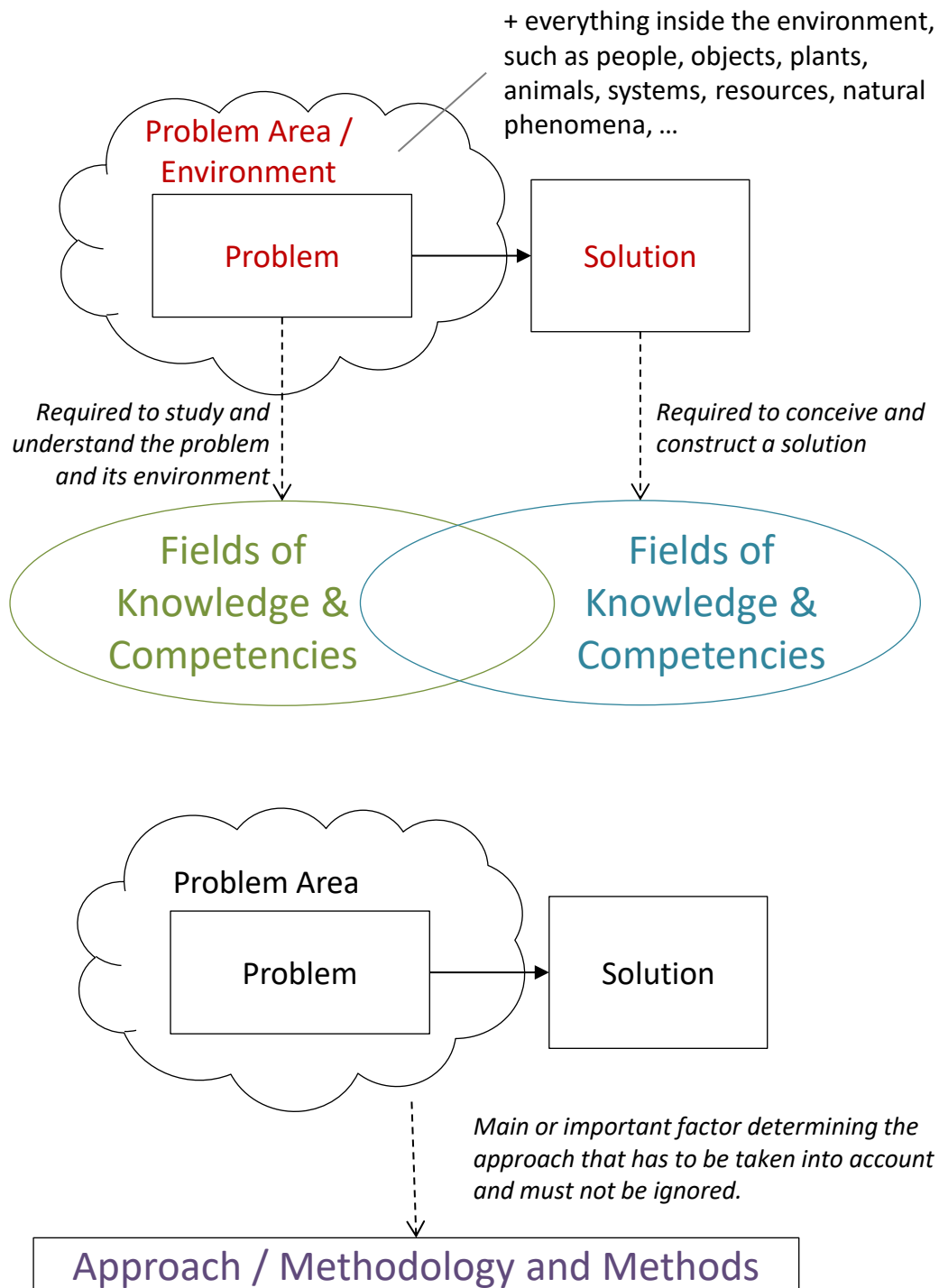
3) Development of Software as product for Final Customers

Considering not just the business user, but also the system (s)he is part of



These are 3 cases of different missions or 3 different considerations, with different goals and value.

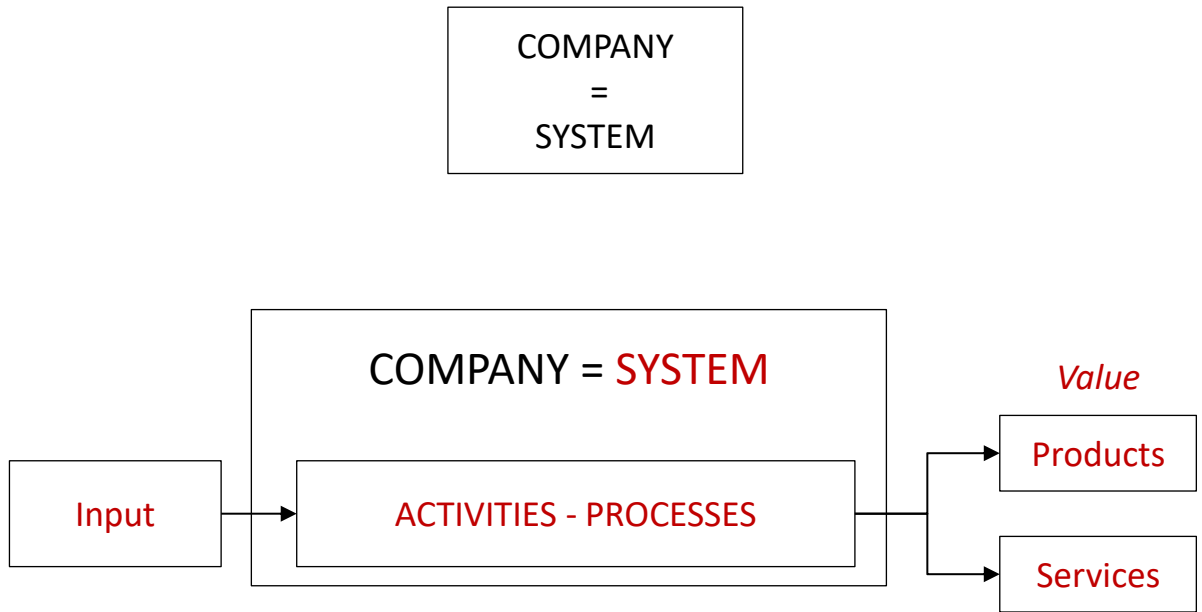
They lead to different Software Development Approaches !!!



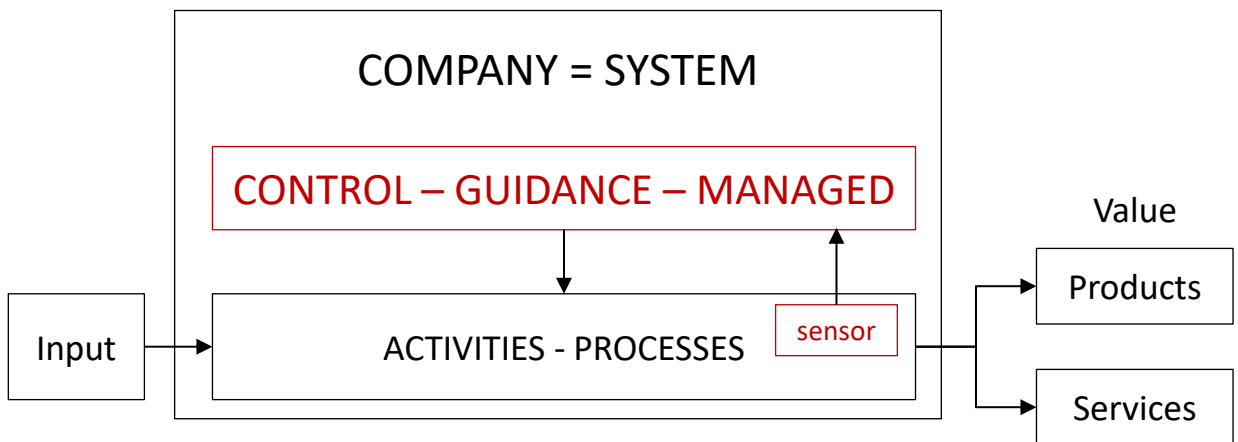
Company

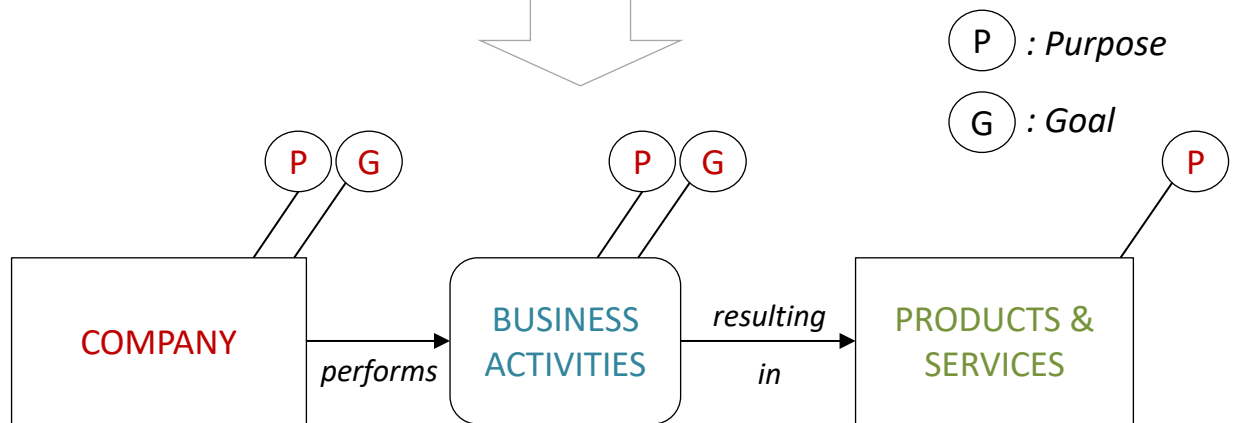
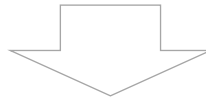
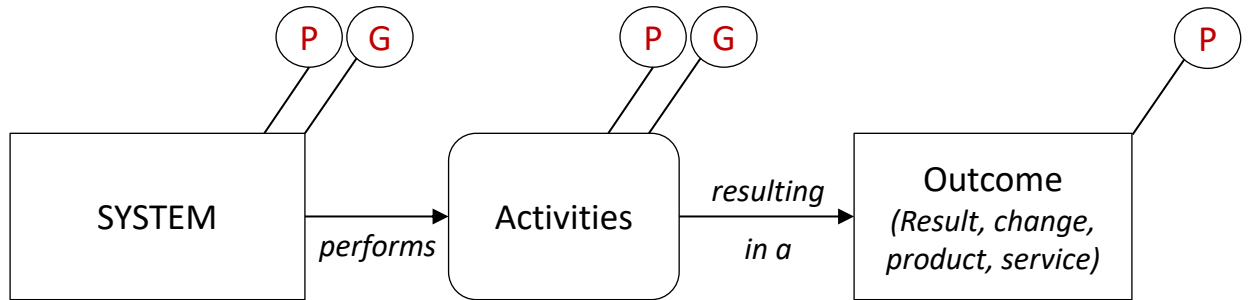
17/09/2019

A **COMPANY** is a **SYSTEM**



A **GUIDED, CONTROLLED & MANAGED** SYSTEM





Purpose

- Serving people or society within its domain of expertise

Goals

- Survive
- Prosper
- Grow
- ...

↓ determines

Structure, functions, logic, behaviour, characteristics, capabilities, ...

≠

Purpose

- Creating and delivering products and services

Goal

- Market share
- Profit
- Volume
- ...

↓ determines

Structure, logic, characteristics, capabilities, ...

≠

Purpose

- Creating and delivering products and services

↓ determines

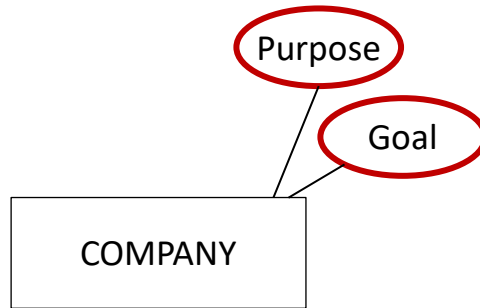
Structure, functions, logic, behaviour, characteristics, capabilities, ...

Notes

The purpose, as intended function critical to play a role, is the main element (but not the only one) driving the design. The purpose is (or should be) reflected in the design. It is embedded in it.

Company : Purpose & Goal

17/09/2019



PURPOSE: Creating and offering products and/or services

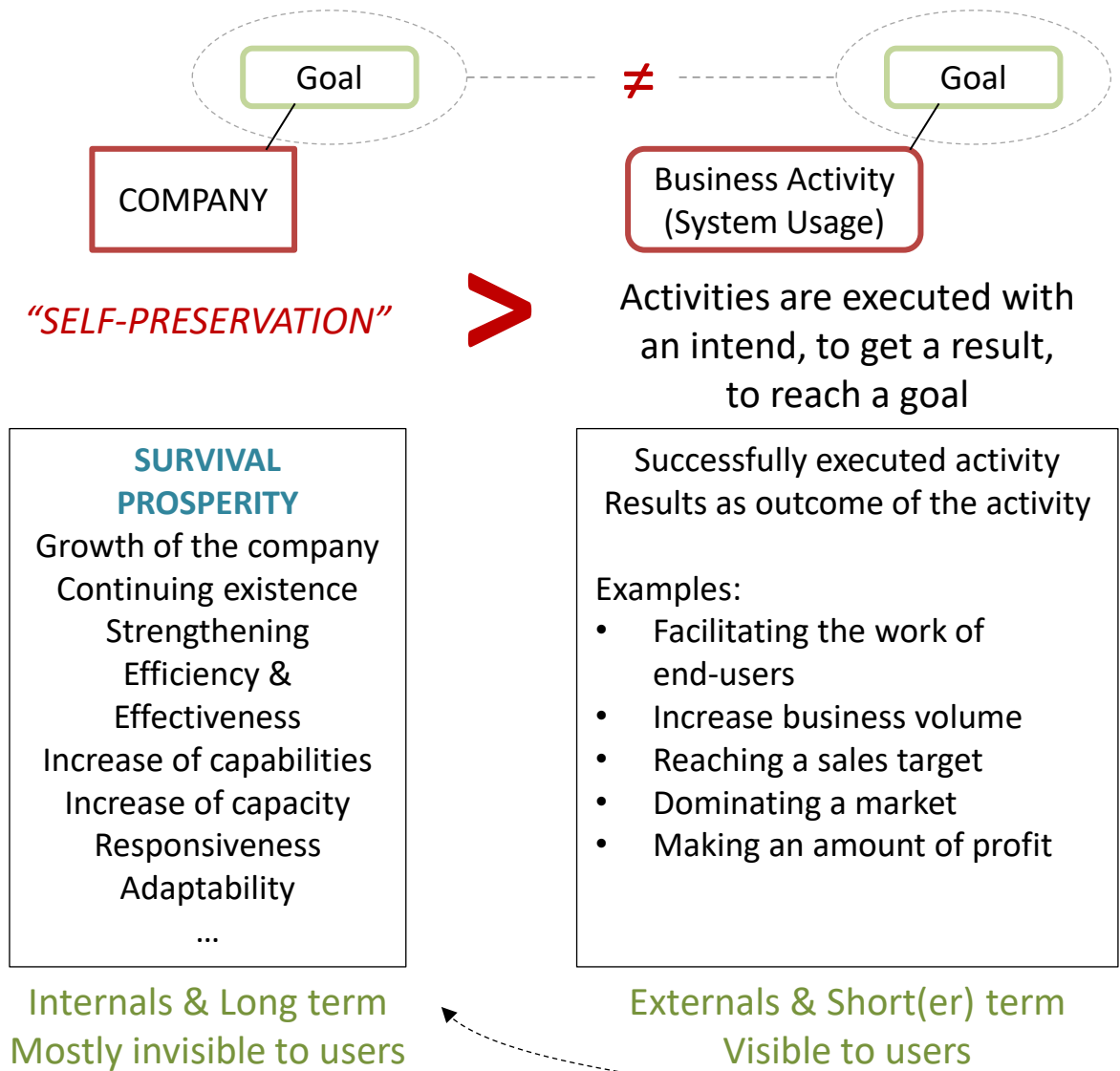
Long Term

Usual Company's **HIGH-LEVEL GOALS** :

- Survive - Staying alive
- Not be worse than ...
- Do Well
- Remain ahead of the competition
- Grow
- Realise its Purpose
- Materialise a Vision
- Executing a Strategy
- Be a Market Leader
- Become “something”
- Bring Innovation
- Thrive - Be Prosperous
- ... ?

- The purpose here is only expressed as a general statement. The real purpose is specific to each company.
- The HL-goals are expressed as possible, but general statement. A company should articulate them in (a) specific statement(s).
- A company can have more than one HL-goal.
- Implicit goals are often assumed and therefore not expressed. They are hidden and have to be brought to the surface.
- Long term goals have to be taken into account permanently, daily, even in small or short term decisions. Short term goals should be aligned with and contribute to long-term goals as much as possible.

Company's Goal vs Business Goal 17/09/2019

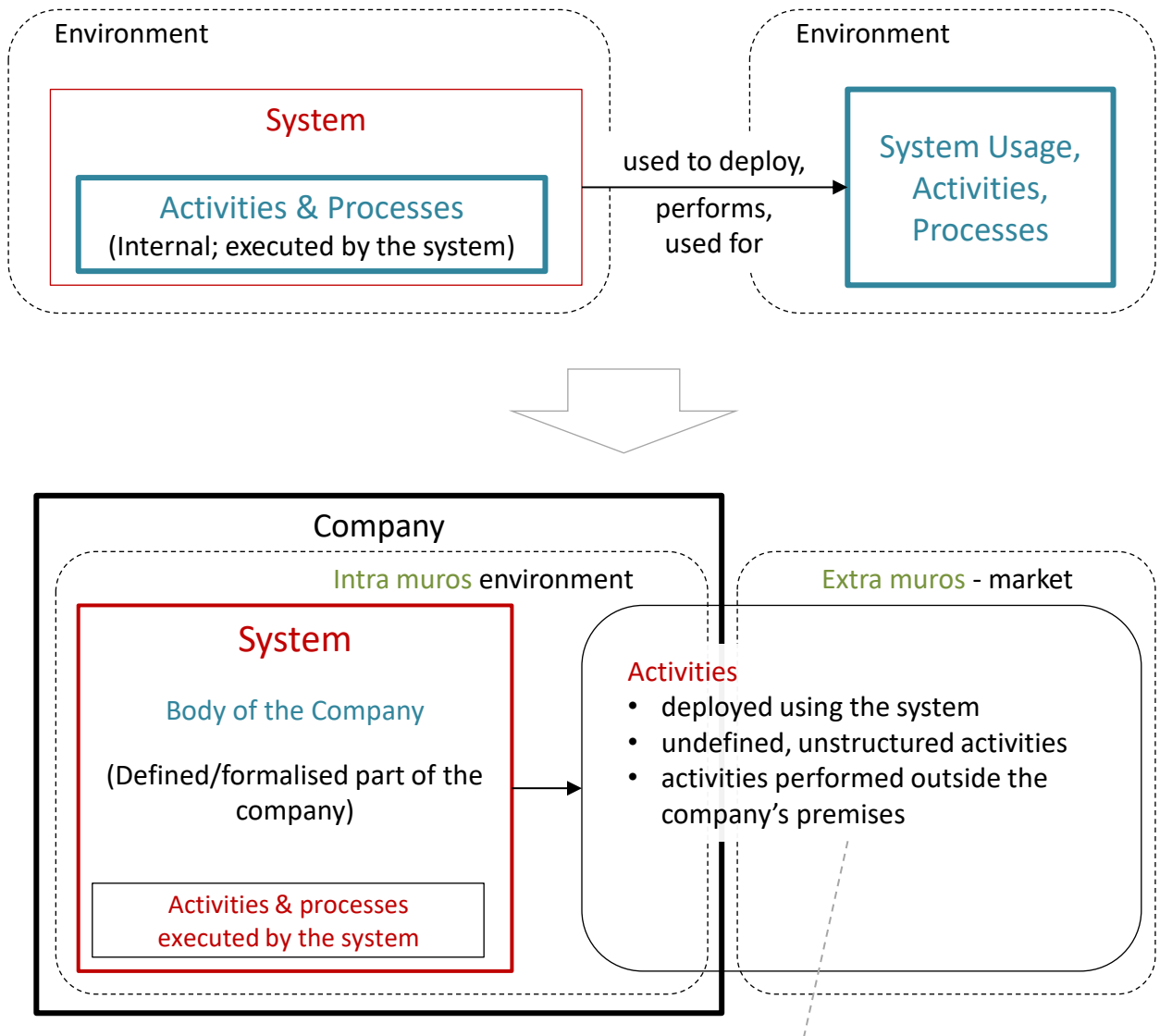


Systems, companies and societies have failed over longer period of time due to their inability to think on much larger time scales.

Even if activities and results are good,

- an abuse of the system
- and a weak system

endanger the sustainability.



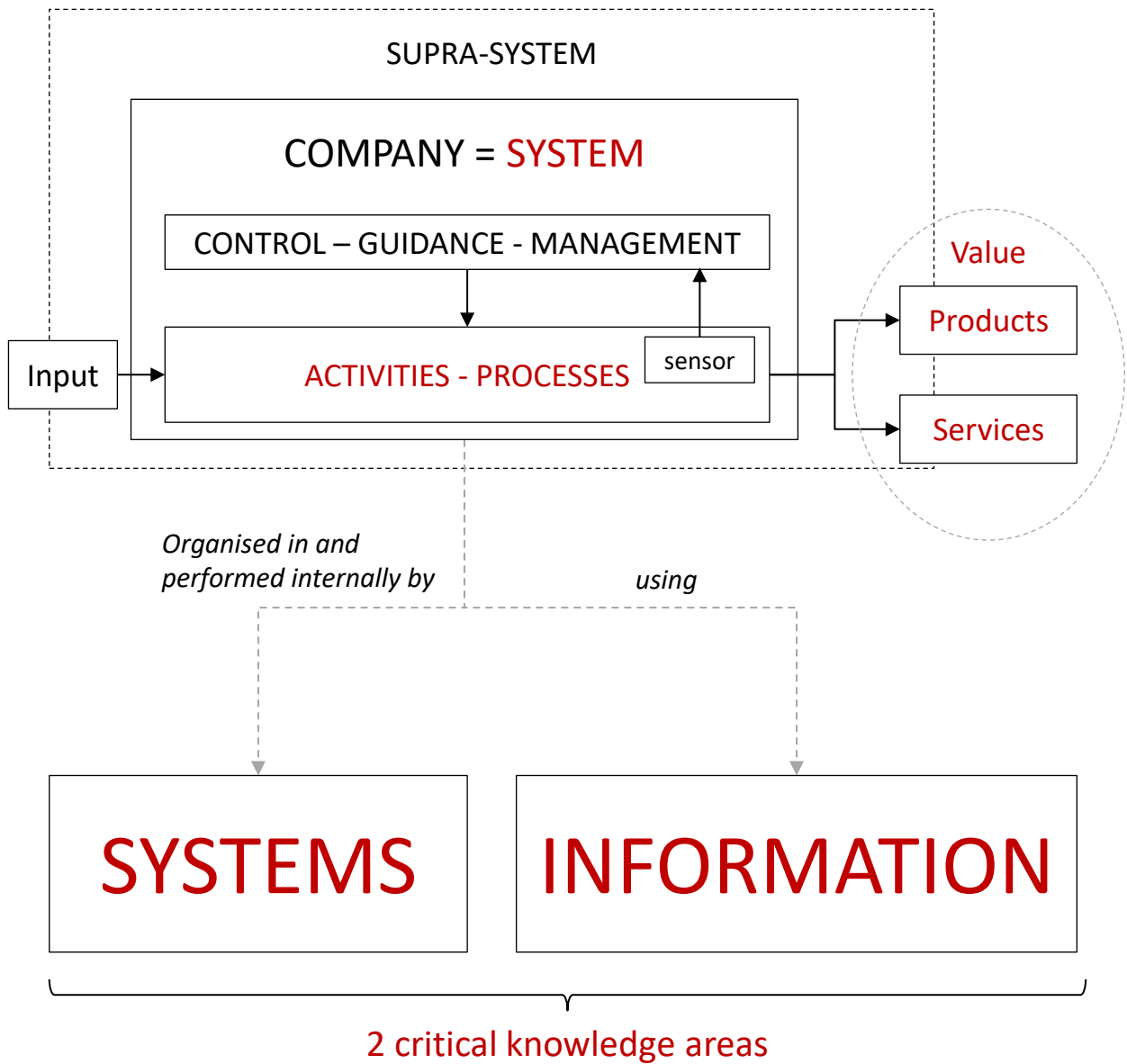
A Few Examples

- Meeting prospects, customers and (potential) business partners
- Spreading advertisement in the market
- Many Public Relations activities
- Many negotiations
- Many management activities
- Management decisions

Strictly spoken, some formal systems and processes may reach way beyond the company's premises, for instance, such as supply chains. For the sake of simplicity, it's not showed in the model. The core idea of intra-muros and extra-muros activities represented by the model remains valid.

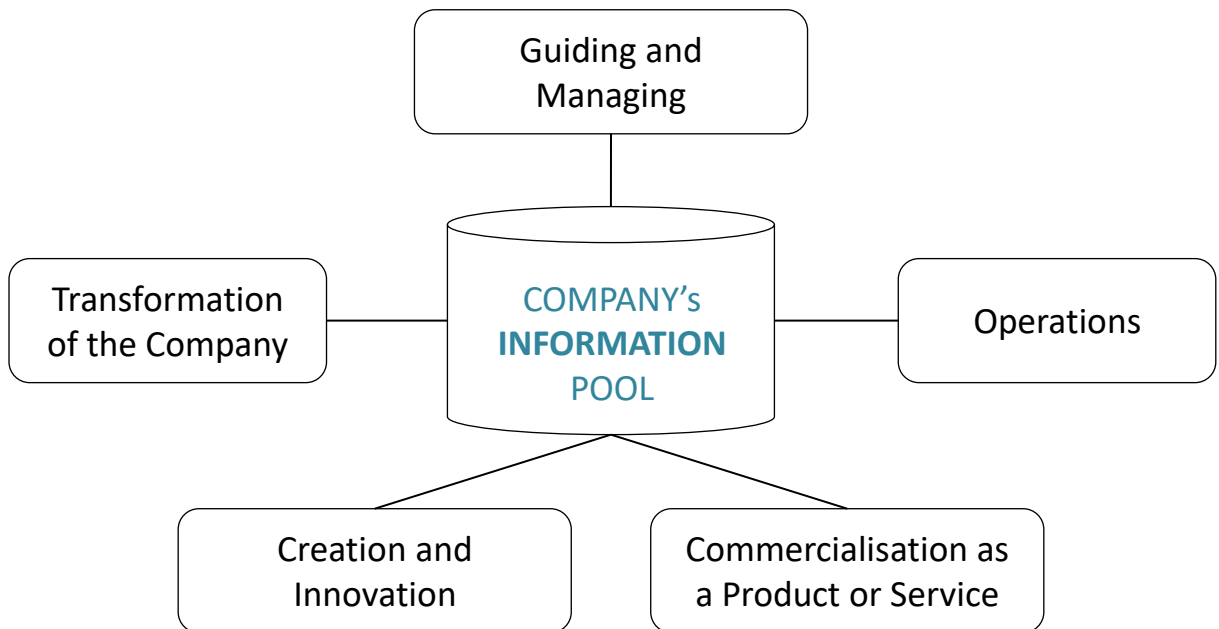
Company: Two Major Perspectives

17/09/2019



Information – A Resource

A COMPANY CAN'T FUNCTION WITHOUT INFORMATION



Information is a **CRITICAL RESOURCE**

Information allows a company

- to execute its activities (operations)
- to be guided and managed
- To create and to innovate
- to sell it as a product or as services
- to transform itself

Information: A Critical Resource

CONSIDER

- A company needs to capture, identify, connect, organise, control, store, process, manage, share, disseminate information (and much more)
- Information **pervades the company** and has **multiple usages**
- Information provides plenty of **opportunities**. It's **key to progress**.
- Information has **Value**. The value decreases and/or increases. It can be preserved and increased, but it can also be **destroyed**.
- Information can be captured or created → **AMOUNT INCREASES**
- We tend to create, add and duplicate, but omit to maintain, to verify and to clean up.
- Tendency to **ENTROPY** (natural tendency to evolve towards chaos).

The **SURVIVAL** and **PROSPERITY** OF THE COMPANY
depends on **INFORMATION**

Information must be **EXPLOITED**
Information must be **MANAGED**
in an OPTIMAL WAY

Therefore, the company needs and must be adapted and equipped to deal with information.

To take up this challenge,
the **Most Skilled People** are required

Information in Companies : RISKS

How is the company's information evolving?

Quantity



(probably)

Quality

Known ? Fluctuating

Value

Known ? Fluctuating

Organisation

.....?

Manageability

.....?

Exploitability

.....?

*Tendency to
Entropy*

Today:

Information Explosion
+
Communication Explosion

CHALLENGE : PREPARE FOR THE FUTURE

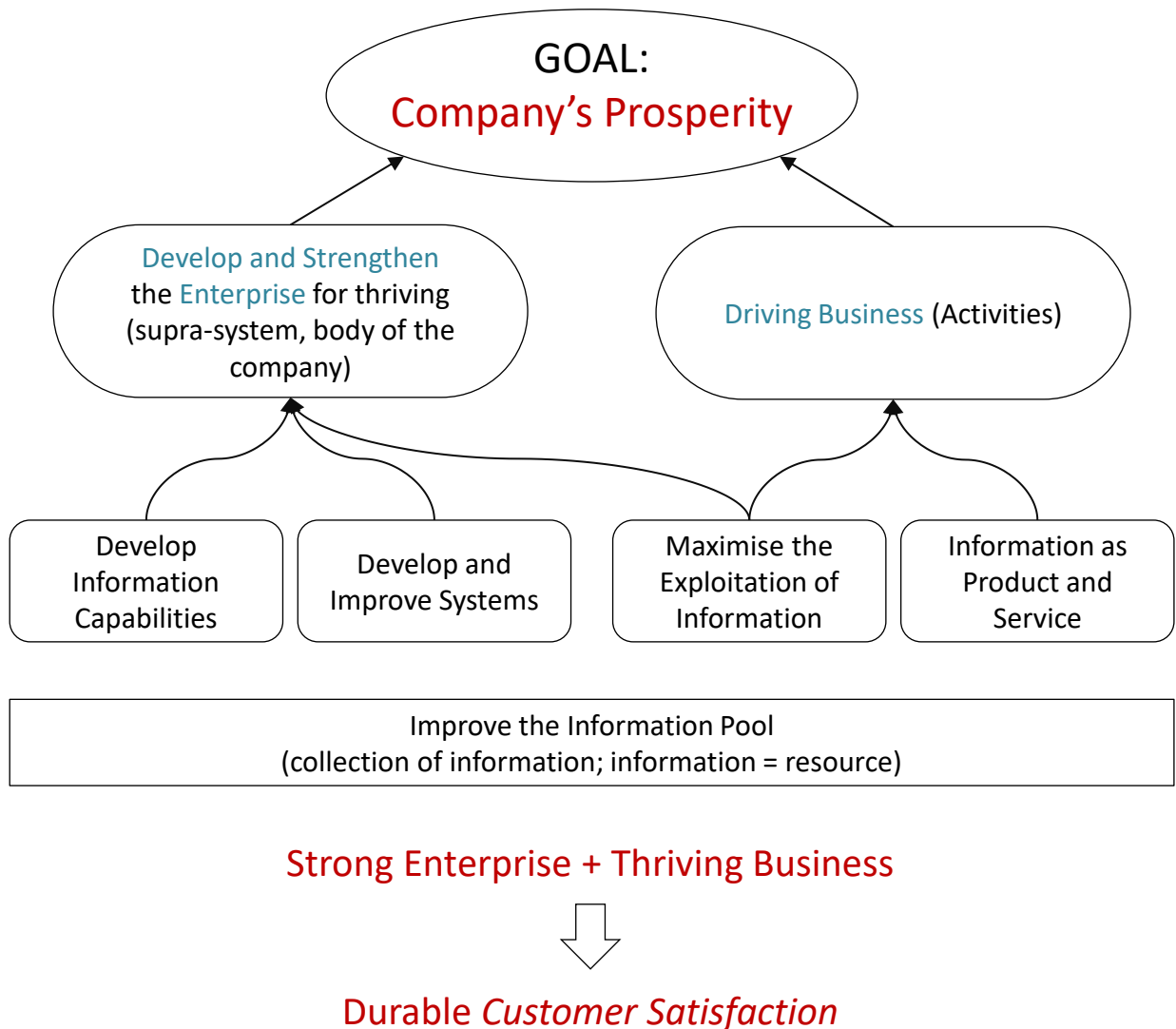
REFLECTIONS

Endless growth of information? Good or bad? To be managed? To be controlled?
How to deal with it? How to adapt systems to it?

Financial means are managed in a formal and strict way. Why should this not be necessary for information?

Global Objective Related to Information

18/12/2018



*Customer satisfaction **about systems** is an indirect goal, a desired consequence.
(more about it later)*

Notes:

How feasible is it to deploy profitable business activities with a weak enterprise? And over time?

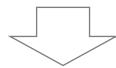
How feasible is it to improve profitable business activities with a strong and well-designed enterprise?

Is a strong, well-designed, fit-for-purpose system or product more able to create customer satisfaction than badly designed and weak systems and products?

Customer satisfaction is discussed more in details in TAR.B.

A COMPANY SHOULD

- Increase the **amount of useful/valuable information**
- Preserve and maximise the **quality** and **value** of information
- Maximise the **exploitability** of information



The **COMPANY NEEDS** an **OVERALL IMPLEMENTATION** allowing the company to perform these operations (see previous page) on information consisting of :

- Capabilities
- Processes
- Mechanisms
- Systems
- Structures
- Rules and Principles
- Information channels (flows)
- Concepts
- Competencies
- ...

Information
Component
(defined later)

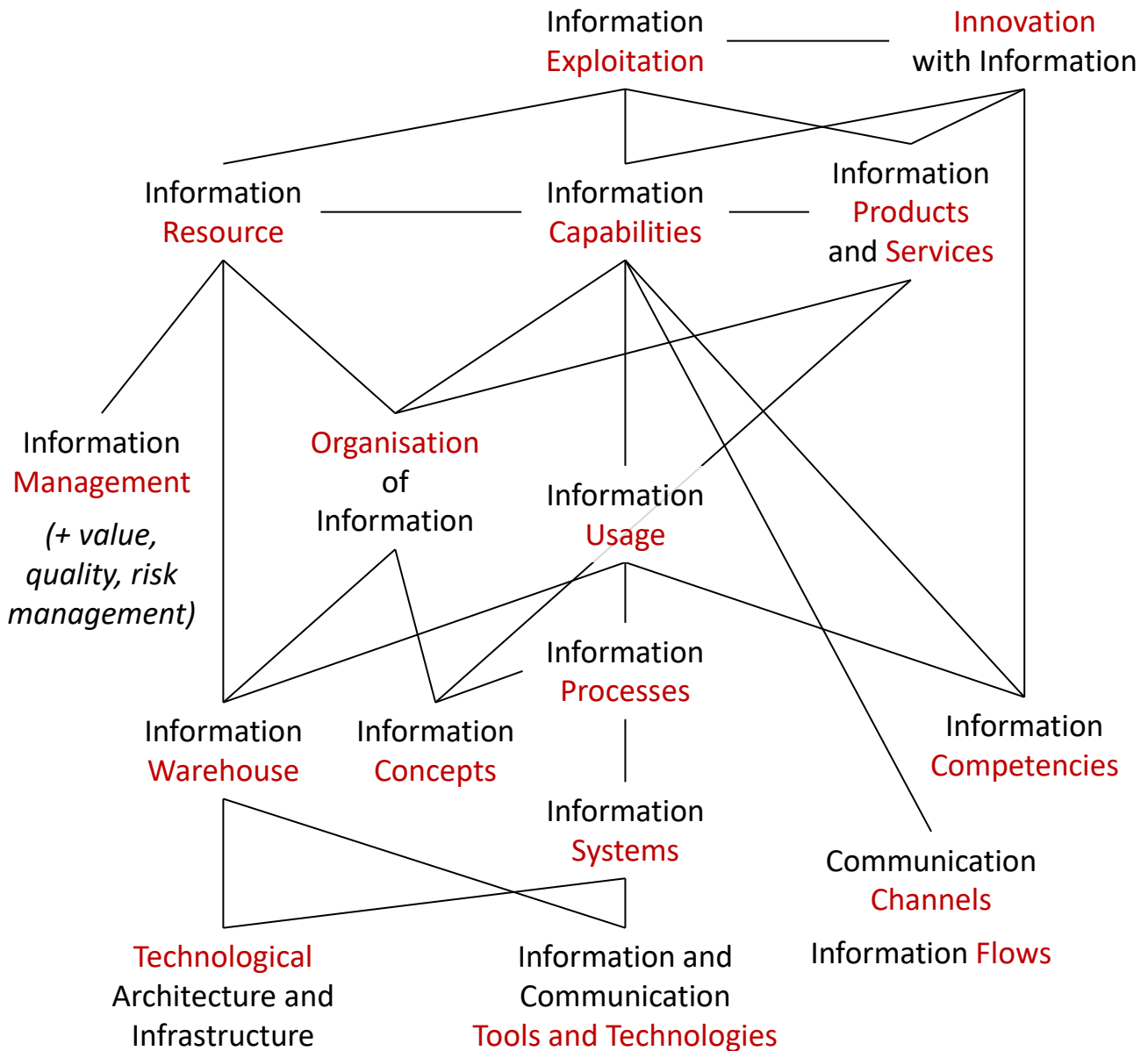
HOW TO DEAL WITH INFORMATION?

HOW TO CONCEIVE SYSTEMS AND
INFORMATION SOLUTIONS?

Company's Needs

30/10/2018

*What does an organisation need
to get the most out of information ?*



Explanation of the graph on next page.

All these aspects form a single whole in which everything is related. The grey lines represent only the most important relations.

A company (organisation) needs to be able to **exploit** information and to **innovate** with information.

Therefore, it needs the **resource** 'information', the **capabilities** to deal with information and to deliver information as a **product** or **service**.

As much '**information**' as possible has to be **organised**, **managed** and stored in a **warehouse** (the organised ensemble of all the means capable of storing information; not to be confused with a '*data warehouse*' – BI concept).

The information **capabilities** require **organised** information. These capabilities are defined by the potential of "**information usage**", which can be a real usage, a possible usage or an expected or intended one. This usage can be manual or it can be through defined information **processes**.

A **correct use** of information necessitates two competencies: subject matter knowledge and **information competencies**.

The execution of **processes** can be supported by **systems** or they can be executed, fully or partially, by them. **Communication channels** and **information flows** transfer information among different entities and systems.

Innovation with information is done through the information **capabilities**, more precisely, how the information is **organised** and **used**. And innovation can be realised through the offered **products** and **services**. Information **innovation** requires strong information **competencies**. Innovation in information is primarily achieved by filling in a need or by new concepts and new algorithms. Information **innovation**, information **concepts** and information **competencies** are actually strongly related to each other.

The concrete **Information Component** contains, among others, a technological **architecture** and infrastructure and information and communication **tools** and **technologies**.

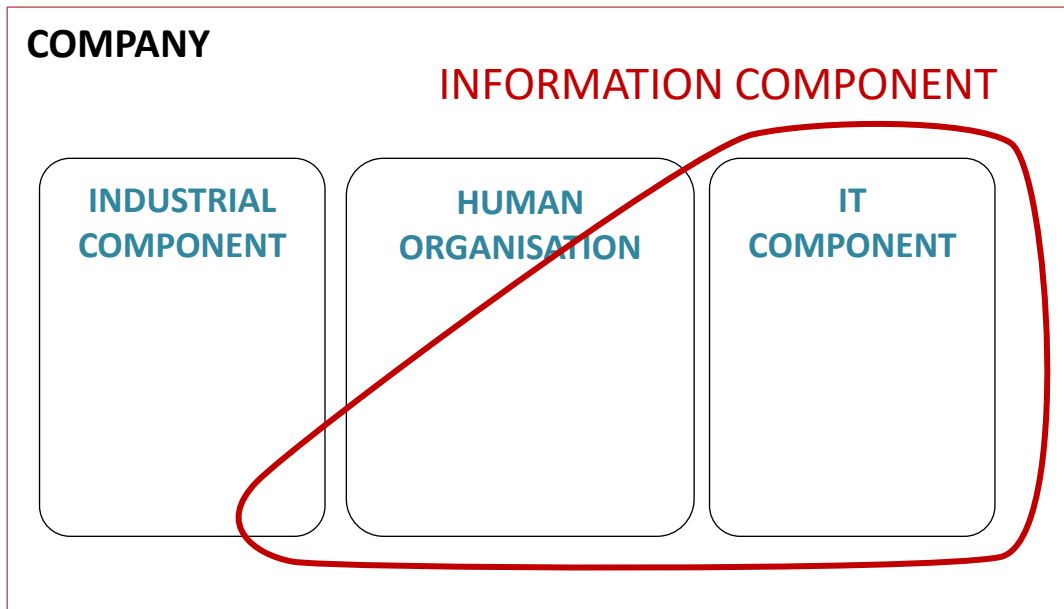
EXECUTING ENTITY	COMPANY			
	INDUSTRIAL COMPONENT	HUMAN ORGANISATION	IT COMPONENT	
	Way of execution	Automated	Manual	Automated
	Processing	Material, Energy	Material & Information	Information
Nature of processes / processing	Pre-defined processes	Interpretation, variable logic, ad hoc decisions, explicit and implicit knowledge	Pre-defined processes	

The **IT Component** is the ensemble of software systems and all computers and other hardware processing, storing or transmitting information.

Note:

Robots and people can handle information and matter.

This model represents a general view. It is an abstraction of reality, a rough classification based on some characteristics. In reality, these components are intertwined. There is an exchange between them. These components collaborate. They are aligned and to fit with each other. They share the same environment. They are not strictly separated.



The **Information Component** is the whole organised environment and means allowing the capture, the storage, the management and the exploitation of **Information**

The *information component* resolves the company's information needs. The information needs determine the information component. But this is the minimal viable objective for the short term. Continuously and actively seeking to maximise the exploitation of information, to innovate, keeping the internals of the information component right and a seamless integration are necessary for the long term survival and for prosperity.

Key Areas of the IC

- Information **Products**
 - Information **Services**
 - Information **Exploitation**
 - Information **Capabilities**
 - Information **Value**
 - Information **Resource**
 - Information **Usage**
 - Information **Systems**
 - Information **Processes**
 - Information **Concepts**
 - **Organisation** of Information
 - Information **Warehouse**
 - Information **Management**
 - Information **Risk** Management
 - **Innovation** with Information
 - Information **Competencies**
 - Technological **Architecture** and Infrastructure for Information and **Communication**
 - Information and Communication **Tools**
-

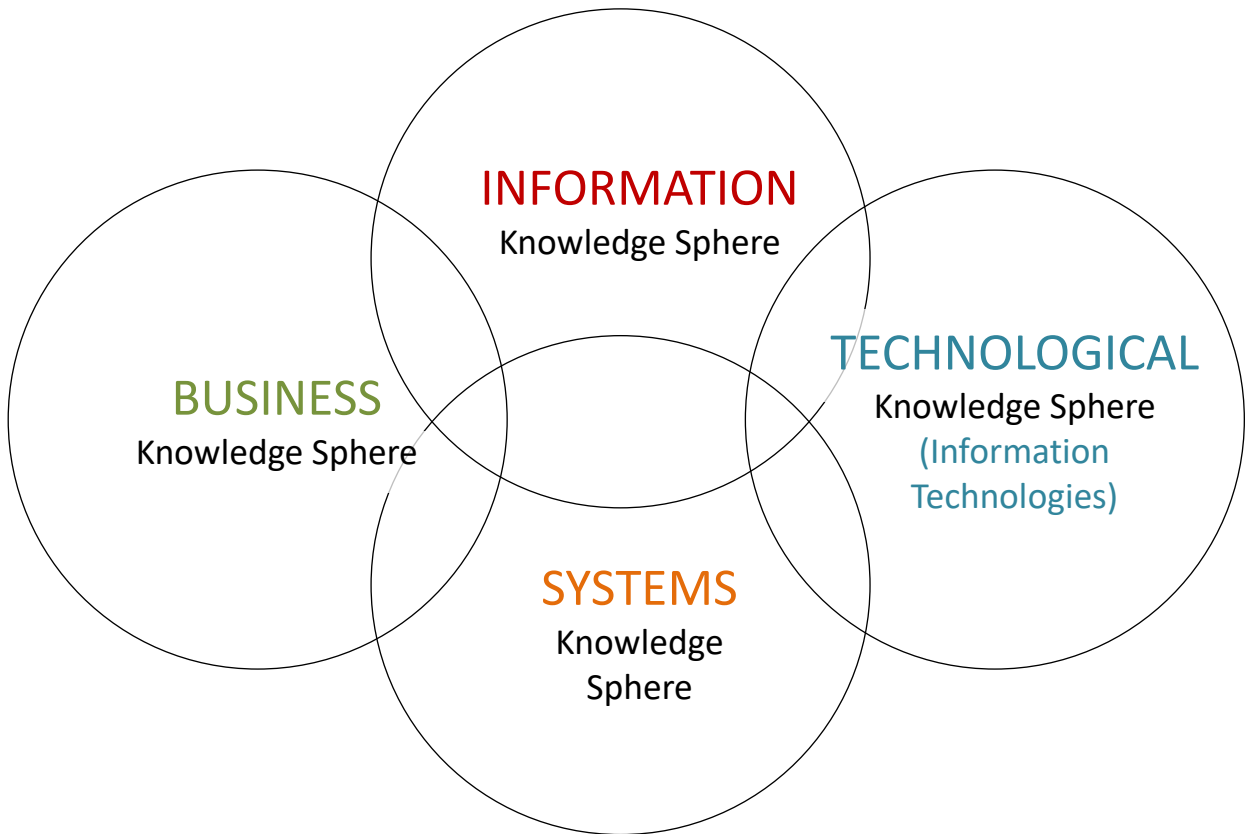
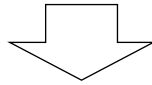
Notes:

The list is not exhaustive. It can be extended.

Key Areas Implemented in the Information Component

17/09/2019

INFORMATION COMPONENT



These knowledge sphere can be neither ignored
nor underestimated

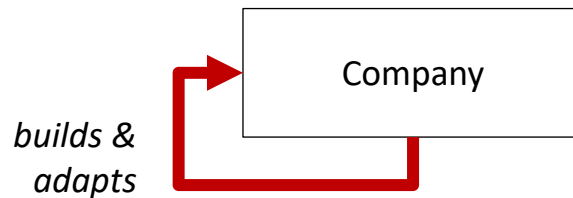
Notes:

Expertise in Organisational Behaviour and Development is another very useful knowledge sphere, particularly if organisational changes are part of the solution.

A Company

- is created
- grows
- adapts
- shrinks
- moves

Who transforms the company?



The company is a system that adapts itself

but not anyhow!

Notes

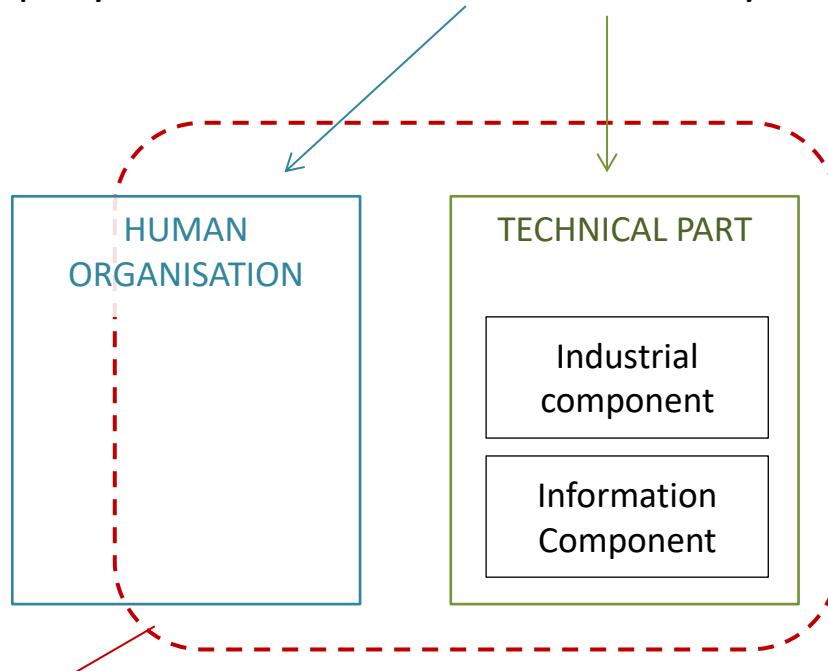
This doesn't mean that any component or element can adapt itself or that adaptation may occur freely or in any way.

A company can also merge, expand, split up, decrease in size, But this is, at this point, implicitly included in the shown cases.

A Company Can Be Engineered

30/10/2018

A company is a **man-made socio-technical** system



Engineerable part

If it is man-made and it can be represented with models, then it can be engineered.

A huge part of the company can be engineered.

- Some parts or sub-systems can be fully engineered.
- Other parts can be engineered up to a certain point.
- A company is a living system. This doesn't imply that all its parts are living systems as well. Some of its parts are complex and will evolve, but are not complex adaptive systems or not living systems.
- Some qualities, aspects or behaviours may still be unexpected, either because of lack of knowledge and thinking, because users are more creative than builders or because they were (really) unpredictable (should be very rare).

Importance of a Well-Engineered Company

Half (*) of
the results
determined by



- Engineered systems and processes;
- Appropriate, performant, reliable equipment
- availability of resources; ...

Other half(*)
of the results



- Culture
- Leadership
- Collaboration
- Communication
- Values
- Norms
- Attitude
- Interpersonal relations
- Social skills
- Competencies
- Motivation
- Intellectual skills
- ...



can inhibit
each other

Notes:

(*) “half” is figuratively.

A good leadership, culture, collaboration, etc. (right column) doesn't guarantee a excellent engineering and well-designed systems and processes (left column).

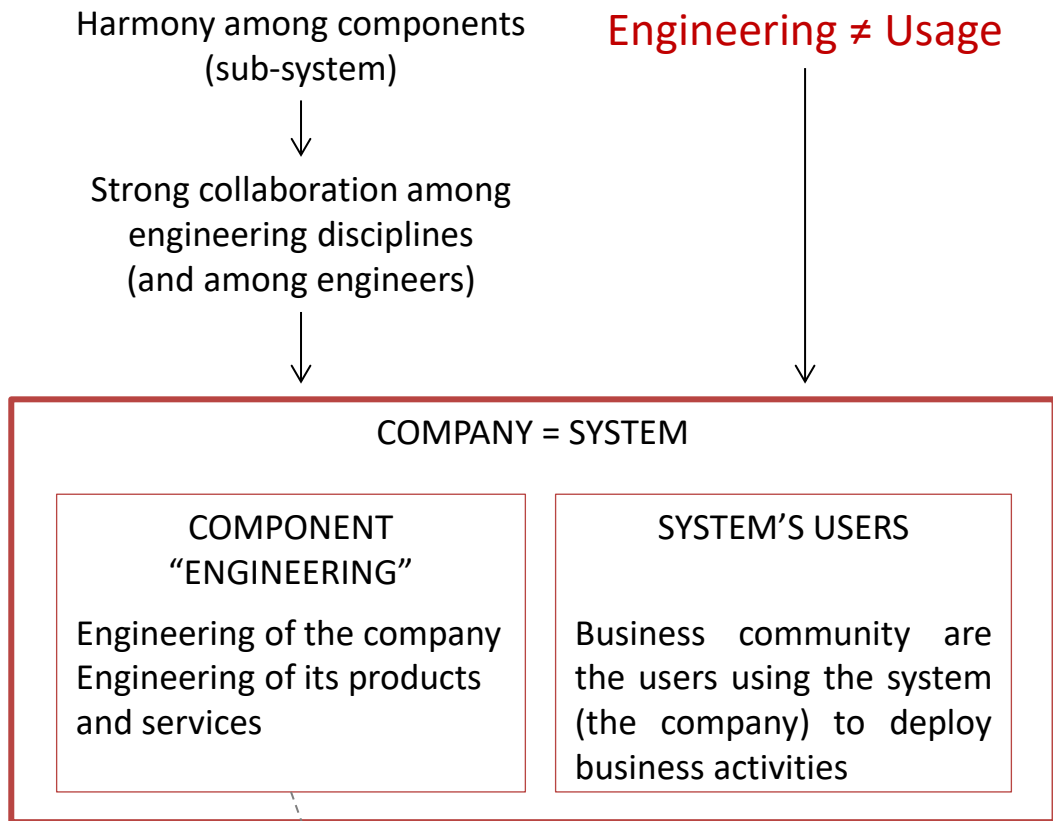
Psycho-social issues can completely prevent a good system to function properly.

An inappropriate organisation, system or process can block skilled and motivated people to deliver results.

A Company (Re-)Engineers Itself

30/10/2018

A company has to adapt. It has to **REINVENT** itself.
It defines its own structure, systems and processes.



Engineering Roles & Disciplines

- R & D
- Product Development
- Organisation Development
- Industrial Engineers
- All architect, analysis, engineering, modelling, developers roles of Informatics

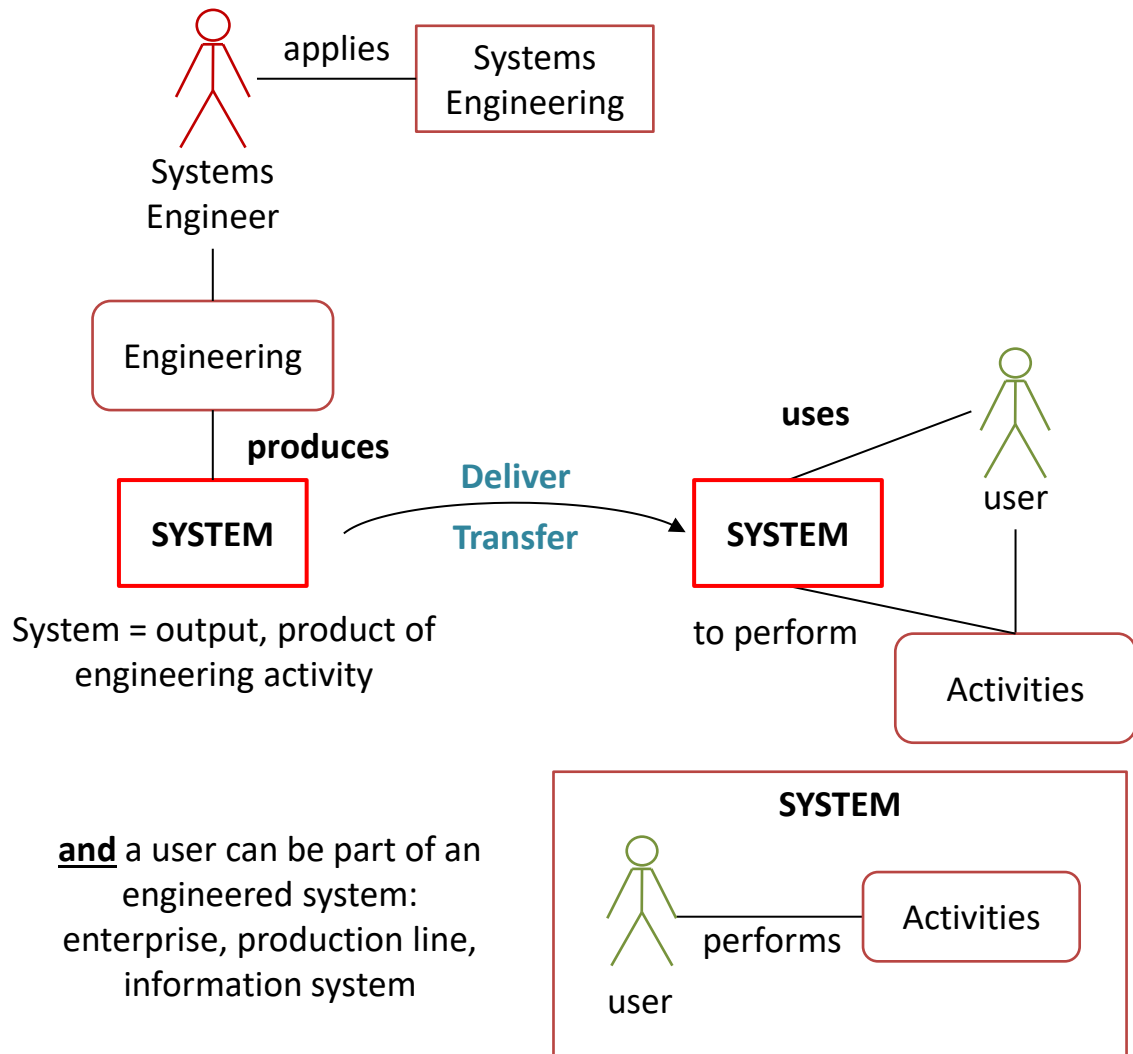
Note:

Collaboration with business community remains crucial as well.

Business people are part of the system (enterprise) and they use the system (enterprise).

From Conception to Usage of Systems

30/10/2018



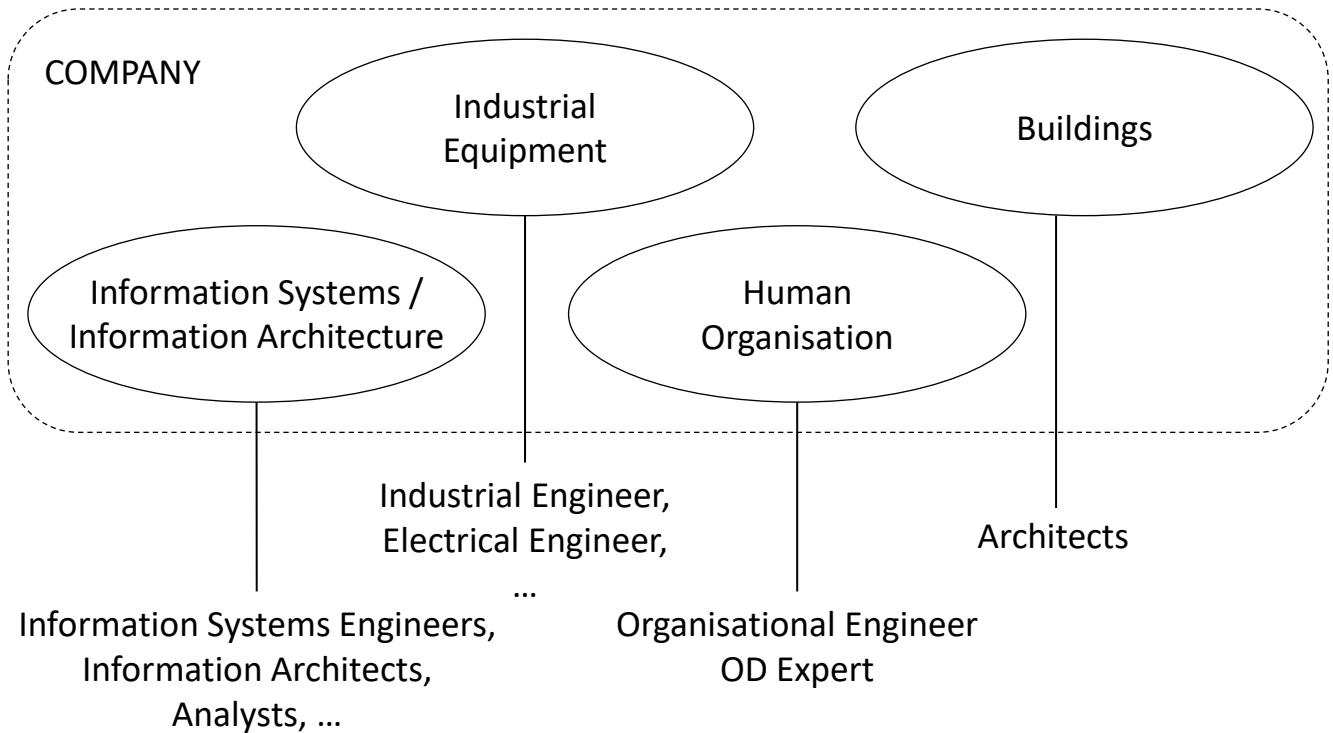
Man-made system is the outcome, the product, of an engineering activity.

Once conceived and built, the system is transferred to the user.

using a system \neq engineering a system

The user's perspective is a very shallow and limited perspective.

Using and engineering a system requires very different knowledge, competencies, goals, priorities and thinking.



Many more Engineering disciplines and Engineers

- Products → Product Engineer
- Services → Service Engineer
- Processes → Process Engineer

All systems form a coherent, integrated, aligned, efficient, effective, manageable, secure, evolvable whole.

Is a loose, sporadic, improvised collaboration among engineers sufficient to conceive this whole ?

**A very close and intensive collaboration
among all types of engineers
is paramount.**

Engineering is the application of **science** and technologies to **design** and **build** machines **structures, systems** and other things.

- **Science**
 - Knowledge + understanding + thinking + principles + methods + techniques + tools
 - Way of thinking (methodical, abstract, logical, ...)
 - Attitude (curiosity, sceptical, doubting, truth seeker, sharing, seeking confirmation, ...)
 - Specific intellectual skills
- **Design and build**
 - Conception (decision making) + specific intellectual skills
- **Structures, systems, ...**
 - Understanding of structures, mechanisms, systems, processes, ...

Notes

Can all building activities be called 'engineering' ?

Often, engineering is limited to imitating the visible actions of engineering and to produce a concrete outcome. This completely misses the essence of it.

Myth: Users can engineer the system they use.

Systems Analysis

is an

Engineering Discipline

Conceiving systems and solutions
to solve more **complex** problems

(solving problems which would otherwise not be solvable)

Informatics

is

the science of information applied in the real world.

Informatics is a science and engineering disciplines studying the usage, the organisation, the processing and the distribution of information in natural and artificial systems while pursuing the effective uses of it.

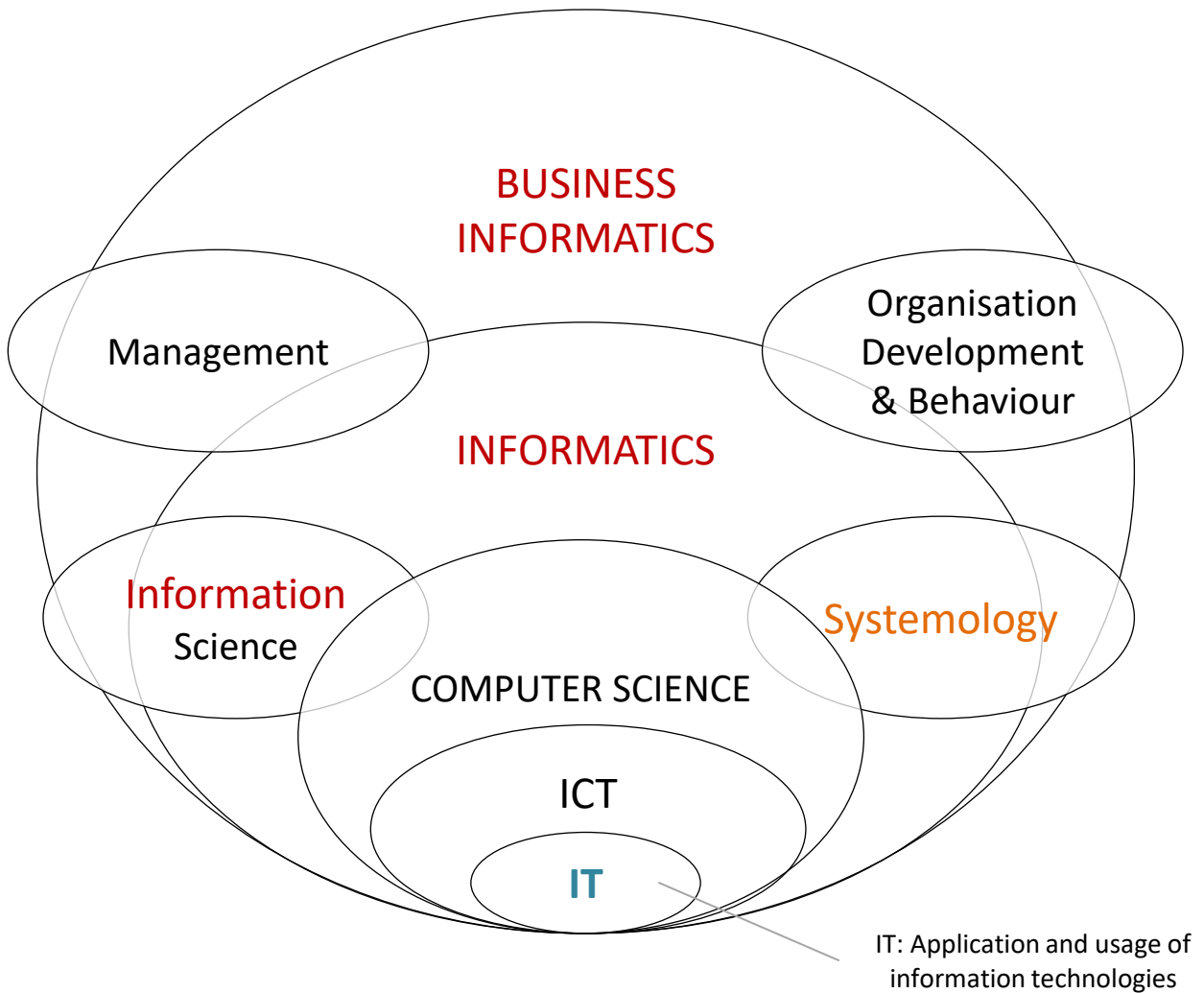
It studies information and knowledge in organisations as well as in society at large.

The objective of Informatics applied in organisations is to improve the organisation and its functioning and to contribute to the value creation for its clients with the common and implicit, but not obligatory, intention of automation (IT).

Advanced level: a scientific field that focuses on different informational phenomena, such as cognition, thought, language, communication, and computation. It is an interdisciplinary and integrative scientific field involving computer, communication, cognitive, and social sciences.

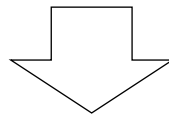
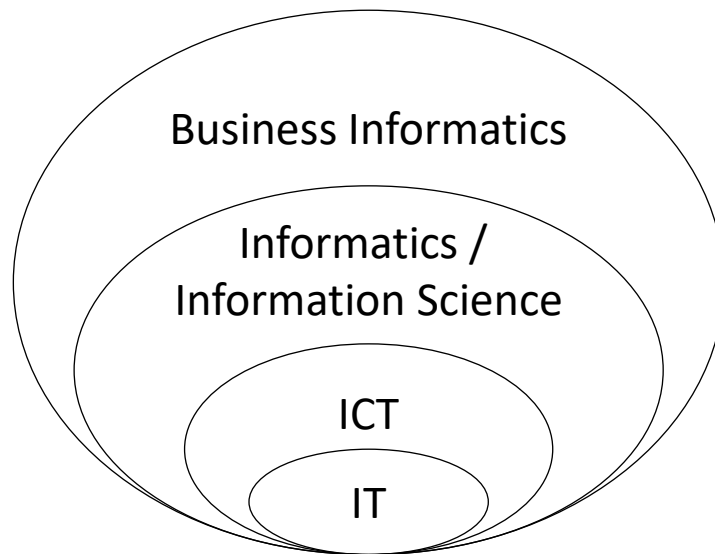
It is commonly practised in multidisciplinary initiatives.

Informatics is an interdisciplinary science

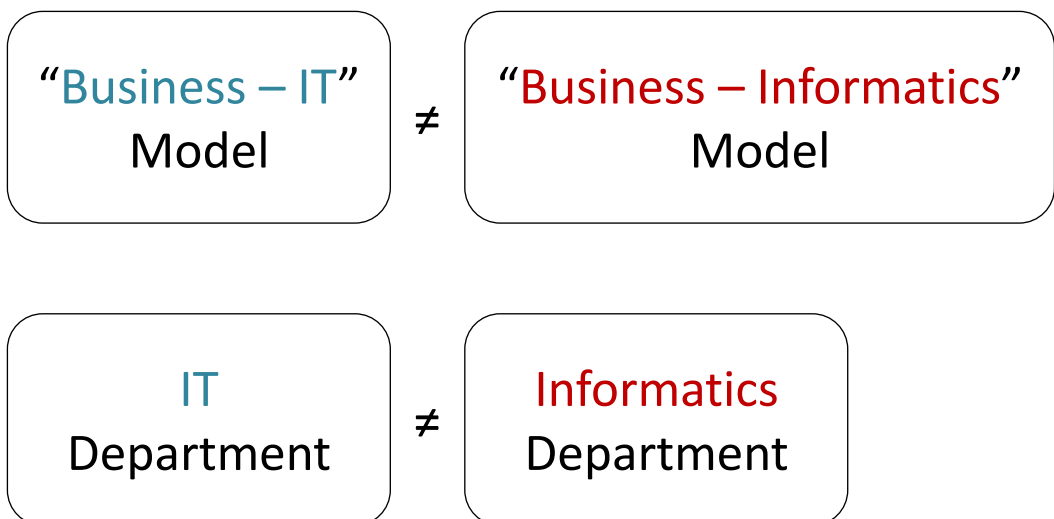
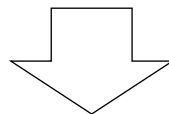


Informatics contains 3 of the 4 Critical Knowledge Spheres, and more.

- Core Notion: organisation, transformation, communication, interactions and usage of information by organisms or machines in natural and engineered systems and their environment.
- Information Science, Information Engineering
- Information Systems Engineering, Information Processing
- Computer Science, ICT / IT
- Links with Mathematics, Statistics, Electronics, Biology, Linguistics, Psychology, ...
- computational, cognitive and social aspects



IT \neq **Informatics**

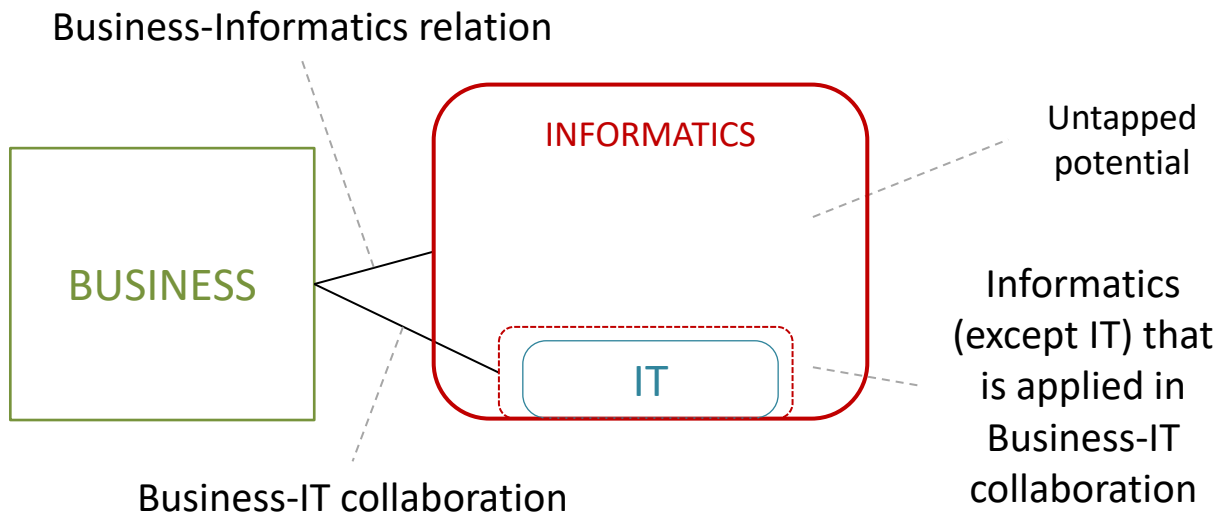


Untapped Potential in IT and Informatics

IT as
sub-discipline of
Informatics



The Informatics (the non-IT part)
defines
what IT implementations should do



Informatics offer possibilities to create value,
the Business community is not aware about.

That's where all the power is !!

That's what can make the difference !!

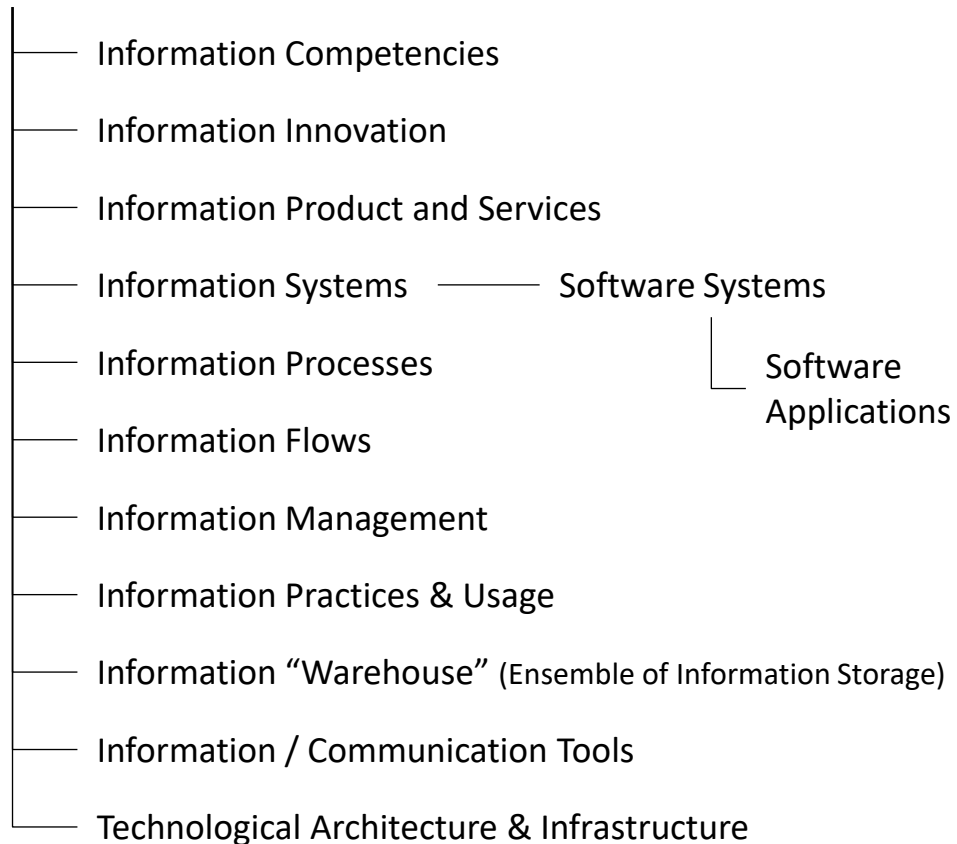
Product of the Informatics Dept. 30/10/2018

INFORMATICS DEPARTMENT



PRODUCES

INFORMATION COMPONENT *implements* INFORMATION CAPABILITIES



Informatics Dept.
Needs :

Capability of Engineering, Developing and Managing
the Information Component

Meta Information about Information Component

Notes:

Informatics Department is an alternative name indicating that the department plays a more important role and does way more than just “IT”. Another alternative name could be the “Information Department” (similar to the “Finance Department”)

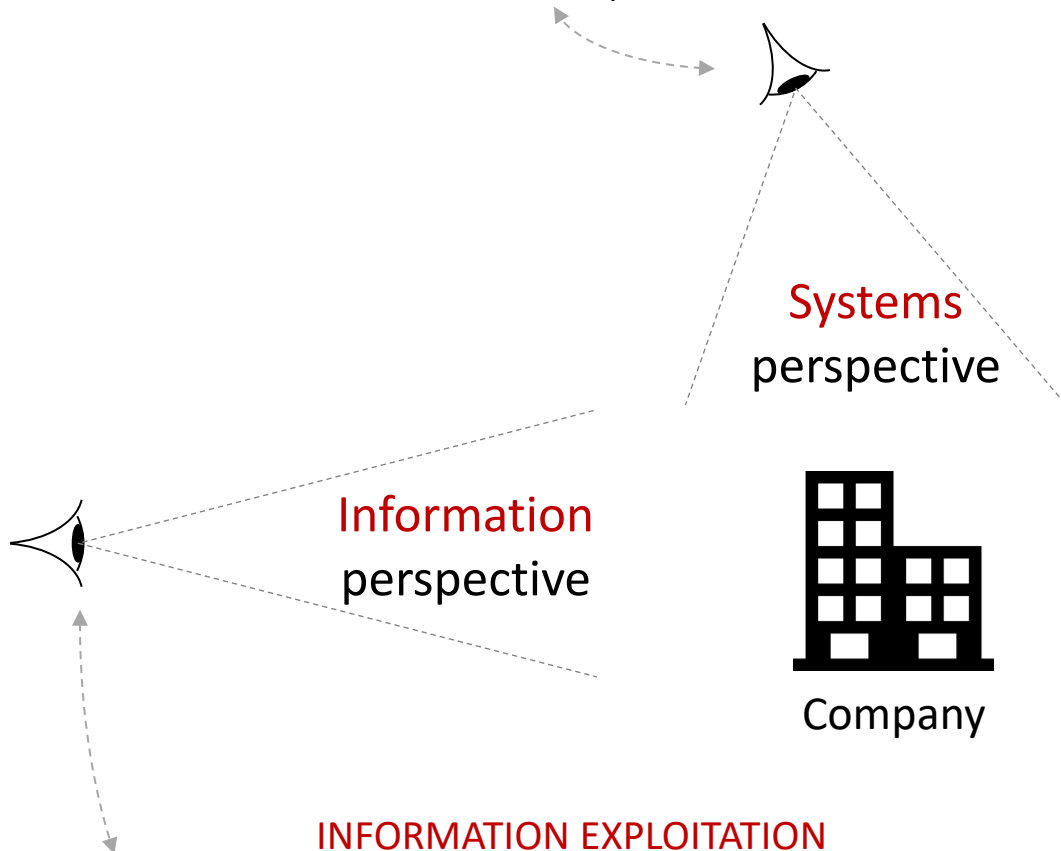
1. INFORMATION AS RESOURCE
2. INFORMATION QUALITY & VALUE
3. INFORMATION CAPABILITIES
4. INFORMATION EXPLOITATION
5. INFORMATION COMPETENCIES
6. INNOVATION THROUGH INFORMATION
7. INFORMATION AS PRODUCT OR SERVICE
8. INFORMATION NEEDS
9. INFORMATION PRACTICES & USAGE
10. INFORMATION RISKS
11. INFORMATION SYSTEMS
12. INFORMATION PROCESSES
13. INFORMATION FLOWS, SHARING AND REUSE
14. INFORMATION CONCEPTS
15. ORGANISATION OF INFORMATION
16. INFORMATION WAREHOUSE
17. INFORMATION MANAGEMENT
18. AUTOMATION / AUTOMATED SYSTEMS
19. INFORMATION AND COMMUNICATION
TECHNOLOGIES, TOOLS AND INFRASTRUCTURE
20. TRAINING / INFORMATION SKILLS DEVELOPMENT

Two Important Perspectives

30/10/2018

SYSTEMIC PERSPECTIVE

How can we conceive, strengthen, improve and expand the systems, structures and processes dealing with information of the enterprise?



INFORMATION EXPLOITATION

How can we maximise the capability of the company or enterprise to exploit information resource? How can we increase the value of the information and use it to the benefit of the company and its customers.

Business

Business Thinking

How do we organise and conduct business? What products and what services can we provide to the market? What materials do we need? Who can supply us? How to distribute the products and services? At what price can they be sold ? ...



Information is required to make this work. So, we have to ask the question of information.

Information

Information and Systemic Thinking

How can information support or execute business activities ?
How does information has to flow?
How can we organise information?
How can information be exploited to improve business?
How can information drive and innovate business? What systems are needed ? ...

Now raises the question of who will execute it. It should preferably be automated (software and computers) and how to translate this into software system.

IT – Application of Technologies

Implementation, Building & Technological Thinking

What technologies should be used ?
How to use technologies ?

Technological vocabulary and skills; solves technological issues

This is a “LAYERED PROBLEM”

(see chapter Problem Solving)

Business

Exploitation of BUSINESS MEANS

Producing and selling goods and services; deployment of business means and activities; making use of business opportunities



Information

Exploitation of INFORMATION

to allow the functioning of the business arrangement through Information Organisation, Information Capabilities and Systems

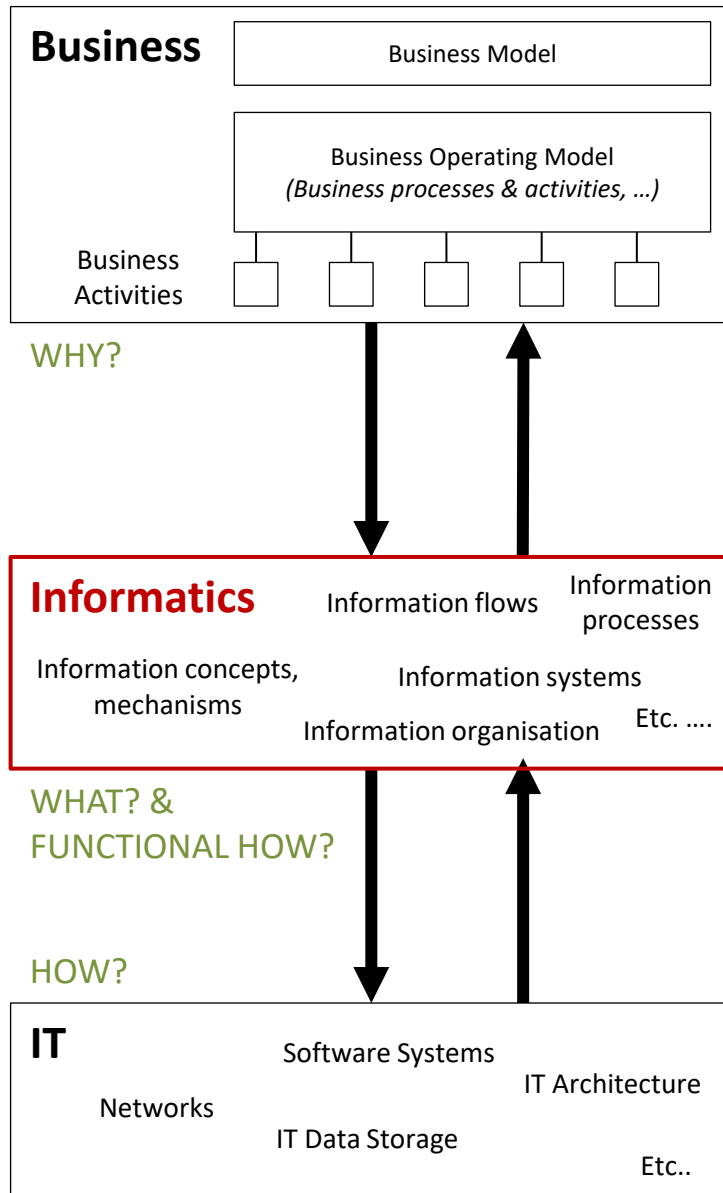


IT – Application of Technologies

Exploitation of TECHNOLOGIES

to allow the exploitation of information

New Business – INFORMATICS Model



Business Thinking

Defining Products and Services;
Determining the business models, strategies, tactics, policies, rules and operations;
Solving business problem;
Executing and managing business activities

Information and Systemic Thinking

Determining what information to collect, to capture or to create
Determining how to maximise the quality and value of information
Determining how to maximise the exploitation of information to benefit in business activities
Determining information solutions supporting the business activities
How to organise this information, how to keep it clean, store it, etc...
...

Implementation, Building & Technological Thinking

Determining how to structure the solution
Determining what technologies should/can be used and how they will relate to each other (technological architecture)
Determining how technologies will implement the information solution
...

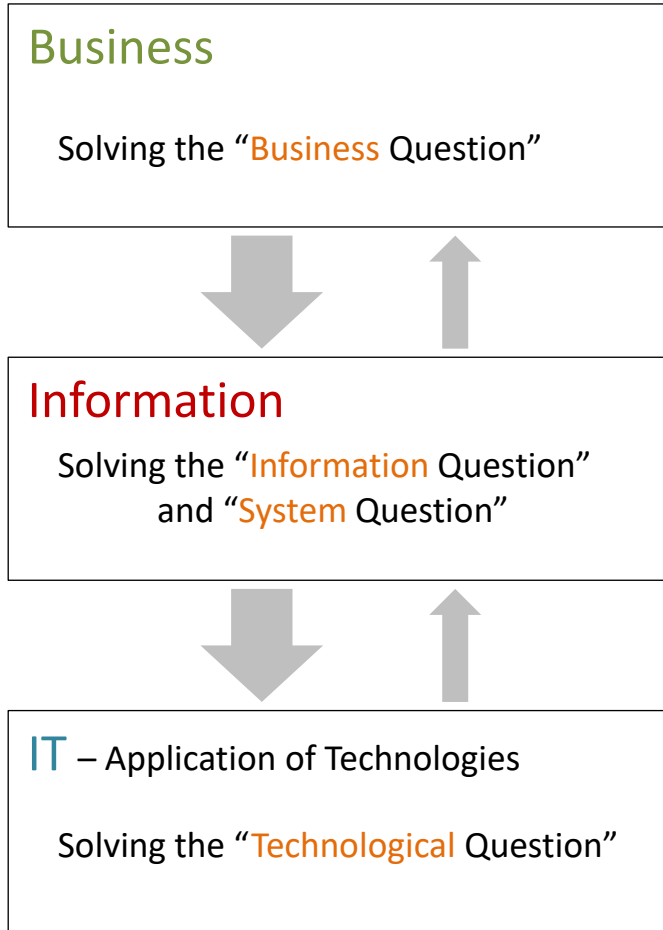
Notes:

It is not that Informatics is absent. It is watered down and undervalued. It is practised by business people and IT people. IT people have some more knowledge in informatics, but it is not really their focus or objective.

The flaw is somewhat mitigated by Analysts and Architects.

IT is a sub-discipline of Informatics. We consider here the part of Informatics that is not IT.

Necessary SKILLS



Business Subjects,
Business Knowledge,
Business Expertise,
Business Skills,
Business Techniques,
Business Vocabulary,
Management, ...

Requiring Information Science,
Systemology, Systemic Thinking,
Problem Solving,
Analytical Thinking,
Informatics vocabulary,
Informatics Techniques, ...

Technological skills,
Programming skills, Ability
to build systems by using
technologies,
...

Different **Purposes** or **Goals**

Different **Intentions** and **Perspectives**

Different **Problem to be solved**

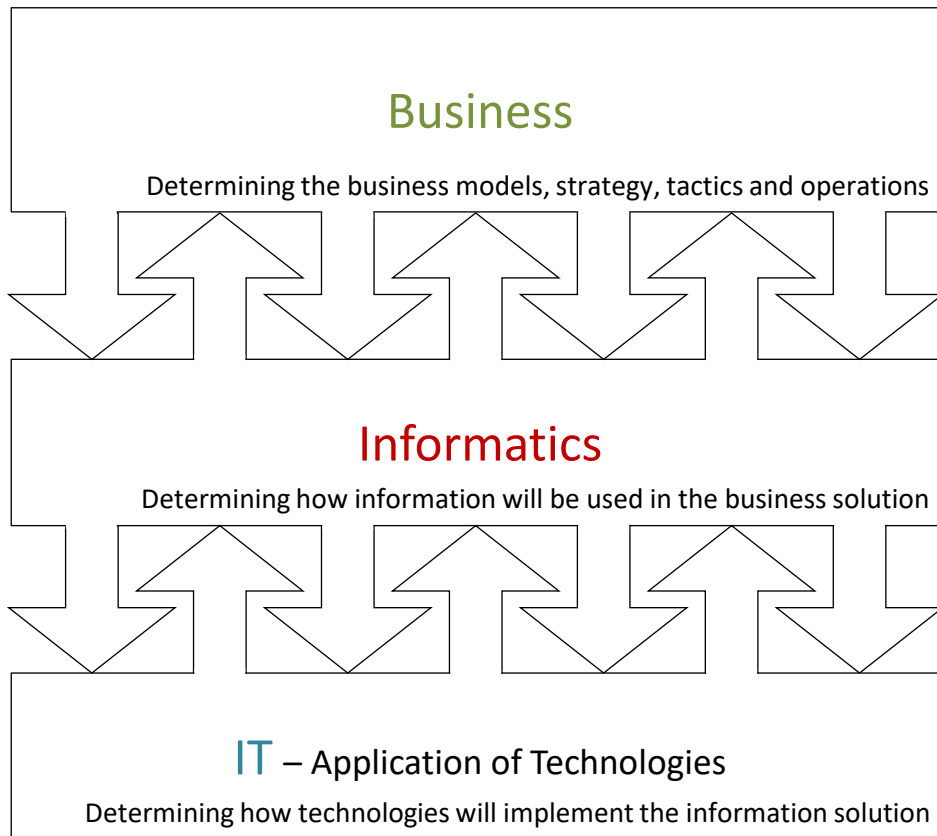
Different **Matter** to be processed

Different **Way of Thinking**

Different **SKILLS**

Informatics in Business Environments

10/01/2020



Traditional and present 2-layered Business – IT Model

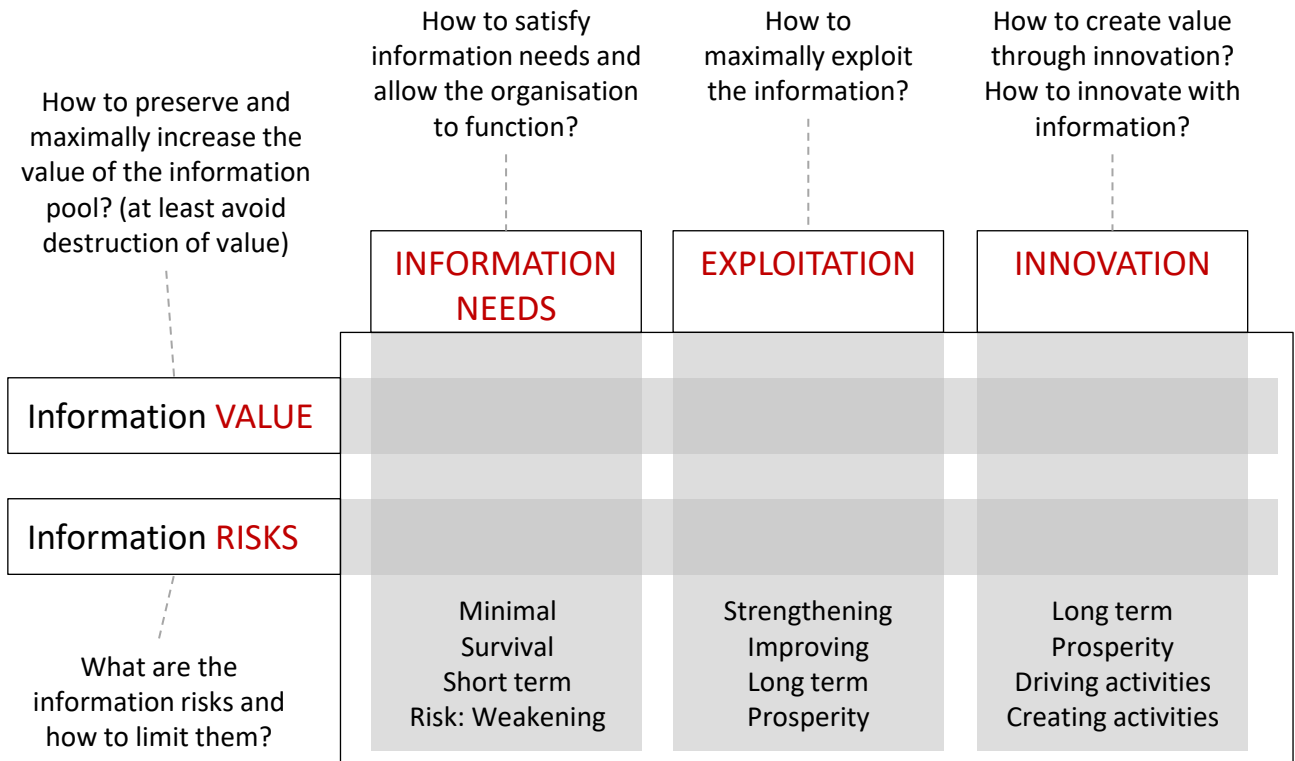


Informatics, the non-IT part of the discipline, is often vague, shallow, and misapplied. And it is often performed by business people and by IT people.

Notes:

NOT BUSINESS TERMS: website, ERP, data warehouse, internet, web, software application, apps, software system, screen, windows, menus, button, entry field, database, tables, records, indexes, information model, information entity, attributes, relations, state transition diagram, data model, object model, Java, Python, HTML, etc..

A PERSPECTIVE ON CORPORATE IT'S MAIN QUESTIONS



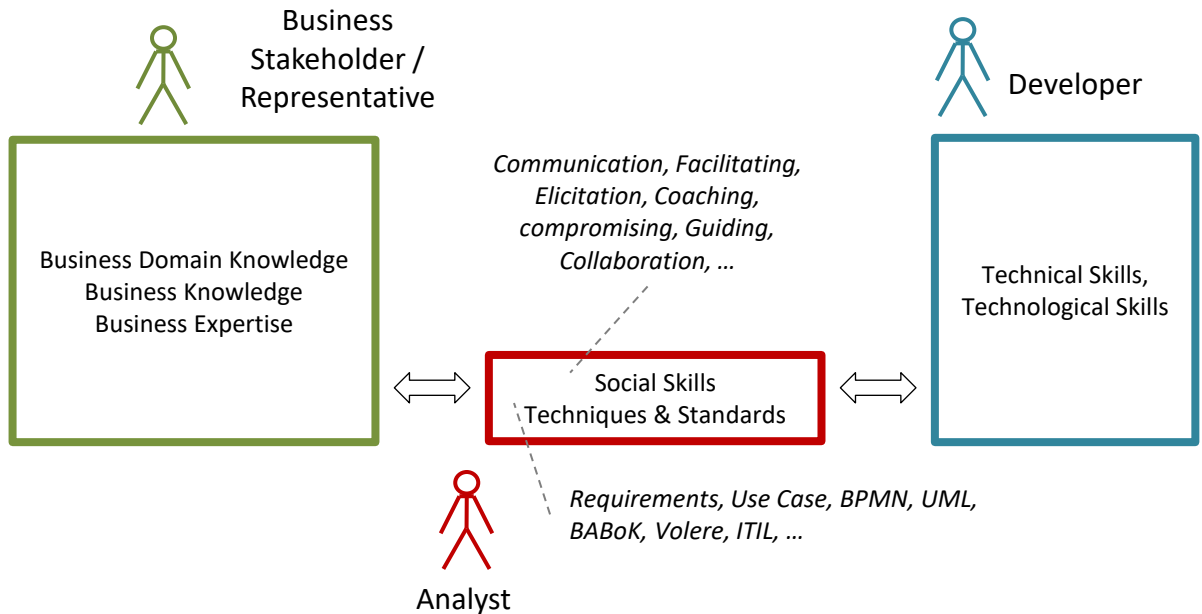
The ANALYST should keep these questions
always in mind

Two other key-questions, regardless of needs or demands are:

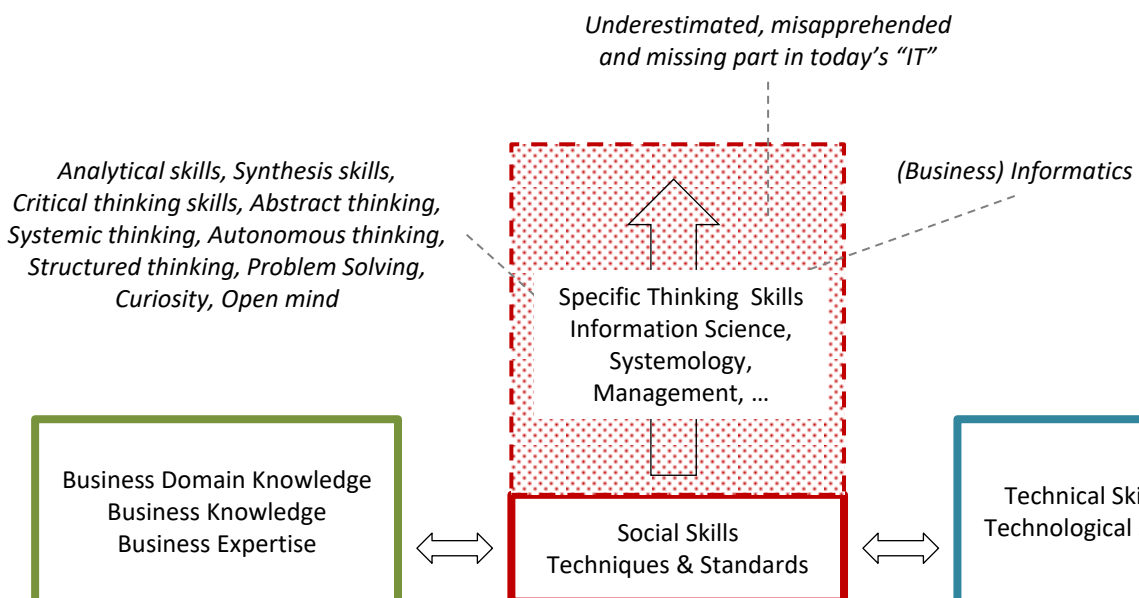
- How to preserve and maximise the value of information?
- What information risks are or will be faced and how to avoid or to limit them?

Strengthening the Role of the Analyst

Presently



Exploitation of the full potential of the discipline and increasing the added value requires the expansion and consolidation of the role of the Analyst.



~~Anyone who performs Analysis activities~~

*Undermines and destroys the
profession of Analyst*

Anyone
who **MASTERS**
the Systems Analysis discipline,
or any of its variants or sub-sets,
and the capabilities to apply it

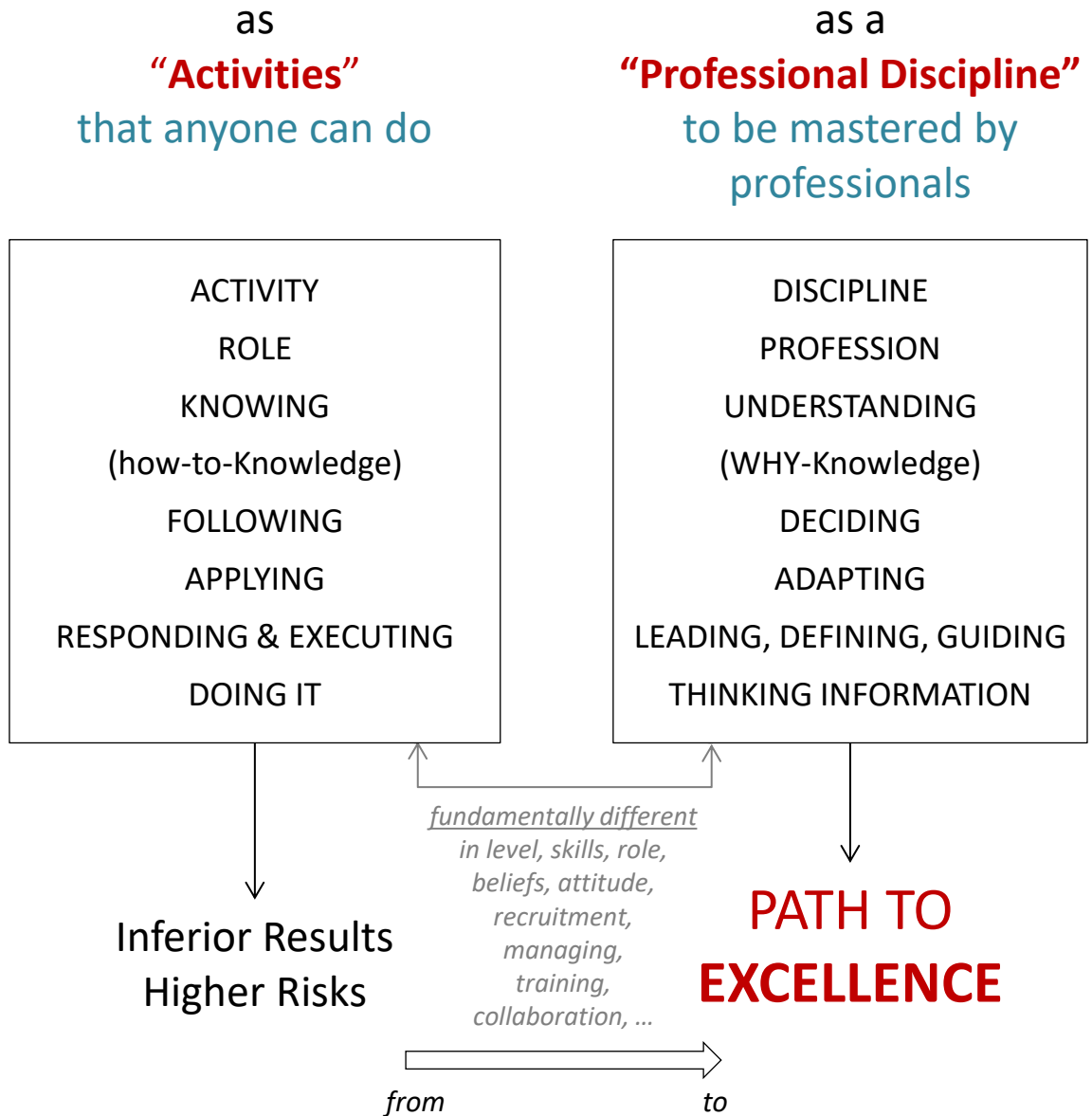
One does not become
professional Analyst
overnight.

"What got you here won't get you there" (Marshall Goldsmith)

Notes:

- Every person can fight, but not everybody can fight like a black belt martial art practitioner.
- Assuming that performing analysis activities turns someone into an Analyst or that anyone can become Analyst in a short period of time is a immense underestimation of the discipline and of the true importance of the role.

Perception of Systems Analysis

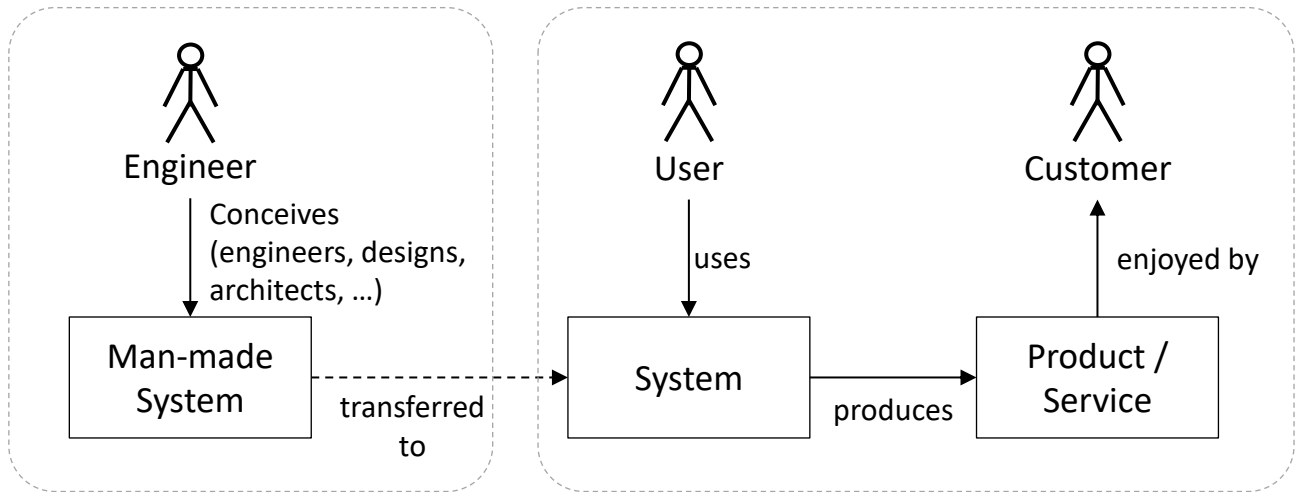


Notes:

- Considering ‘Analysis’ as a set of activities, assuming anyone can be an analyst, assigning the role to an employee, learning and applying techniques and following standards, responding to demands, seeking to satisfy others, being lead, copying, perpetuating (bad) habits, ... → key to disasters
- Developing oneself to an expert, learning to master the discipline, thinking independently, determining and organising own work → professional

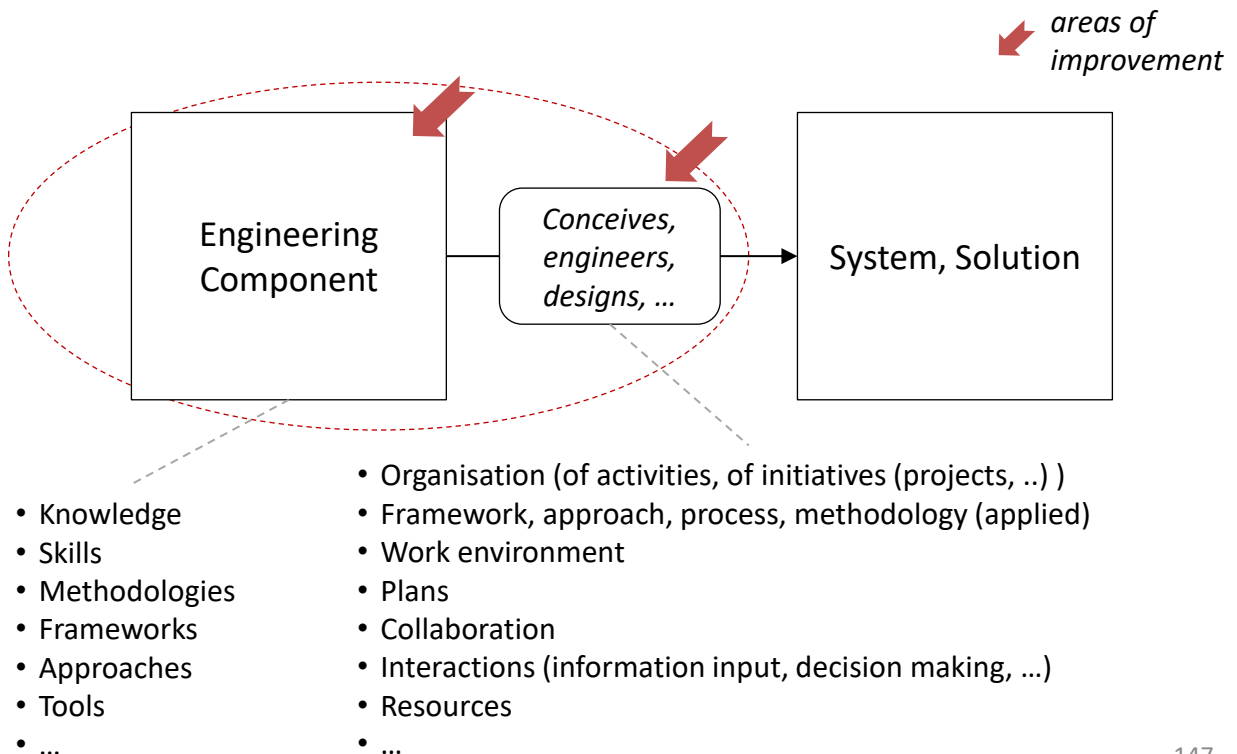
Improving the Engineering

17/09/2019



Improving this enhances the ability to improve this.
(= **PROFESSIONALISATION**)

Improving the Engineering Component and its initiatives
leads to better systems and solutions.



Pursuing Higher Goals and Professionalisation

30/10/2018

- Better tools
- Better collaboration and social interaction
- Better methods and techniques

- More **knowledge**
- Better **insight**
- Understanding ineffective **approaches** and **habits**
- Clarify and eliminate **assumptions**
- **Beliefs**

TARB

Our overall view, our perception, our belief system
and how we perceive our role,
determine
the results we can produce
and thus the value we can create.

It can limit it or it can increase it remarkably!

Professionalisation is **not a luxury**.
It's a **must**.

*First things learned
by a junior*

Mastery of disciplines:

- Vocabulary – Major terms
- Positioning of the tool, technique, methodology, concept, ...
- Knowledge of the process(es), method(s), tool(s)
- Understanding the general idea behind the process(es), method(s), tool(s)
- Ability to use the tool or to apply the method/technique/process as prescribed
- Understanding the expected result
- Develop the own skills
- Understanding the meaning, purpose and value
- Understanding preconditions, applicability, key aspects, importances, priorities, options, alternatives, strengths and weaknesses, constraints, the limits, risks, the evolution, forces, influencing factors and tendencies
- Understanding the why and effects of every step or aspect
- Able to adapt the technique, method or usage of the tool to the context and circumstances. Understanding what can be done and what can't be done.
- Understanding variances; ability to tune
- Able to take a broader range of the aspects into account
- Understanding the underlying purposes, mechanisms, laws and principles
- Ability to see and understand the true nature of things
- Ability to put everything in broader perspective
- Limiting the unwanted effects
- Understanding effects, consequences and implications of decisions and choices
- Identification of opportunities to apply the technique or to use the tool
- Optimal exploitation of the approach, method, technique, tool
- Able to further develop the discipline
- Attitude: open-minded, ability to learn, curiosity, critical thinking, ability to identifying and understanding what is imperfect or problematic, seeking to do right, methodical, sharing knowledge, ready to go the extra mile, pursuing **higher goals** or harder challenges, using **higher norms**, autonomous thinking, **truth-seeker**, ...



To find out if lower or more higher goals are pursued:

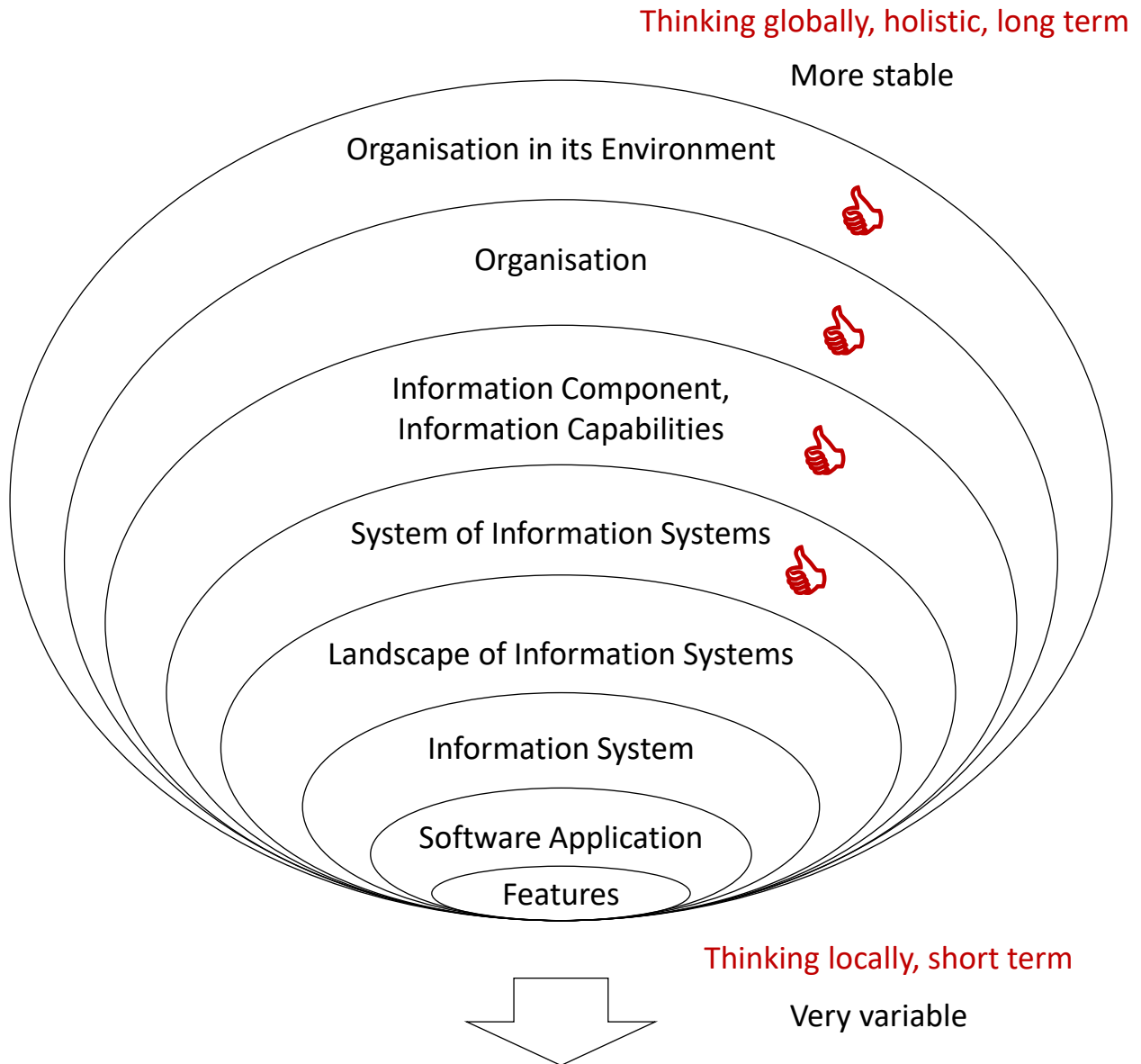
- What goals drive the thinking, decisions, actions and initiatives?
- What defines our priorities? What do we consider as 'valuable' ? ...

Notes:

- Companies continue to computerise. More and more work is executed by software. Some systems are of strategic importance (example: supply chains). But the Analyst can't contribute at the strategic level if the mind still is driven by the idea of "developing demanded software features to satisfy the user".
- Being driven by the higher level goals requires a more holistic approach, different goals, different competencies, different relation with the business, different responsibilities, ...
- Working at a higher level is harder, but much greater value can be created.
- Trying to satisfy a lower goal, being driven by the lower motives, does NOT automatically imply a positive contribution to the higher plans and goals. Even the opposite can be true (example: sub-optimisation; creation of obstacles, limitations and incoherence; tendency to fragmentation and chaos; ...)

Levels of Thinking

30/10/2018



*Each level has a different type of **goals, scopes, environments** and leads to different types of **products and results**. It requires different **approaches** and **methodologies**, different **responsibilities** and **authorities**, different **methods** and **techniques** and requires different **competencies** !!!*

The thinking of the lower level is inappropriate to conceive solutions for the higher level.

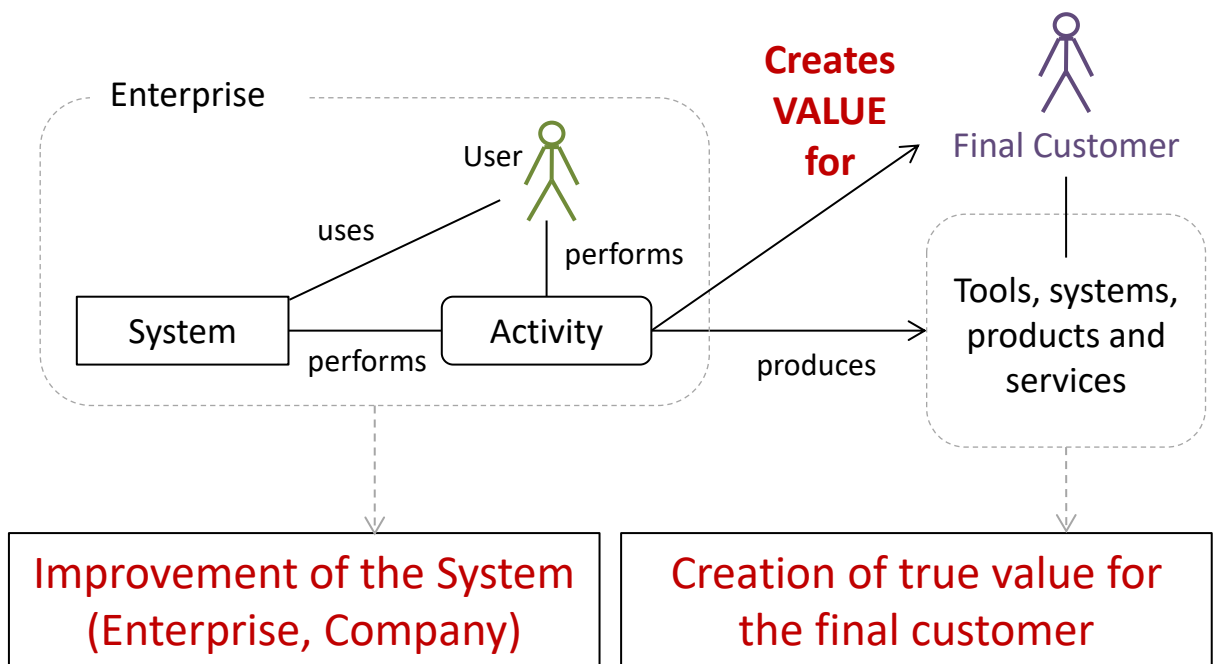
Creating Value Through Engineering

17/09/2019

How do we, as Analyst/Engineer/Architect, create value?

A system creates value by its **functioning** and by its **usage**. This value is **for the final customer**.

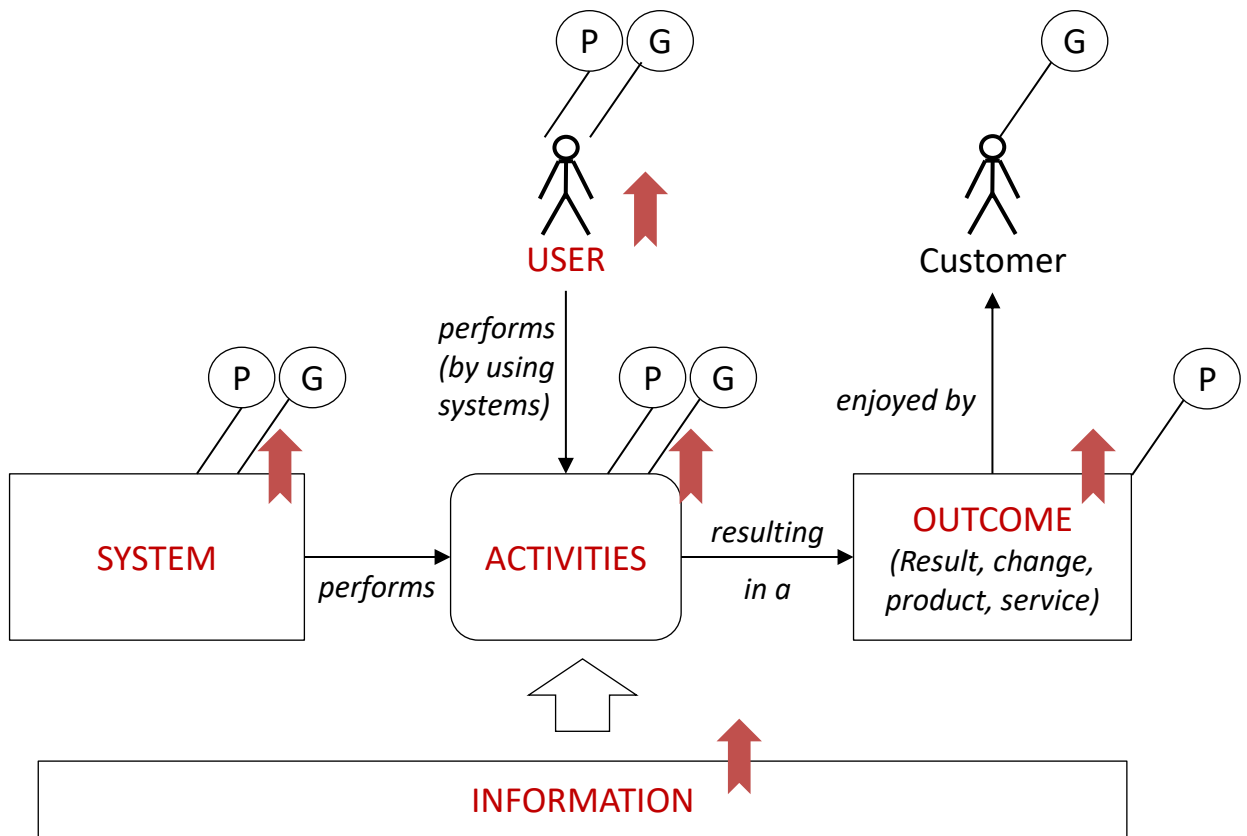
Example: A car creates value by functioning and by the usage the driver does with the car.



This can and has to be further detailed.

Creating Value Through Engineering

17/09/2019



↑ areas of improvement

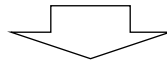
1. Improving and developing the **SYSTEM**
2. Improving and developing **ACTIVITIES**
3. Improving and developing **PRODUCTS** and **SERVICES**
4. Increasing the **AMOUNT** and **VALUE** of the **INFORMATION**
5. Improving and increasing the **EXPLOITATION** and **USAGE** of **INFORMATION** (User's knowledge, skills, behaviour & actions)

In line with **PURPOSES, NEEDS** and **GOALS**

Higher outcomes are the trigger, the goal and the motivation of the Analyst

Improve and renewing the
supra-system (company)

Driving and renewing the
Business (Final Customer)



How?

	System	Activities - Processes	Outcome, Products, Services
Reduce required material			
Reduce waste			
Reduce risks			
Creating opportunities			
Improve capabilities			
Improve characteristics			
Innovation			

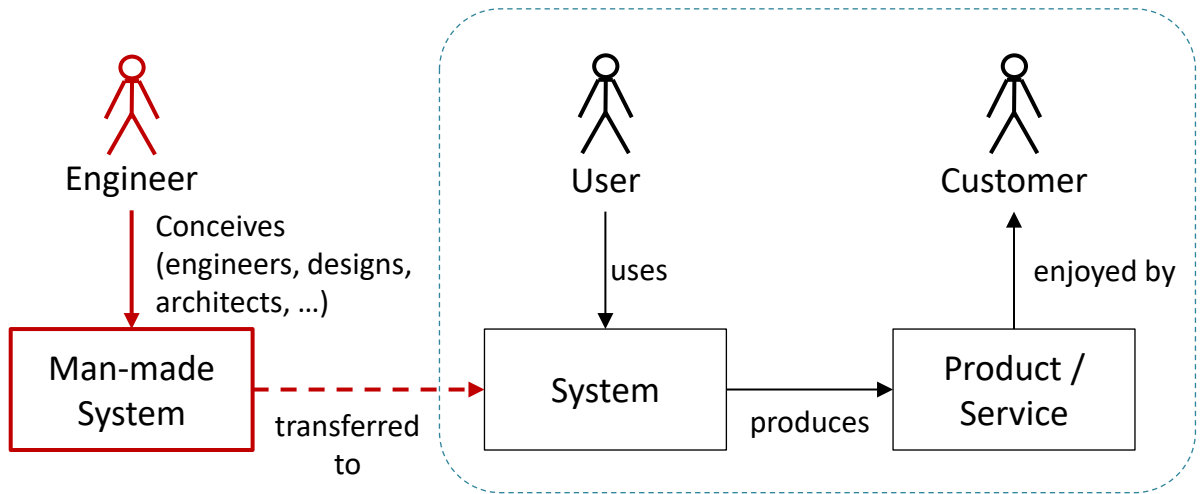
Main constraints:

- Not degrading or endangering the supra-system (enterprise), its environment or the resources
- Aligned with strategy and plans

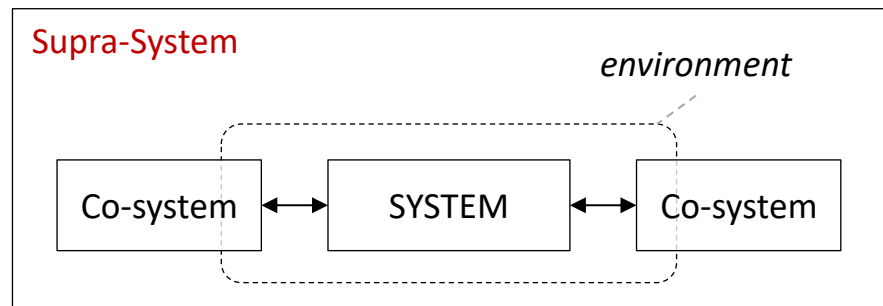
Creating Value Through Engineering

7/09/2019

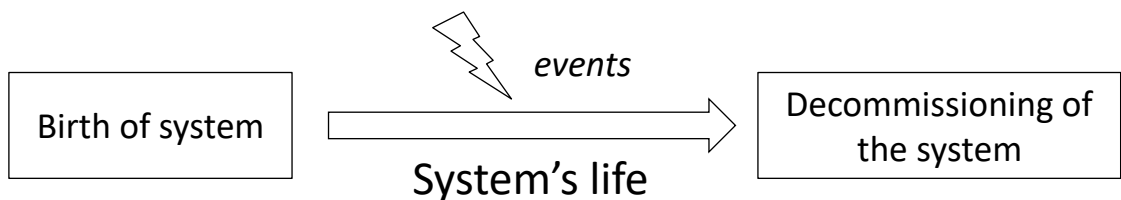
1) Functioning, Usability and Value Creation



2) Integration and interactions with other systems and environment



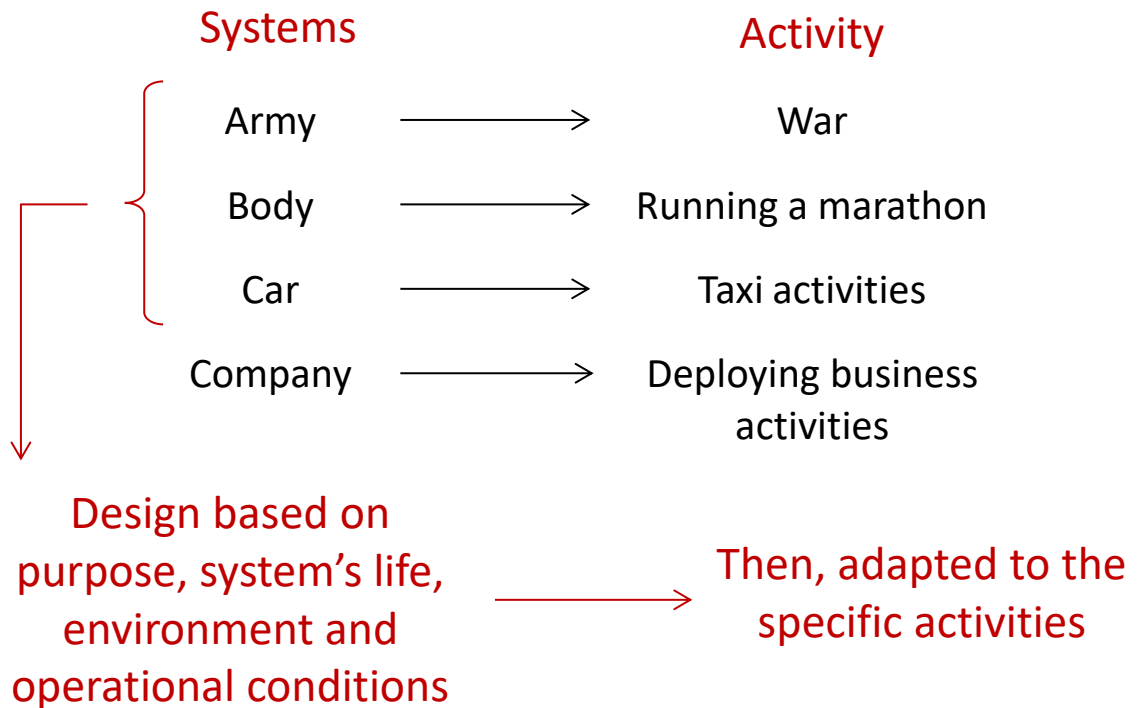
3) Capability to maintain its qualities during the whole system's lifespan (includes capability to evolve)



The ANALYST
is the
INFORMATION &
INFORMATION SYSTEMS
“DOCTOR”

The Company & Business Activities
are
the patients

even if they are impatient



Also true for company? For information systems?

Principles

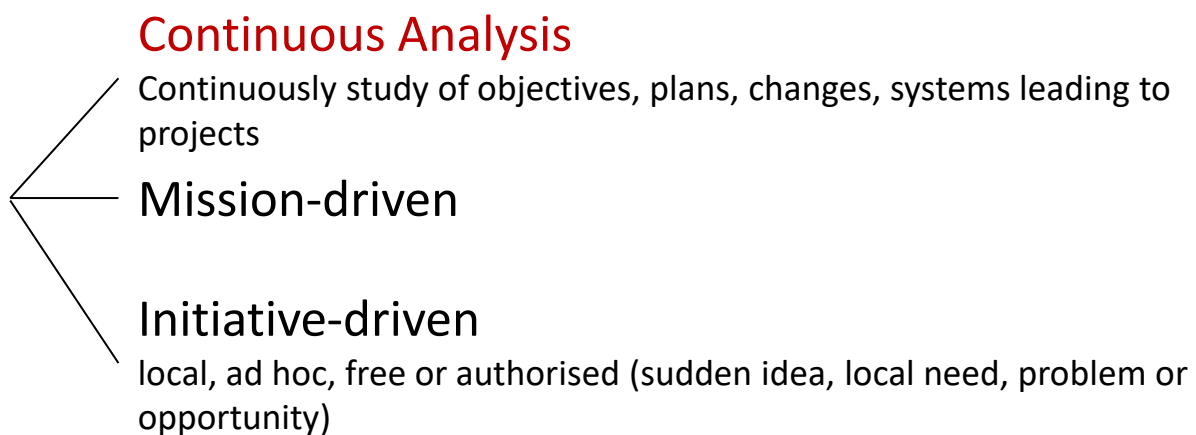
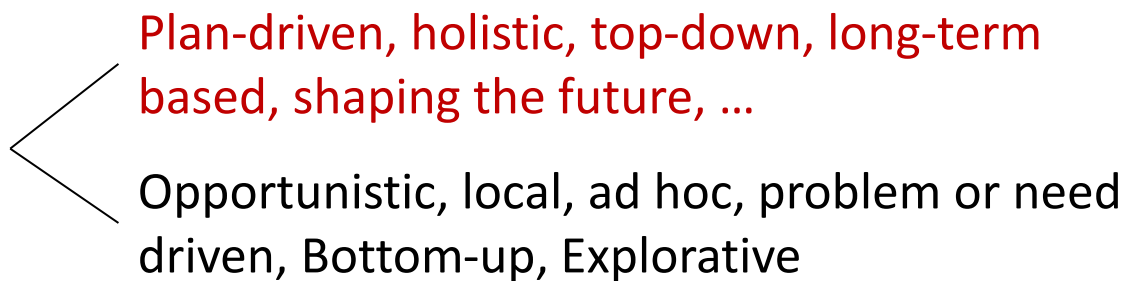
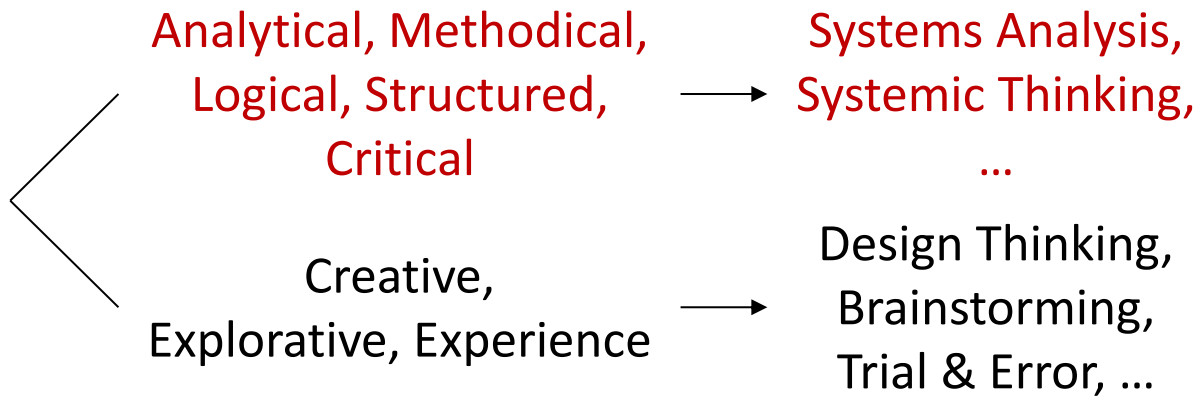
- Simple and cheap tools or features can be designed for purpose and usage.
- Complex, expensive and lasting systems must be designed primarily based on their purpose, their future life and their environment (operational conditions).

Notes:

- What if we declare war, without building or strengthening the army first?
- What if we run a marathon, but ignore the capabilities of the body and its signals?
- What if we are able to run one marathon, but finishing exhausted. What if we have to run continuously marathons one after the other? (sustainability)
- Or, can we do a sprint, after sprint, after sprint (running)?
- What if a car can transport people only for a too short period of time or if it can't operate when it's raining, when it's freezing or when it's too hot?

Some Major Engineering Approaches

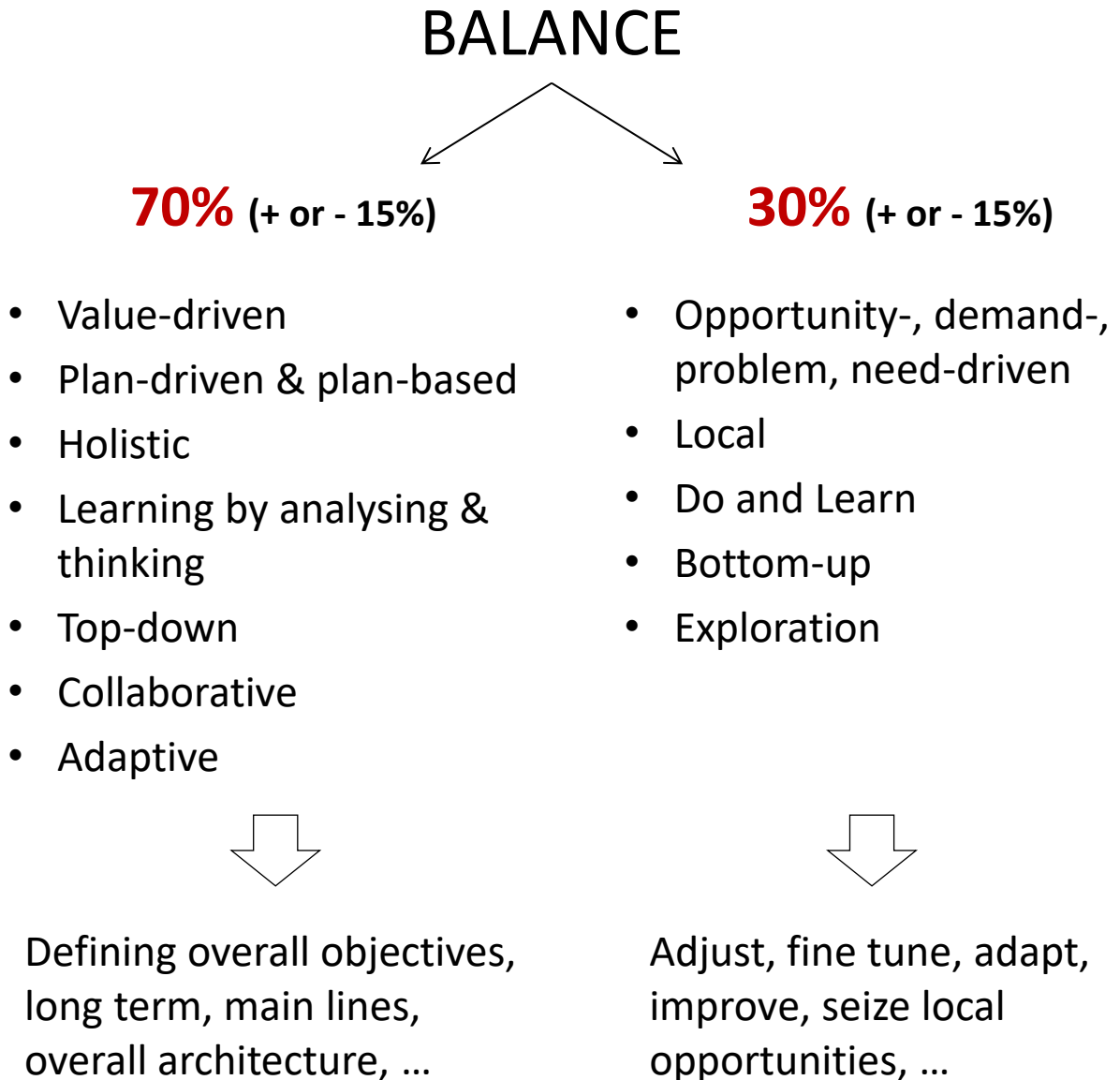
30/04/2019



Notes

- Some approaches can be combined in one.
- A company may need to have or allow different types of approaches. It's not about choosing one single approach for the whole company. Usually, one main approach can make sense.

Two Major Engineering Approaches

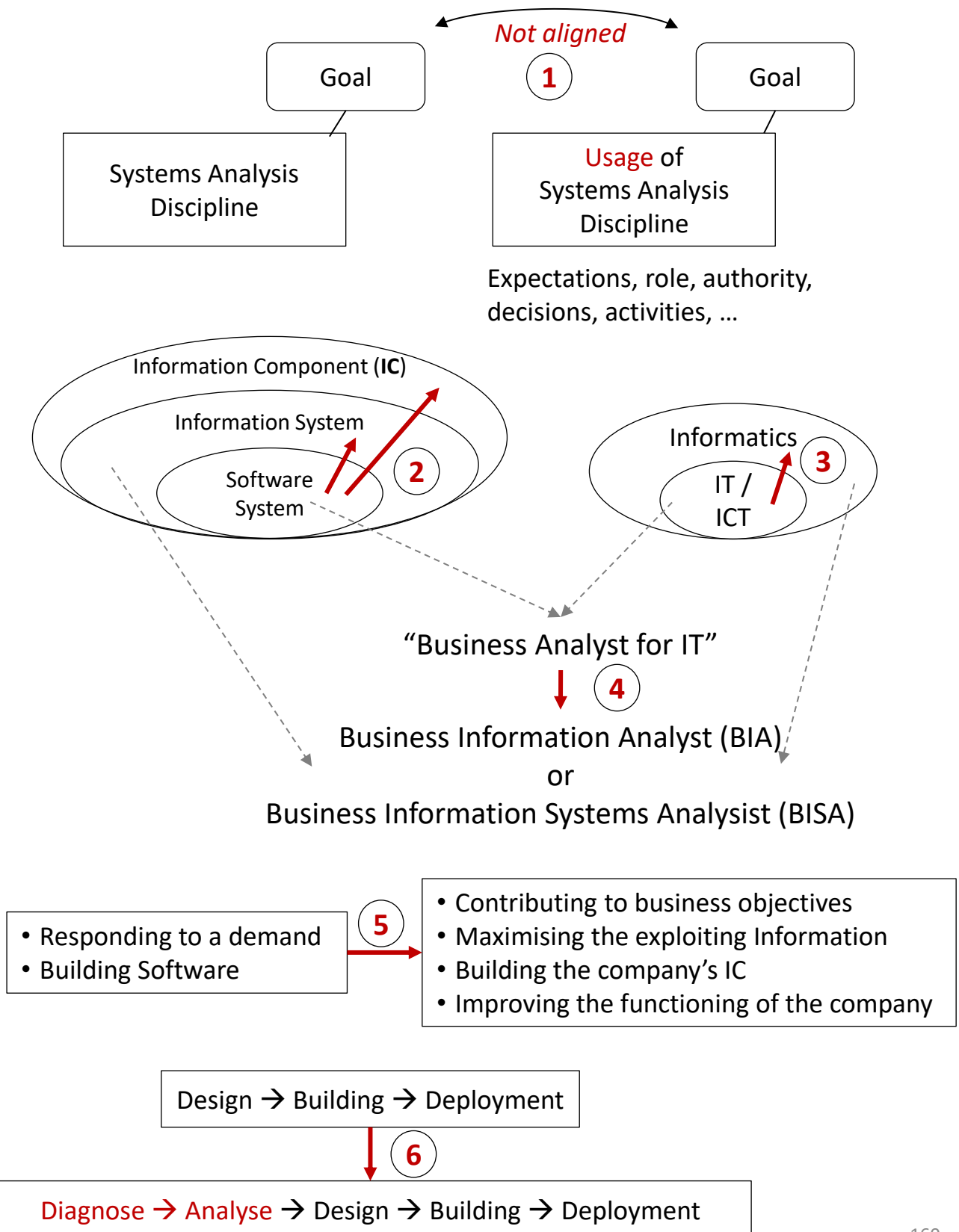


Notes

This depends largely on the industry, on the type of system (core vs peripheral) and on the customer (example: operations vs marketing)

Some Important Shifts

15/02/2019



PARADIGM SHIFT

Goal

Facilitating the activities of the business community by putting in place technologies and by implementing software systems



Driving the company's business (activities) by

1. Improving the functioning of the company by using S.A. methods and by working on the information domain
2. Maximally exploitation of the available or obtainable information
3. Developing information capabilities
4. Implementing an Information component
5. Innovation through information

Product

Implementing **software** systems, software applications, software features and IT infrastructure



Delivering ...

1. Optimised and streamlined company **processes**
2. Enhanced information **capabilities**
3. Optimised, streamlined, manageable and flexible **information component** aligned with other company's sub-systems and integrated into the company's body
4. Organised, flexible and manageable **information** of increased value
5. Conception of new **information products** and **information services**

Value

Value comes from the implementation and usage of **technologies**.

Huge focus on technologies:
How can we use technologies to the company's advantage?



Value comes from **information**

Focus on information.

- How can information benefit to the company?
- How can we better exploit information?

The technology-question is still present, but it is subordinate to this.

PARADIGM SHIFT

Trigger / Input

Responding to **demand**
Delivering what is **wanted, liked or preferred**
with assuming to create value
and to satisfy customers



Mainly driven by **goals, plans, context** and **constraints**
Based on what is **diagnosis** and **necessary**
Requires the understanding of the real situation
Also through ***Continuous Analysis***

Approach

Reactive, short-term, local:

Problem-driven,
Opportunity-driven,
Need-driven,
Demand-driven.



Global system design should be (mainly)

- Vision-based
 - Long-term oriented
 - Inspired by Leadership
 - Pro-active
 - Creating the future
 - Holistic, integrative, multidisciplinary
- Problems, needs, opportunities and demand will continue to be dealt with, in particular when it is aligned within the global approach.

Client

Serving

1. the business stakeholders
2. the end-users

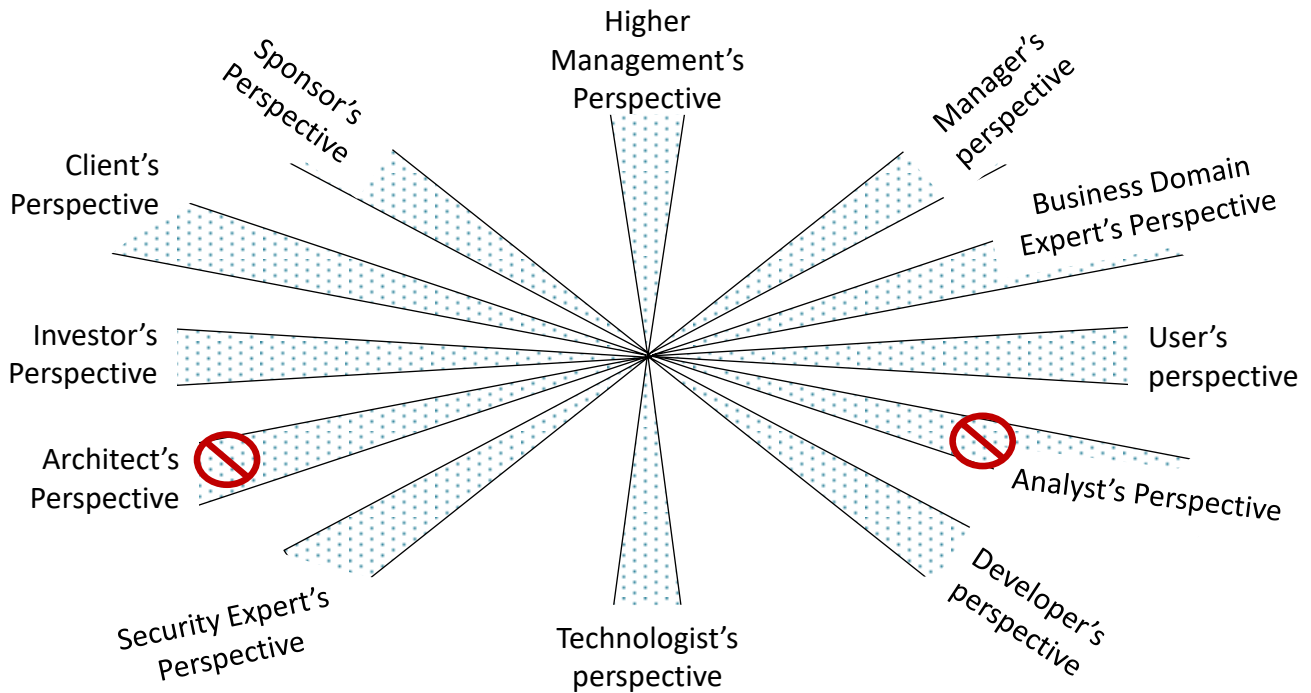


Satisfying the needs of

1. the company and its clients
2. the business community and the IT/informatics department
3. direct business stakeholders
4. end-users

Analyst Being Part of the Problem

10/01/2020



Examples:

Top Management's perspective:

Can we implement our vision and strategy? What will be the position of the company and its results? Can the company grow? Can the company have a stronger market position?

Management's perspective:

Can we achieve our objectives?

Sponsor's perspective:

Priority? Cost? What results will it yield?

User's perspective:

Will I be able to do my work? Will I reach my criteria? What will I have to learn ? Will my work become easier and more interesting?

Analyst's Perspective:

What do customer and users want? What solution will they find acceptable? What are the business requirements? What will I give to the developers?

Developer's perspective:

What do I have to develop? What technologies will I be able and allowed to use? What will the user like and use ?

Very Ineffective: If architects and analysts work from their own perspective, then they are PART OF THE PROBLEM.

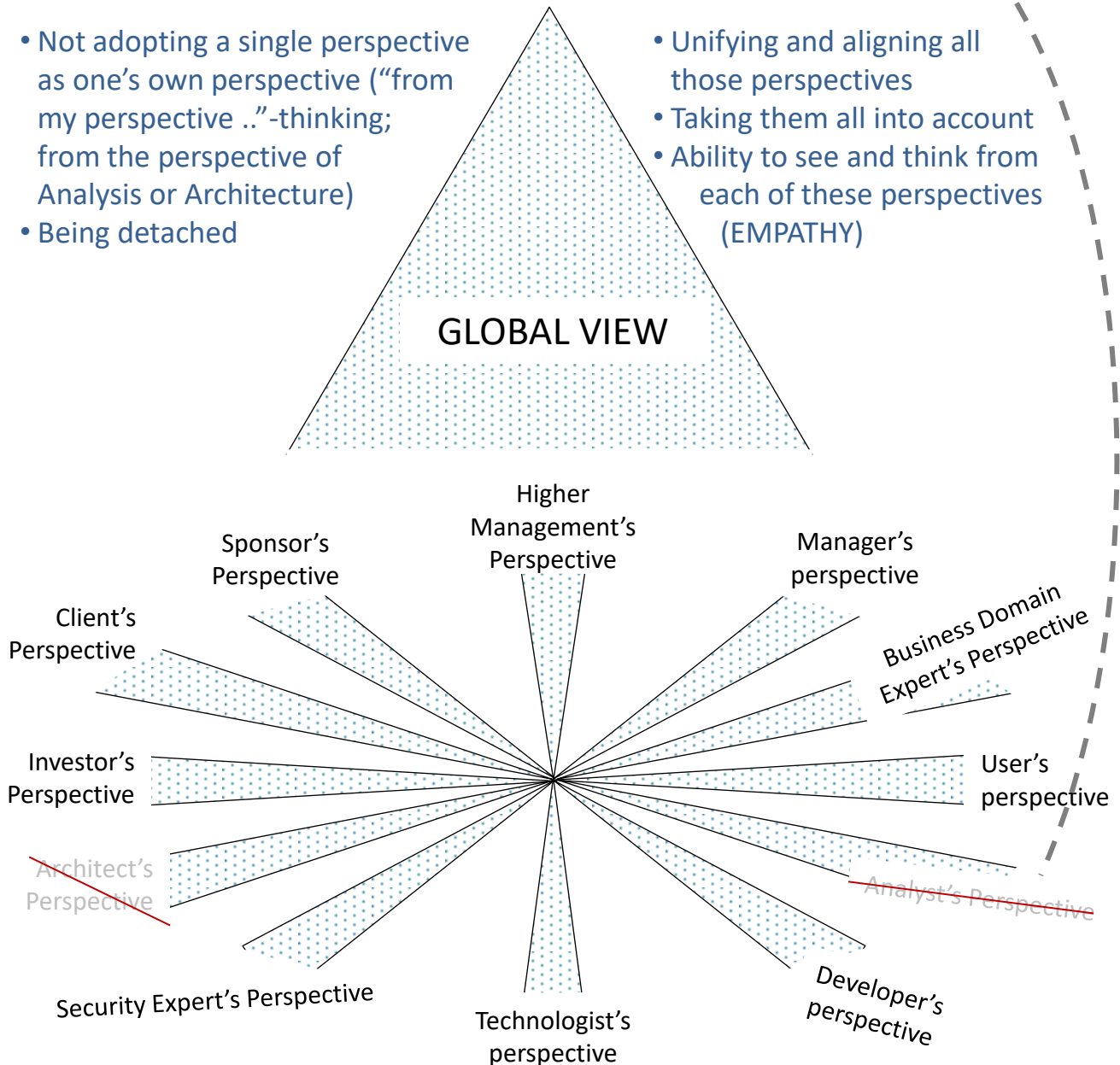
Architects & Analysts Above the Frey

10/01/2020

Architect's & Analyst's Perspective

- Not adopting a single perspective as one's own perspective ("from my perspective .."-thinking; from the perspective of Analysis or Architecture)
- Being detached


- Unifying and aligning all those perspectives
- Taking them all into account
- Ability to see and think from each of these perspectives (EMPATHY)



A Problem Solver a Solution Creator can not do his/her job if she is part of the problem. (S)He has to position himself/herself outside and above.



FUNDAMENTAL TOPICS

1. SYSTEM
 2. COMPANY
 3. INFORMATION
 4. PROCESS
 5. PROBLEM SOLVING
- 



- SYSTEMS -



A more
profound understanding of systems
is required to

- conceive systems
 - use systems
 - manage systems
 - deal with systems in general
-

Notes:

A lack of understanding in how a system works, its constraints, prerequisites, right handlings, limits, risks and anything that may harm the system may damage the system and even destroy it.

Modifying a system, ignoring or not understanding the effects on other systems downstream, may create problems in these systems (domino effect).

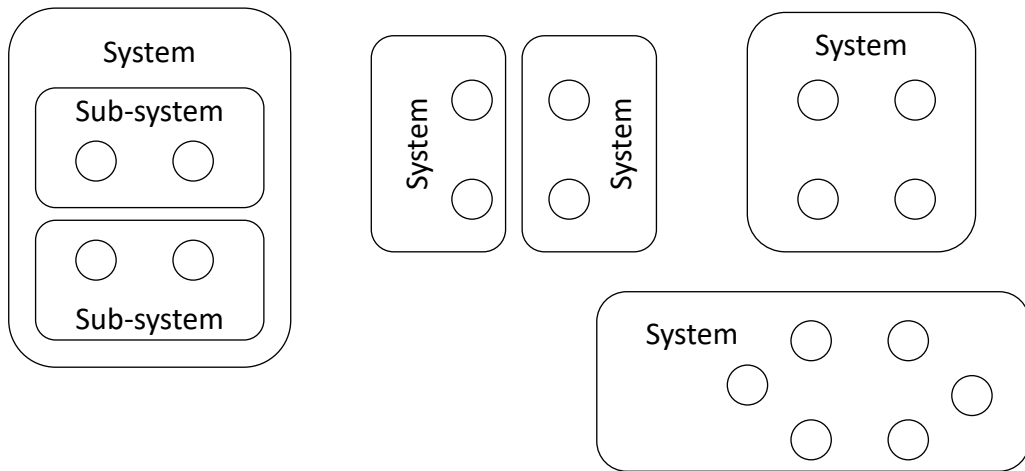
In software development, this risk is increased if information is shared (several systems accessing the same databases).

Identifying a System

30/10/2018

The concept of “system” is a method to structure and organise the world to help the human mind to understand this world.

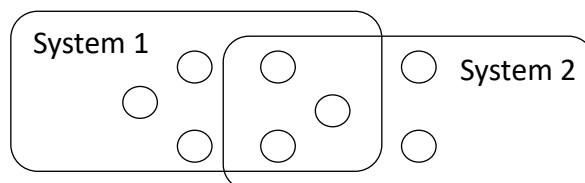
A system can be determined by selecting sub-systems and grouping the parts in many different ways.

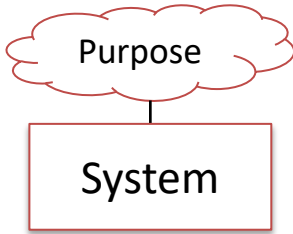


The purpose, the goal, the collaboration, the cohesion, the logical domain, the spatial and natural ‘boundaries’ and how it is best to be managed are good criteria to define a system and its boundaries.

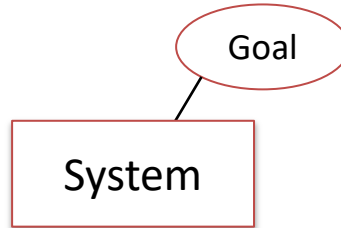
What is true for a system, is also true for supra-systems, sub-systems and co-systems.

Overlapping Systems





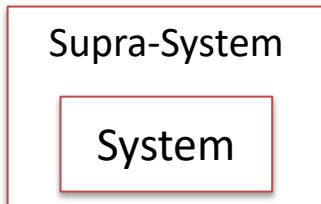
has a purpose, a role



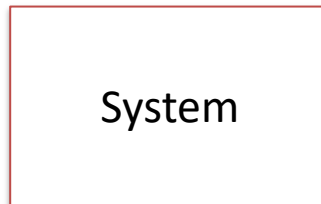
has a goal



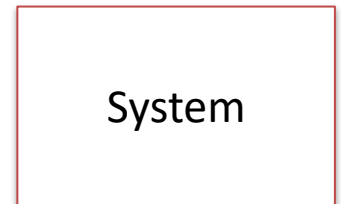
has a boundary



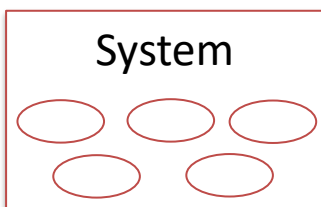
can be part of a supra-system



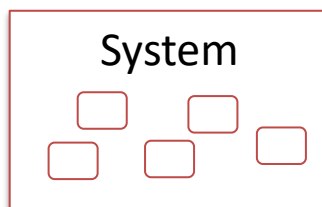
has characteristics



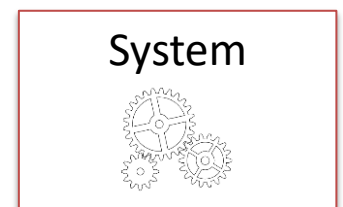
has limits, weaknesses, critical parts, ...



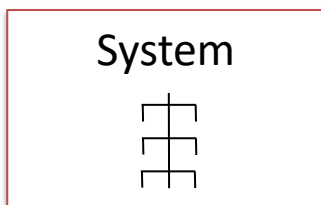
has capabilities



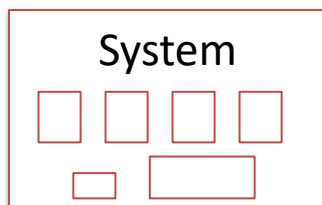
has functions



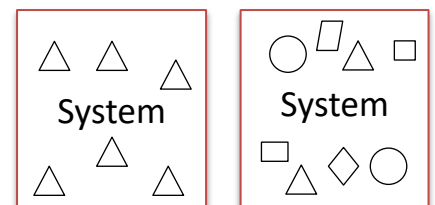
has a behaviour, mechanisms



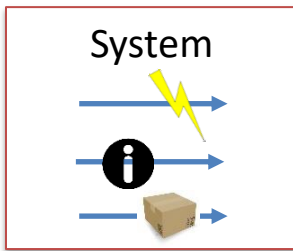
has internal structure, organisation, architecture



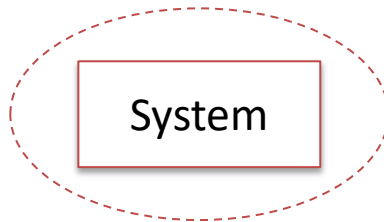
has sub-systems and components + interactions among them
Note: Sub-systems are systems (co-systems)



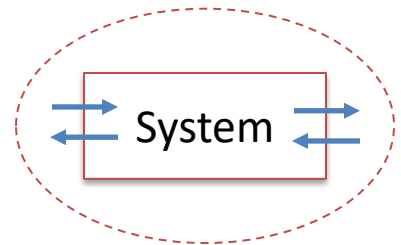
can be homogeneous – heterogeneous (components of different nature)



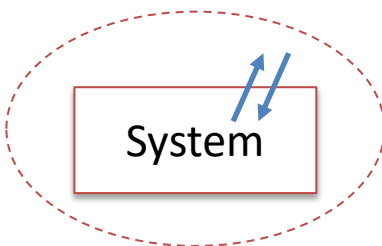
flows of matter,
energy and
information



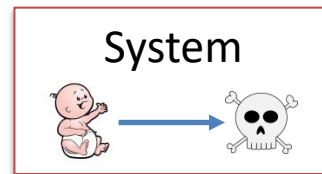
exists in an
environment



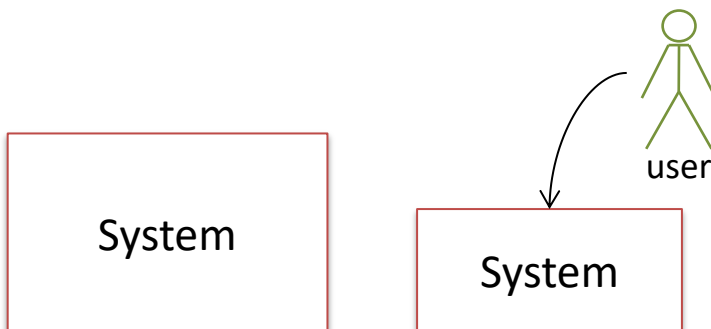
exchanges and uses
information, energy
and (often also)
matter



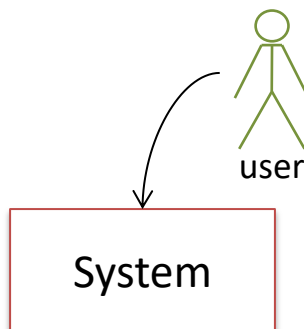
Influences, forces, exchanges and
interactions with environment.
Even its existence, the fact of
being present, has an influence on
the environment.



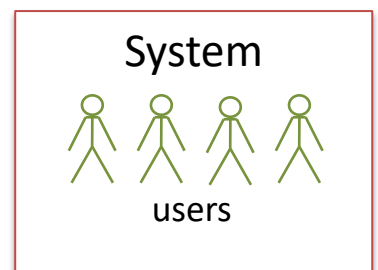
is temporary,
has a life (full of events), has a
lifecycle (even if the end is
not foreseen)



can be
autonomous



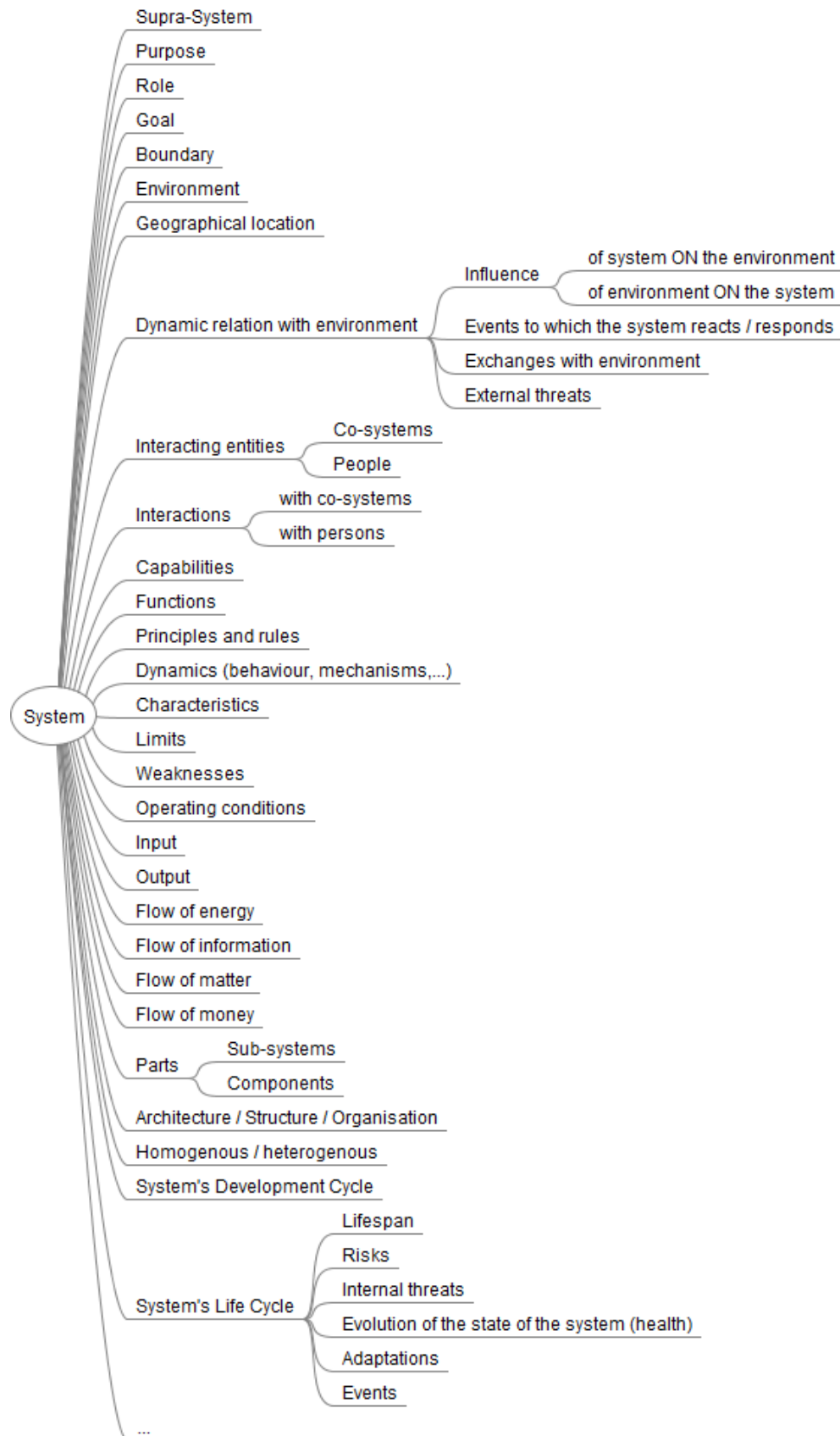
can be used (by 1
or more users or
users groups)



may contain users

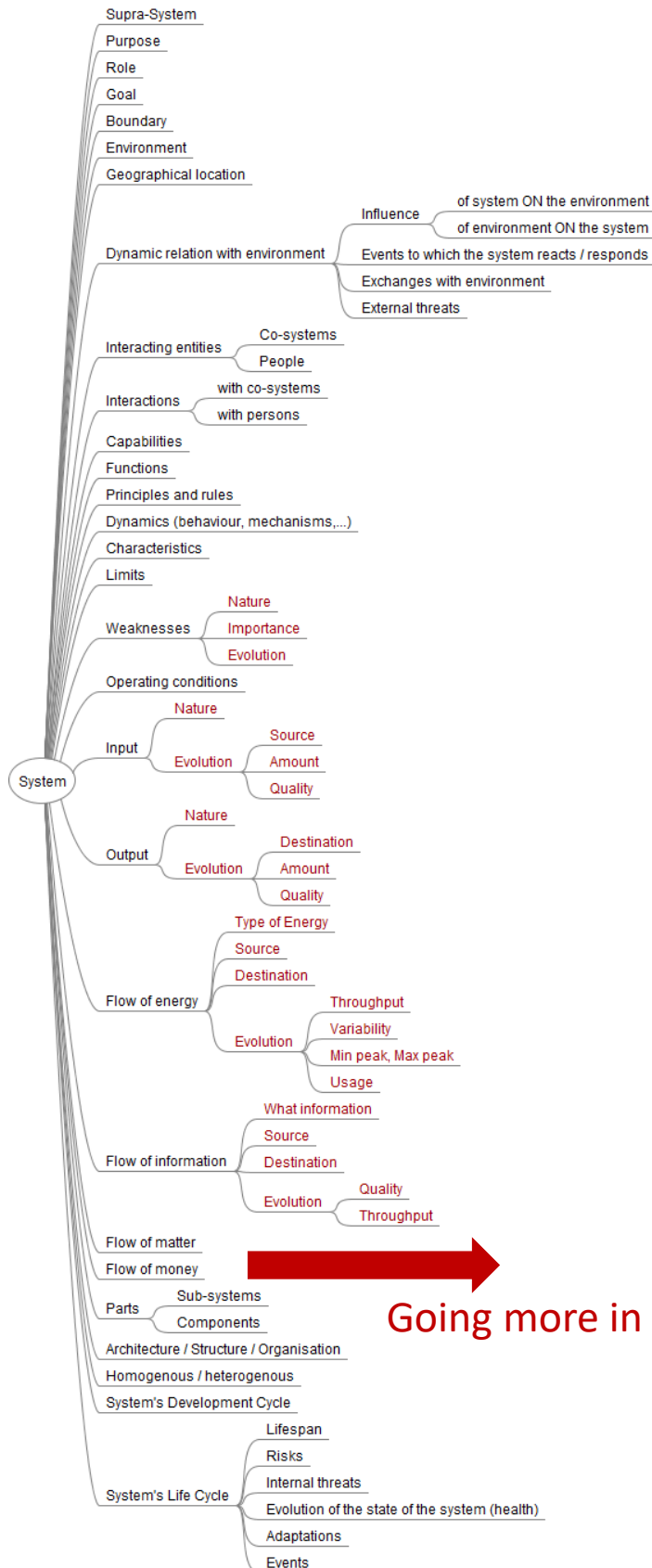
System's Characteristics

30/10/2018



System's Characteristics

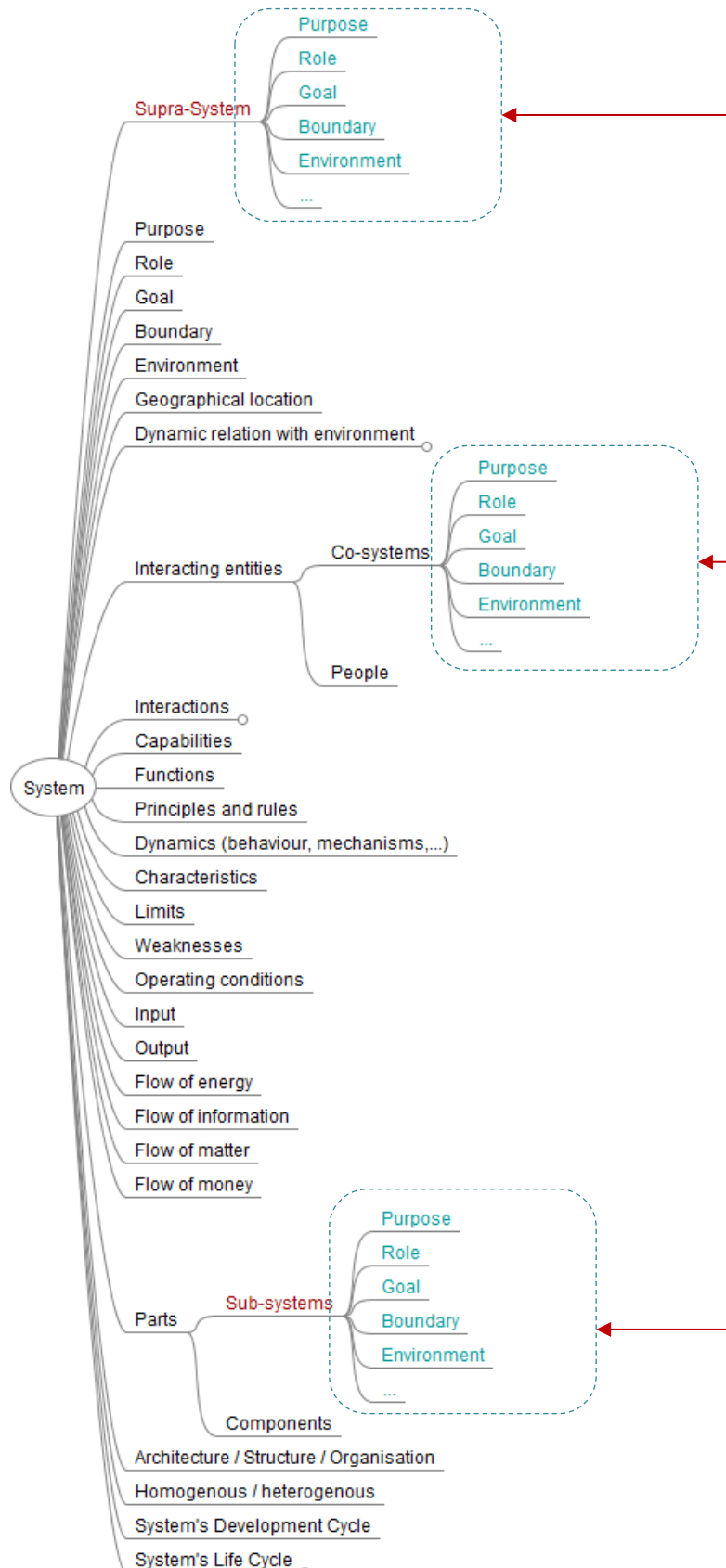
30/10/2018



Going more in depth

System's Characteristics

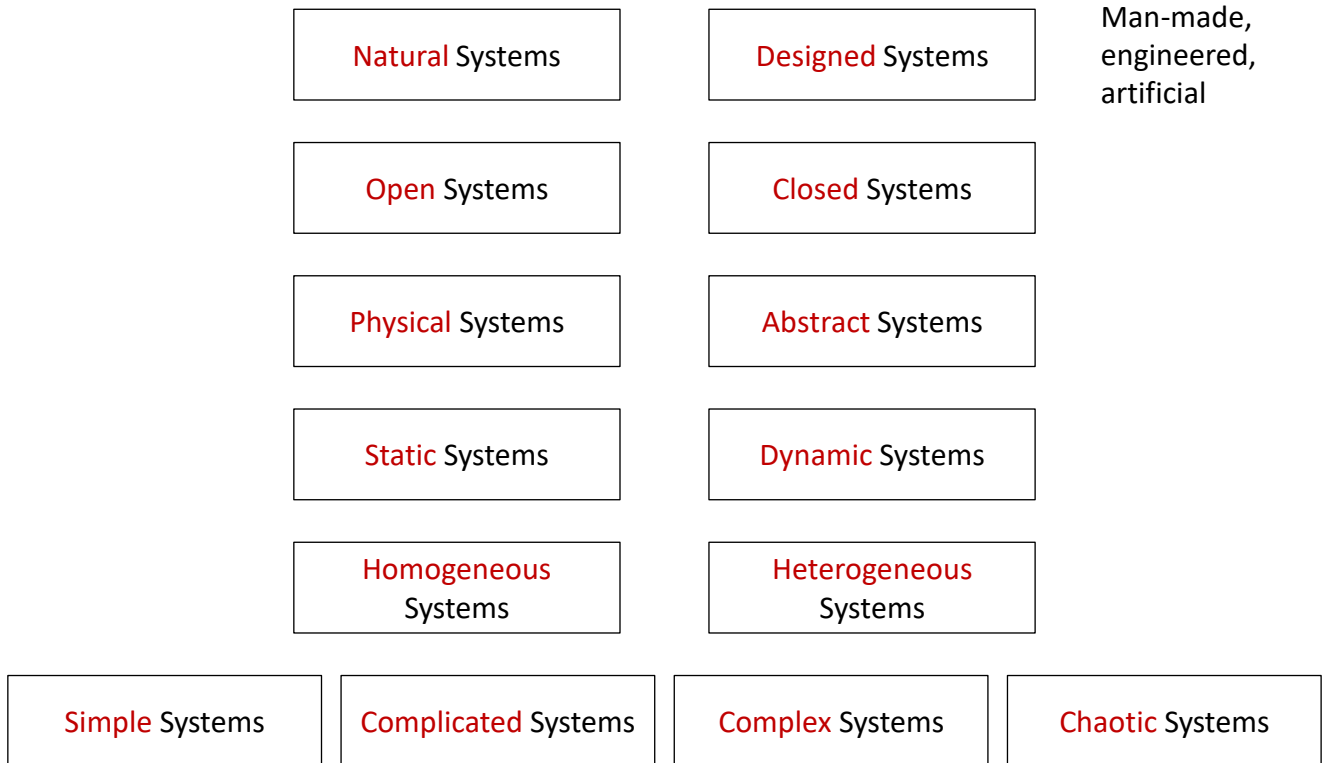
30/10/2018



NESTED
Different components
are similar.
Groups of characteristics
may be repeated.

Classification of Systems

10/01/2020



Other possible classifications:

- a) Linear, non-linear systems
 - b) Time variant, time invariant systems
 - c) Stable, unstable systems
 - d) Causal, non causal systems
 - e) Continuous time, discrete time systems
 - f) Invertible and noninvertible systems
 - g) Fixed, mobile systems
 - h) Predictable and unpredictable systems
 - i) Manual and automated systems
- Social systems

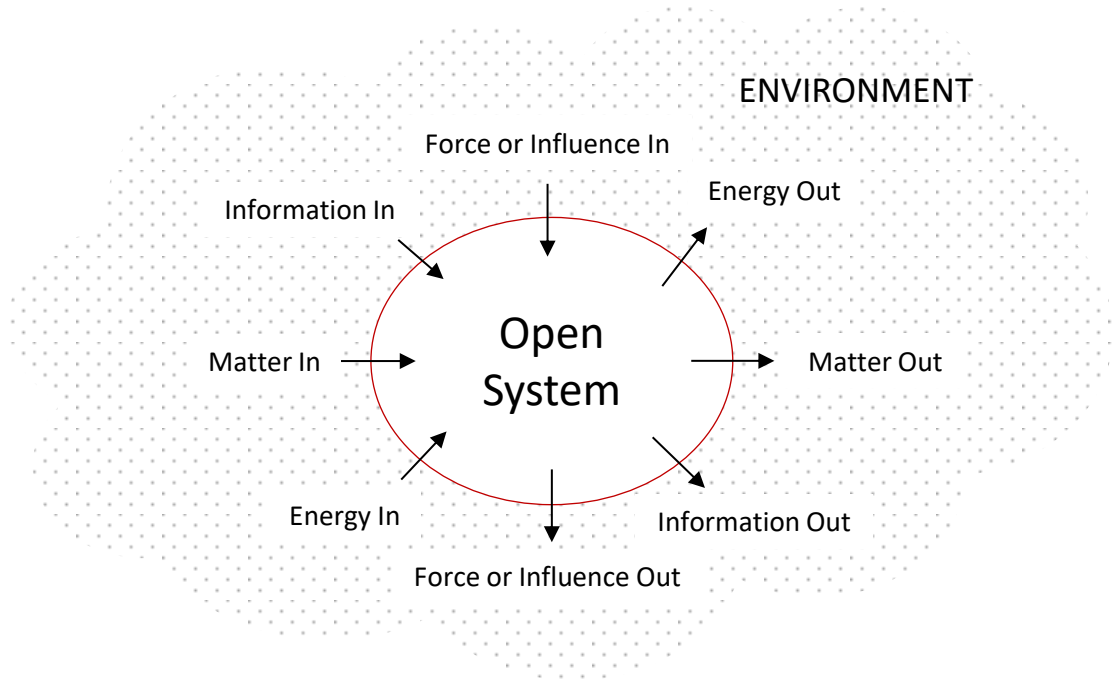
Note:

...

Not all man-made systems are engineered. Just like we can't call any creation "art", not all system creation can be called "engineering".

Sometimes, gradation (not always black or white): Is a forest created by people a natural system or a designed system ?

Any system's characteristic can be used to classify systems



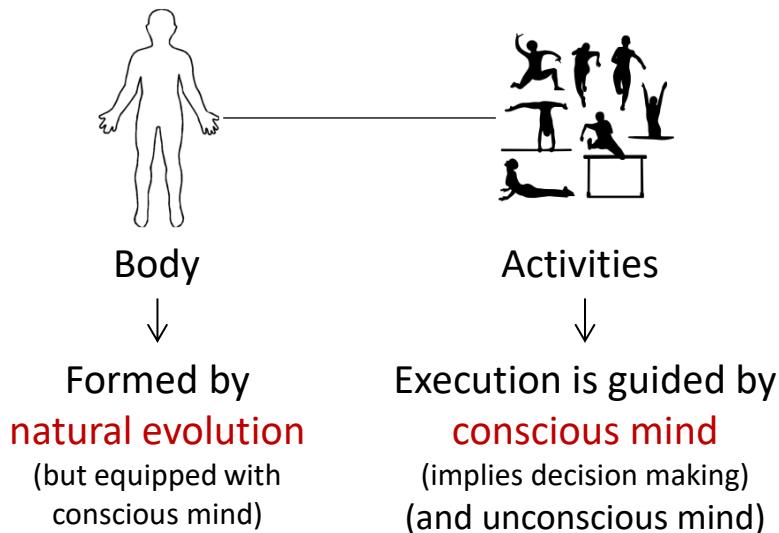
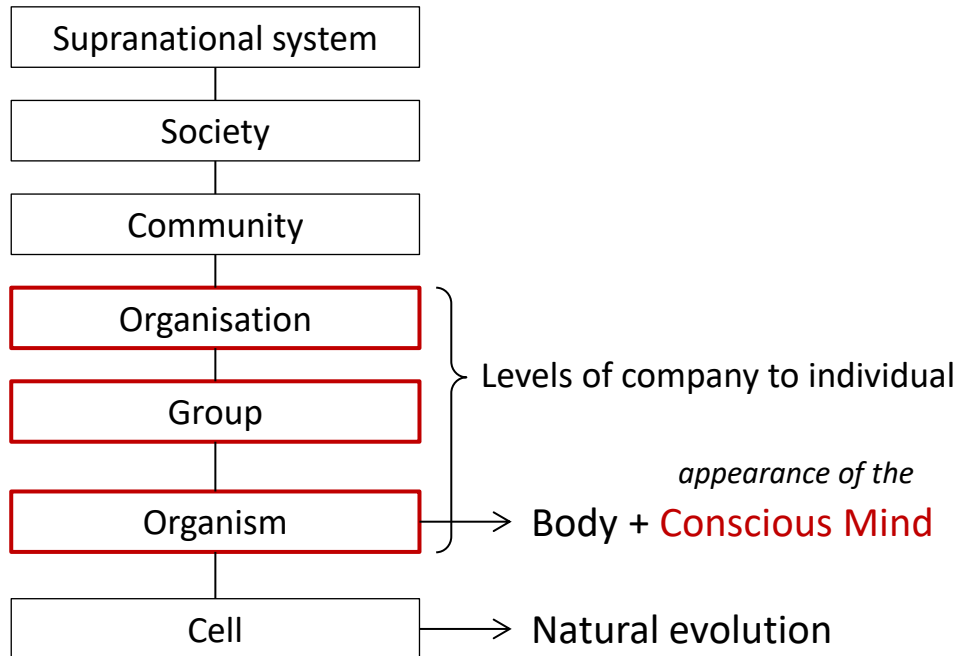
An Open System is a system that interacts with its environment.

Not all 'in' or 'out' have to be present.

Matter and Energy are not created out of nothing.

A LIVING SYSTEM SEEKS TO EXIST INFINITELY AND TO PROSPER

Hierarchy of living systems



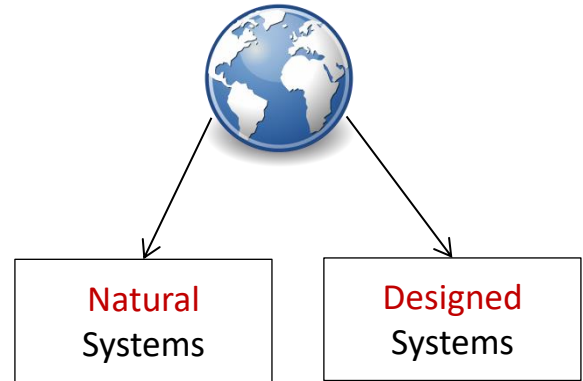
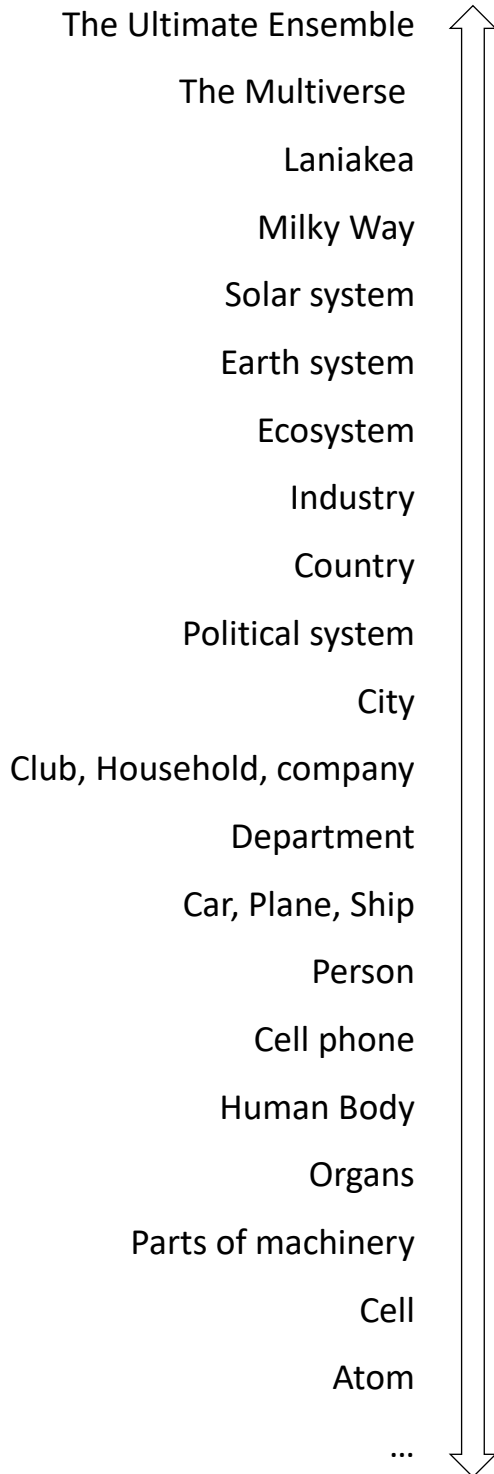
Notes:

Decisions (intellect, mental, conscious mind) and activities may shape the body somewhat. But basically, it's a product of natural evolution.

A company is a living system. However, this does not mean that all its sub-systems and parts are living systems as well.

Scale of Systems

30/10/2018



*We are **interested** and focus on **designed open dynamical systems (artificial, engineered, man-made).***

*We **can LEARN** from **NATURAL SYSTEMS***

BUT

*natural systems and artificial system **are different.***

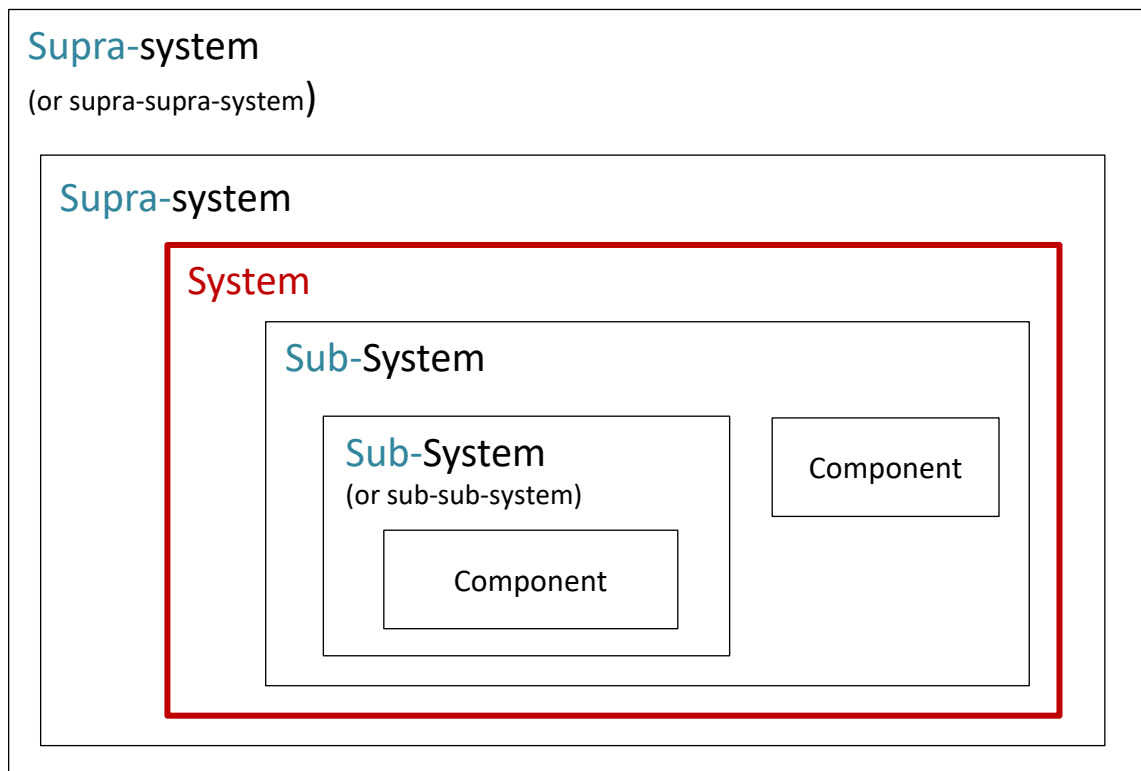
Therefore, not all principles, concepts and methods applicable on natural systems are valid, applicable or desired on man-made systems!!

Hierarchy in Designed Systems ^{30/10/2018}

A system is a view of something having a set of characteristics defining a system (boundaries, components, internal interactions, ...)

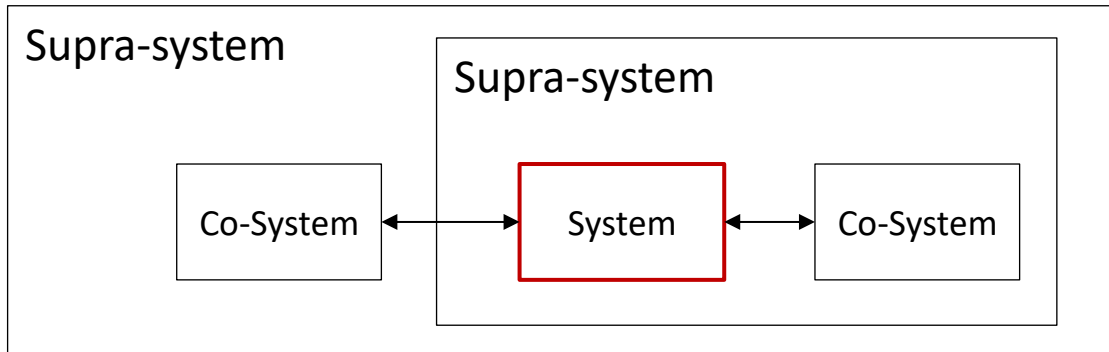
Supra-Systems and sub-systems are all systems.

In the hierarchy of systems, a system is considered. Systems inside this systems are then sub-systems. The system(s) to which the system under consideration belongs are the sub-system.



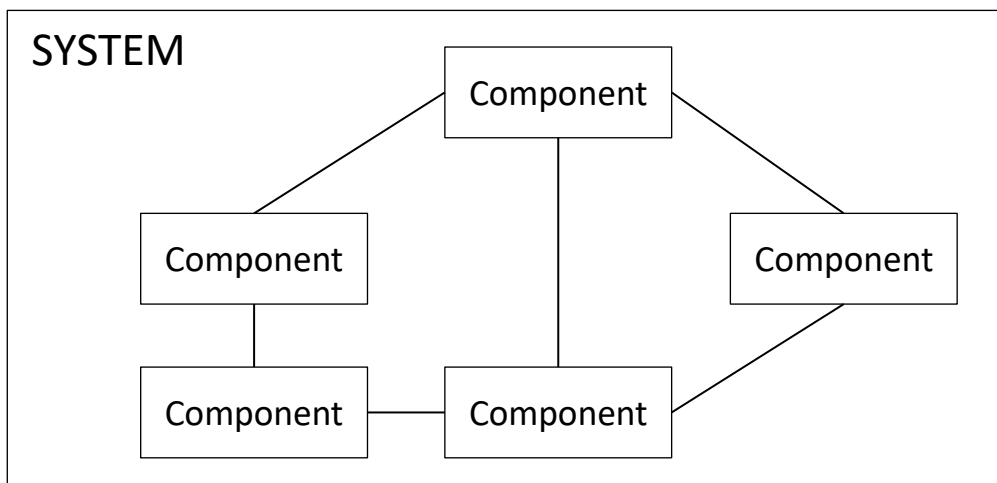
1. Sub-systems and components are within the system's boundaries and contribute to the system's goals, be it actively or passively (ex. back-ups). Else, they are superfluous.
2. A hierarchy can have many levels of supra-systems and sub-systems.

Hierarchy in Designed Systems ^{30/10/2018}



A co-system is a system at the same level as the system under consideration. Often there is an interaction, an exchange, a relation among them. They can be part of the same supra-system (a system is a view).

A system and one or several co-systems, together, can also be seen as a system.



- Components and sub-systems are connected. This structure is one of the architectures.
- These systems and components interact.
- They collaborate towards a common goal.
- Together, they implement the system's capabilities and qualities.

Hierarchy of Systems

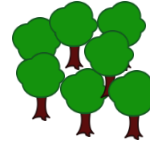
14/02/2020



a heap of blocks



a single block



a forest



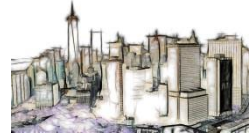
a tree



a group of people



a person



a city



a house or building

Forest

A forest is more than trees. It's an ecological environment. It's a whole dynamical environment.

Group of People

A group of people is more than a number of people placed together. Communication & group dynamics.

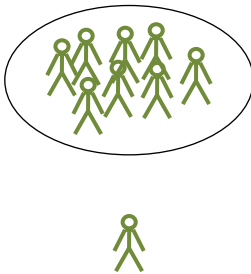
City

A city has road infrastructure, emergency services, communication network, energy supply, ...

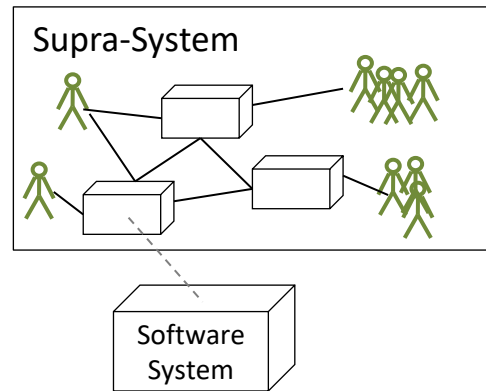
**A cluster, grouping, aggregation of similar systems
may form a new supra-system**

- A block is not a system. A heap of blocks doesn't form a new system.
- Not every collection of systems forms a larger system (supra-system).
- We may easily perceive a group of systems as a collection of systems. Our focus is directed to the systems individually. And so we may fail to consider consciously that the collection of systems can also be a system. They may form a system of systems or a supra-system. This is not without consequences in Systems Design.

Nature of the Supra-System differs from that of the System

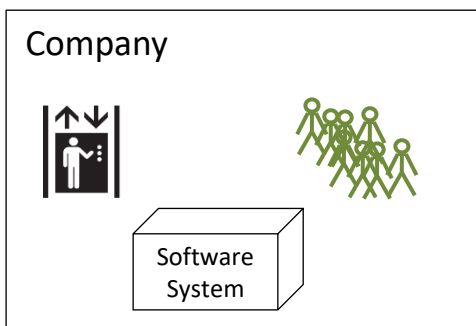


A person is a system. A group of people may also form a system. A group of systems is a very different type of system than the individual.



A Software System is a system with specific characteristics. Its supra-system may include other software systems, but also users (persons). Users, and group of users (or organisational units) are systems which are very different from the software system.

Nature of the Sub-Systems differ from that of the Supra-System



A company is a socio-technical system.

But this does not imply that its sub-systems and components are socio-technical systems.

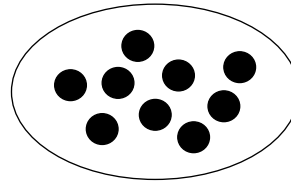
An elevator and software systems are systems of a very different type.

Characteristics, laws, rules, principles and methods applicable on a supra-system are not necessarily valid for all its sub--systems and components. And vice versa.

We may not assume supra-systems or all sub-systems to be similar to the system.



Individual
system



Group, collection, ensemble, cluster,
aggregation of similar individual
systems forming a supra-system

A Supra-System is formed:

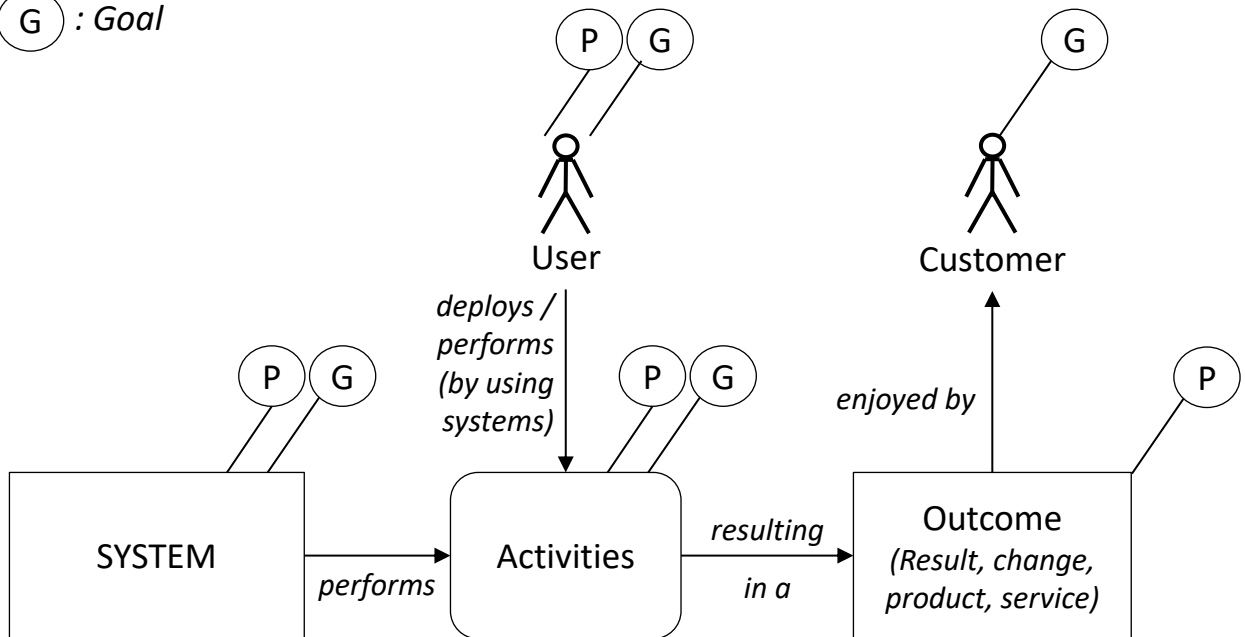
- has its own purpose
- is of different nature than the constituting individual systems (tree, person, house).
- has relations among the individual systems. They interact with each other.
- has its own characteristics.
- Obeys to its own principles, rules or laws
- Has its own dynamics (mechanisms, forces, evolution, behaviour)
- contains components or systems other than the individual system.
- has its own issues that doesn't exist at the level of the individual system

Limits of Scalabilities in Approaches and Methodologies

We can not solve problems of a higher level with a thinking (and approaches) of a lower level. The level of thinking (or maturity) used to design a philosophy, an approach, a methodology, ... determines the limit of its scalability.

(P) : Purpose \approx Function \approx Role ($\approx \neq = !!$)

(G) : Goal



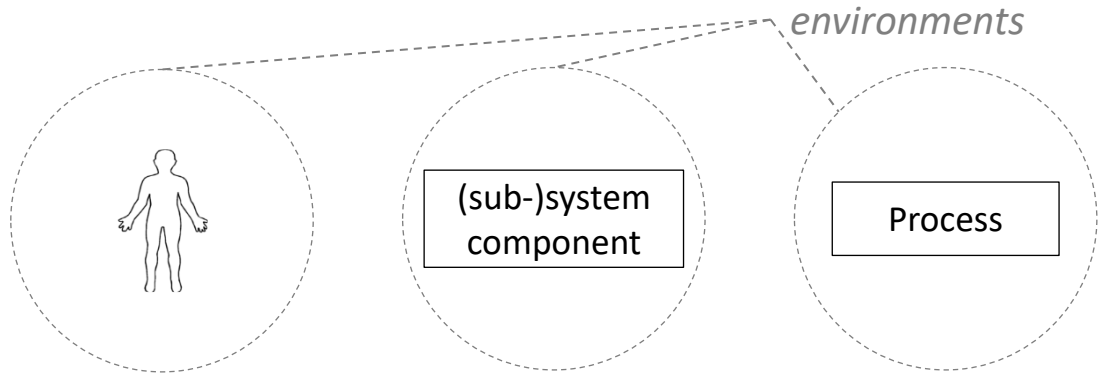
- A **customer** has goals (and/or Needs)
- The **product** has a purpose.
- **Activities** have a purpose (meaning) and their goal is to produce or to contribute to the production of a certain outcome.
- A **user** has a goal. The user, as part of a system, has also a role (a purpose).
- A **system** has a purpose. It has also some goals.

Purposes (,roles) and goals have to be aligned.

Absence of alignment → Problematic

Notes:

User and customer are roles. They may point to a group of people. A single person can have different roles.



A person, a group, a team, a project, a community, a society, a component, a (sub-)system, a process **exist in** an environment.

- They are **influenced by** the environment.
- They **influence** the environment.
- They **depend of** the environment.

Their existence, their thriving and survival depends of their environment: how well they are **adapted** to it; how well they are **accepted** by it and/or how well they **serve** it.

We can't function, decide, solve, conceive, heal, restore, guide, build, grow, ... while ignoring the environment(s).

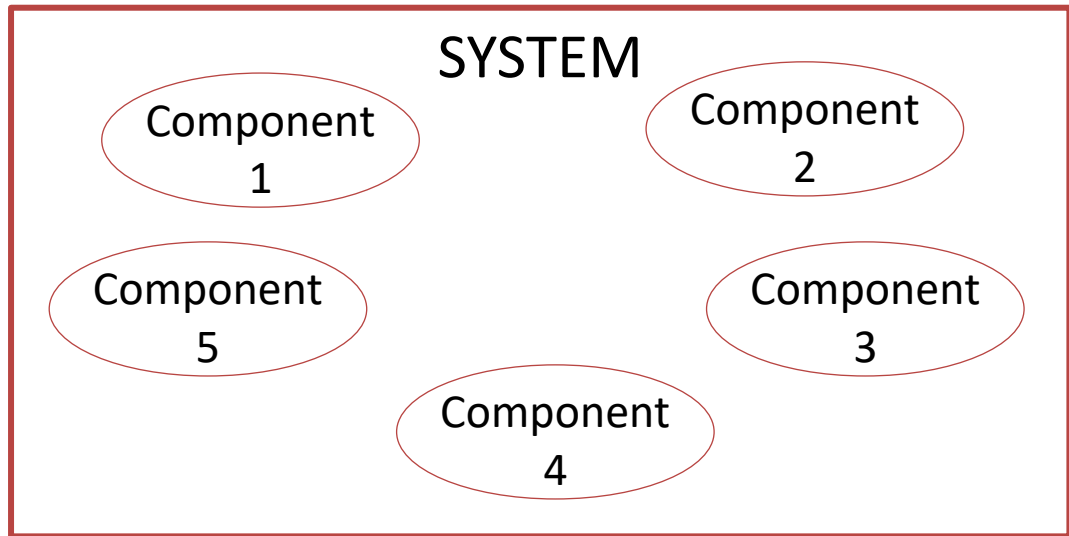
We can't do without understanding the environment.

See, consider, focus also on the environment...

SEE the ENTIRE picture

We can build the nicest and most advanced submarine. But it will be completely useless if it is in the desert. And building it with little knowledge of the ocean (the environment is a recipe for failure).

Achieving a well-functioning system



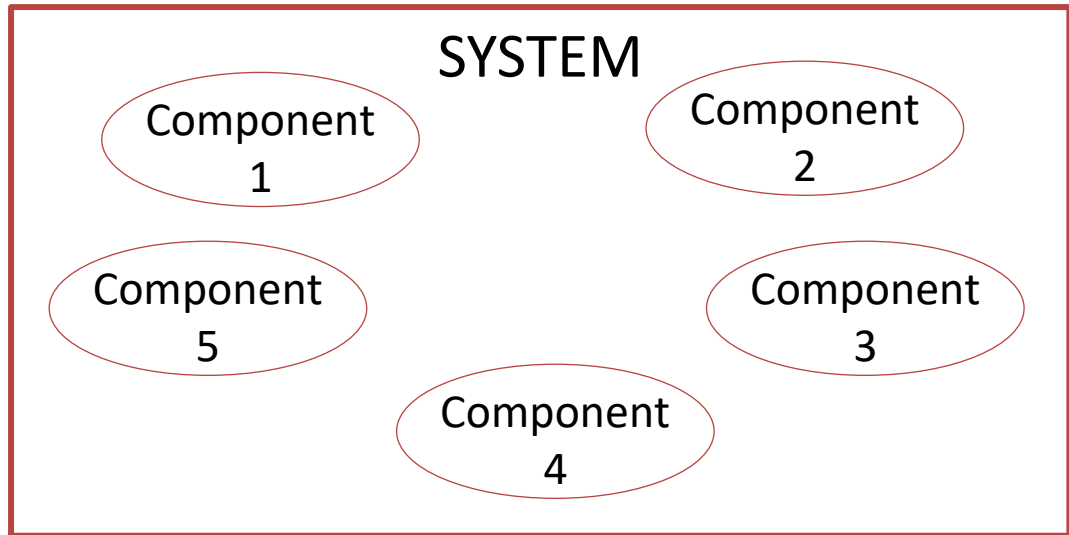
What is HARMONY?

- Components form together a larger unified whole
- Optimised contribution of every part to the whole's interests
- Environmental fit
- Functional alignment
- Alignment of capabilities
- Optimal coordination
- Coherent logic
- Balance in behaviour, capabilities and in physical qualities
- Efficient and effective collaboration
- Internal logical, ordered and balanced structure
- Systemic Integration
- Mutual accommodation of the components
- Optimal use of resources (*including information*)

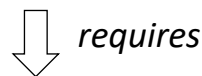
How to achieve harmony in a heterogeneous system of systems (organisation) ?

This quality of systemic unity is a **crucial** aspect of any stably perduring whole.

Achieving a well-functioning system



Systemic harmony is achieved when different parts of a system form a harmonious whole and work seamlessly together. This unity is crucial for the long term stability and existence of the whole.



COLLABORATION among engineers

The organisation must structurally facilitate the collaboration among the engineers & company must be structurally integrated.

Note:

No or difficult collaboration → no well-functioning system → problems

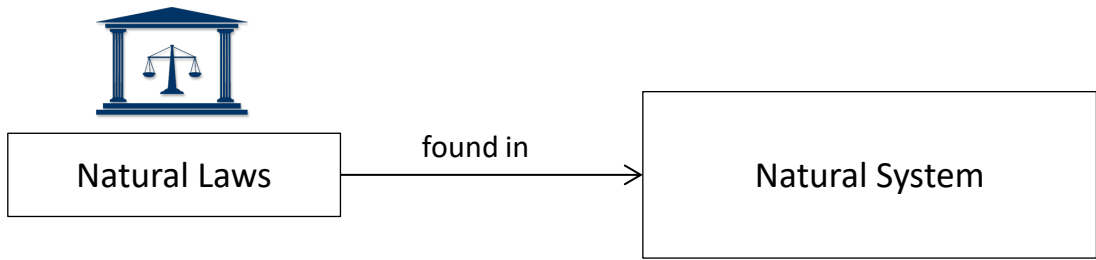
The collaboration is not restricted to a collaboration between the engineers only.

Natural Laws and Natural Systems

30/10/2018

Natural Laws

- Laws of physics
- Laws of biology
- ...



Nature obeys only to its own laws.

A natural system always acts and reacts

- accordingly to the natural laws
- and accordingly to its own nature, its logic, its mechanisms, its capabilities, its characteristics, ...

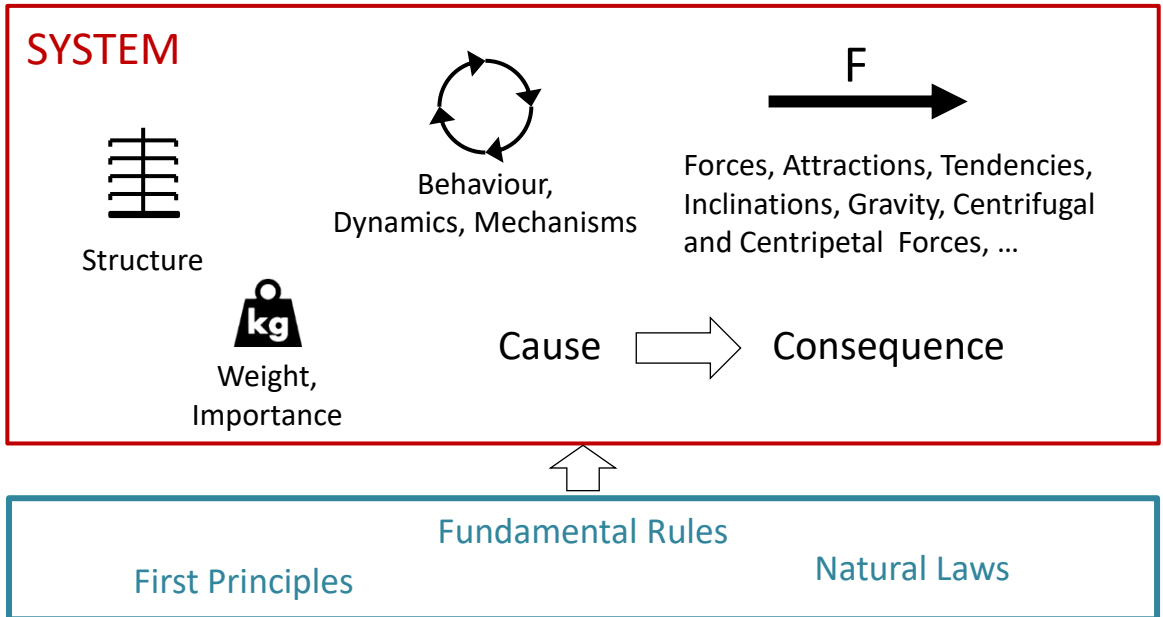
Breaking natural laws and disrespecting natural systems beyond the point of resilience, inevitably leads to consequences or at least to increased risks.

Nature is an excellent, but intransigent, teacher.
It lets you fail until you learn and respect its laws.

Systems kill us if these laws are violated

Equilibria in Systems (1)

30/10/2018



The respect of these principles, laws and rules is critical for achieving equilibrium

Where do these principles, laws, rules come from?

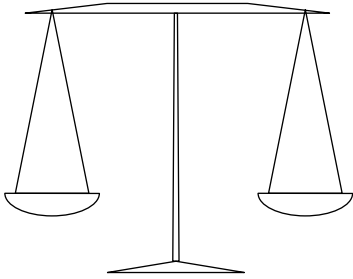
- 1) Find the elements that makes up the future solution and its environment
- 2) Identify the disciplines related to these elements
- 3) Or study the nature of the elements

Element	Laws / Discipline
People	Psychology, organisational behaviour, ...
Computers	Computer science
Money	Financial management, Economics
Electricity	Science of Electricity
Information	Information Science
Metal parts	Materials science
Water	Fluid Mechanics, Hydrology, ...
...	

+ Systemology + Engineering + Problem solving + ...

Notes:

To be further investigated



Dynamic Systems, and their components, have a kind of **balances**. There are parameters of which the value must be within a specific range to achieve a **state of equilibrium**. These balances belong to and are formed by structures, behaviours, forces, weights and cause-consequences. Since these balances are part of the same system, they might be **connected** and **influence each other**.

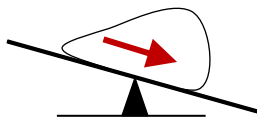
THIS IS ABOUT THE **WELL-BEING OF THE SYSTEM**

A value may change, but as long as it remains in the acceptable range, the state of equilibrium is achieved. Some values must stay above or below a certain limit. Or, values have to exceed a certain threshold. There are trigger points or a critical mass setting the system, or a part of it, in motion.

Example:

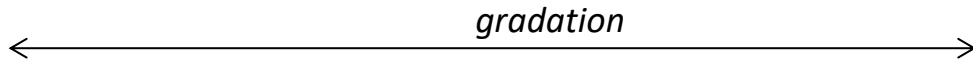
- Speed must be kept in a range. It may not exceed a limit. Sometimes it must be zero.
- An energy supply must be kept within a range. And it must be continuous.
- The mass (weight, or volume) has to be spread across the system.
- A bearing structure has to be able to support a certain weight.
- A bearing structure has to be stable and to mitigate bumps, shakings and other lateral forces.

A disequilibrium may hinder a system from functioning properly, or even causing its collapse. A system can also become unstable or go in an uncontrolled or abnormal state of hyperactivity. It may react in many unforeseen ways.



Many states of equilibrium and states of well-being of the system have to be identified, understood, managed and the necessary protections need to be built in. It is the task of the engineer to detect them in existing systems as well as in the systems being designed. This is not an easy task for engineers and, a fortiori, for end-users (in existing systems).

Critical – Important – Useful – Optional



System's Perspective

Structures, principles, mechanisms, components, capabilities, capacities or resources important for the **survival**, well-being and thriving of the system

User's Perspective

Capabilities, features, capacities, outputs important for the **usage** of the system

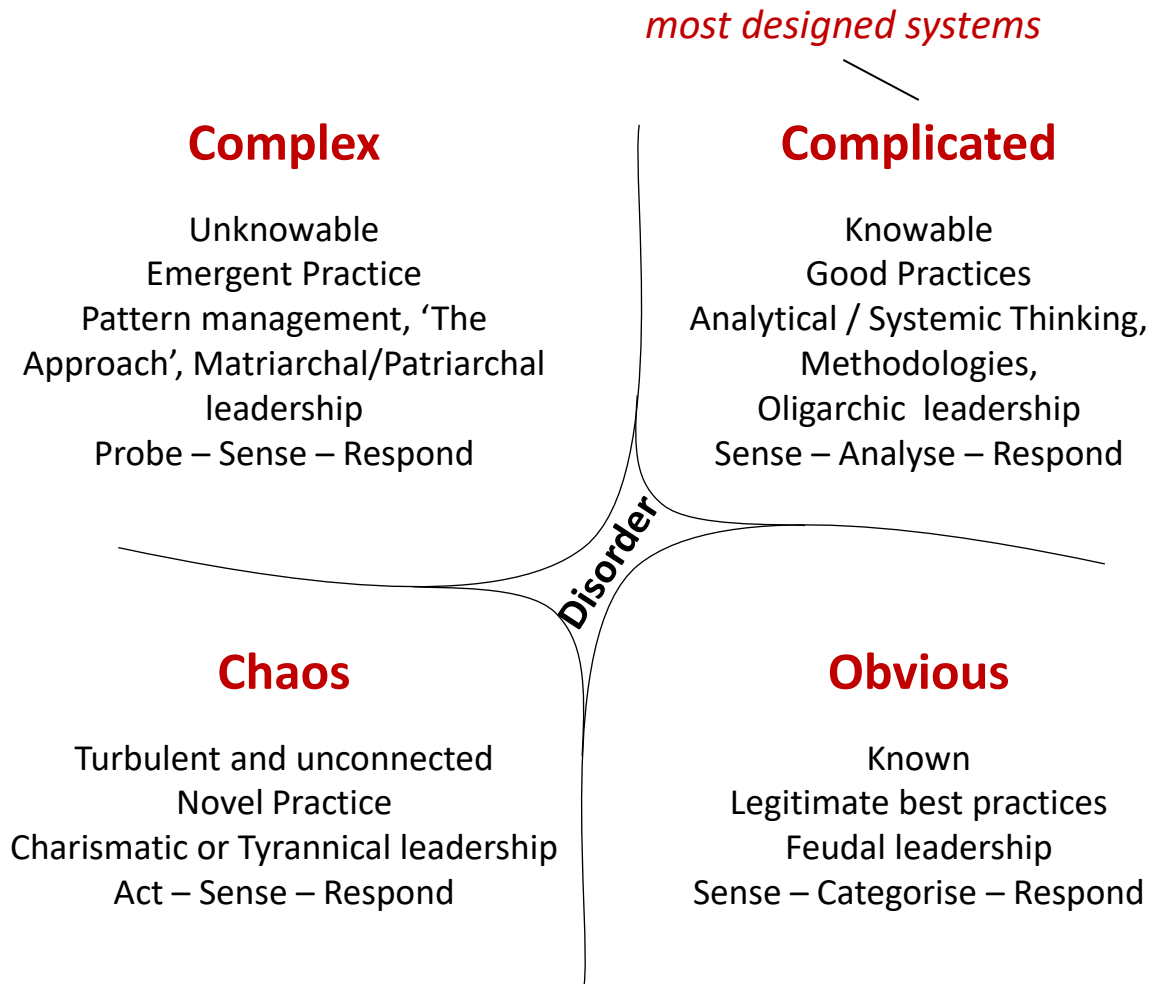
Notes

Example: A system may be useable, but without internal controls and maintenance, over time, the system will degrade.

The more important the item or aspect, the more attention of engineers it deserves.

Cynefin Framework (“KUN-iv-in”)

15/03/2019



Inventor: Dave Snowden

TRAP:

It is too easy to qualify a system as a complex adaptive system (CAS) based on the personal inability to learn, to analyse and to understand it or in order to avoid the effort of acquiring the understanding it.

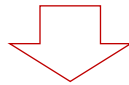
Systems Analysis has been developed in order to understand more complicated systems.

Functions of Information Systems and Software Systems

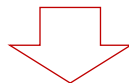
10/01/2020

Source of Inspiration for New Services and Innovation

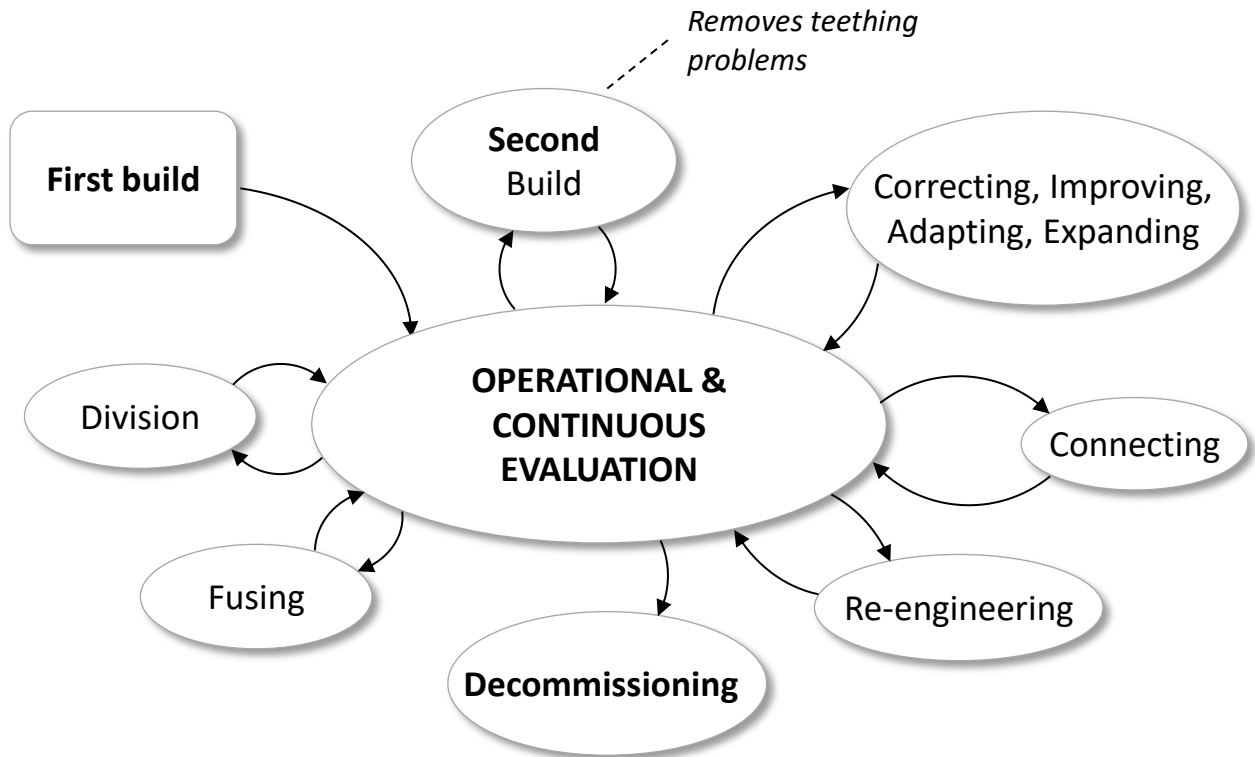
Capturing, controlling, processing, storing, securing, transporting and providing Information



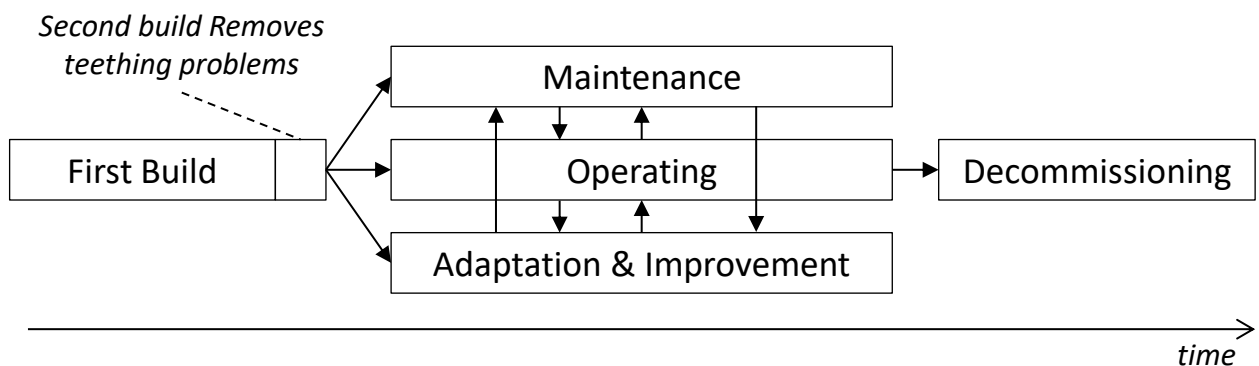
- Performing tasks of people (related to information) in a much faster, unvarying and reliable way.
- Amplifying Human Capabilities (good and bad)
- Relieving people by performing its activities
- Controlling people
- Guiding people
- Preventing people from making mistakes
- Informing people



entertaining ppl, influence ppl, drawing the attention, interest ppl, attract ppl, assist them in taking decisions, providing choice, teaching ppl, assisting in thinking, ...



Systems have to be able to cope with events happening during their future life. They have to be engineered for it.



Huge changes, such as re-engineering, may also happen. This can be considered as a first build of a new similar system.

Evolution by Natural Design

~ DESIGN
DARWINISM

- Purposeless
- Foresightless
- Randomness, directionless
- Slow
- Unpredictable (unsure 'if', 'when', 'what', 'value', 'meaning', ...)
- Extremely costly (Waste on the many failed trials)
- Produces organism suitable to the environment
- but in the end ... it is brilliant
- Happens by many clueless minds in a very huge amount cycles (iterations)

Intelligent Design (Artificial Systems)

- Purposely (goal oriented)
- Mainly top-down
- More foresighted
- Directed search
- Relatively fast
- Governed by cost considerations
- Beautiful designs is done by great minds

Notes:

Today, we can simulate evolution in computers (huge amounts of iterations are possible; genetic algorithms, deep learning)

Intelligent design matches better the world of the company. In specific cases, natural design can be applied, or as a combination of both.

Additionally, we can use natural evolution by getting inspiration from nature (biomimetics or biomimicry). It all still requires insight and intelligence.

The whole is greater than the pieces.

Why?

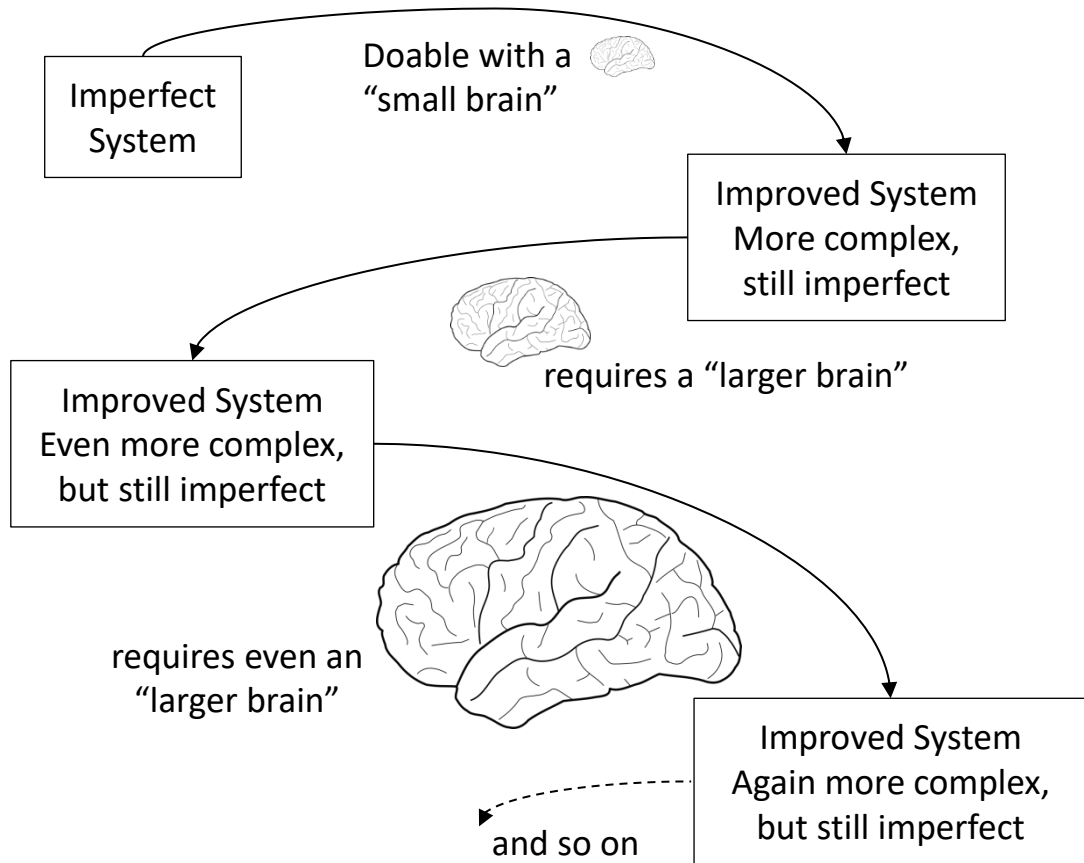
- Having a common goal
- The implementation of an overall concept
- Implementing the system's capabilities and characteristics
- Forming the architecture (structure and order)
- Interdependency among the parts
- Alignment among the parts
- Interaction: communication, exchanges, transfers among the parts
- Collaboration and support among the parts
- Components may share same elements
- Shared environment (influencing the environment, reacting to same changes in the environment)
- Sometimes built on same foundation or container
- Sharing the same infrastructure
- Common principles, policies, rules, standards ...
- A limit or risk of one component is a limit or risk for the whole
- Sharing resources

Implications:

Building pieces and assembling them (bottom-up) without overall design, is likely to lead to a lower quality of the whole.

Over time, by repeating this many times, it leads to a chaotic result.

Important for integration



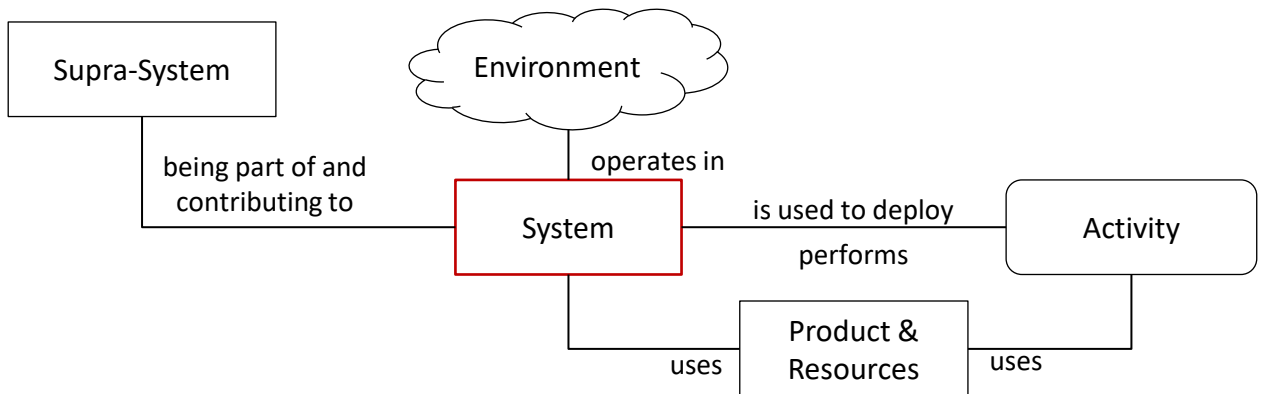
If a bigger brain is required, it may imply a maturation step (change of perspective, improved belief system, paradigm shift, ...)

How to combat this tendency?

- Don't repetitively add layers, cases, exceptions, rules, criteria, differentiations, ... → this adds complexity
- Get back to the fundamentals, first principles
- Clean-up, re-engineer, re-simplify

Neglecting this will increase the effect of the tendency of complexification

Five major sources determining the capabilities and characteristics



Environment

Structure, capabilities, mechanisms and characteristics

- adapted to the environment,
- able to deal with the environment,
- allowing proper interactions with the environment
- fitting in the environment

System

Structure, capabilities, mechanisms and characteristics

- supporting effectively the purpose and goals of the system
- able to protect the system (verifications, monitoring)
- able to maintain harmony (monitoring, measuring, controls, management, avoiding misuses and abuses, ...)
- allowing the system to survive during its lifecycle (lifetime) (adaptability, scalability, ability to deal with unforeseen events, ...)
- able to maximally exploit the resources
- able to maximally create value for the environment and/or for the supra-system

Supra - System

The Supra-system combines the three others (environment, system and activities). The Supra-System also provides a context of needs and requirements.

Activity

Structure, capabilities, mechanisms and characteristics

- able to perform the necessary activities in line with their overall purpose and goals

Product, Resources, Raw Materials

Structure, capabilities, mechanisms and characteristics

- able to produce the intended product
- able to optimally process and exploit the resources, the raw materials

Defining the System's Environment

10/01/2020

System's survival depends (also) of their environment.

Therefore, the definition and the understanding of the system's environment is critical to design meaningful systems that can survive in it.

The environment is defined by the elements and the space between these elements

- with which the system interacts
- able to influence the system
- of which the system's survival depends
- impacted by the, present or future, system

one way or another, in the present or future: input, output, constraints, obligations, forces, opportunities, exchanges and interactions, impacts, ...

The more perspectives (holistic) are taken into account, the more meaningful the defined environment is. The environment is **not static and may evolve over time**.

The objective is not to identify the perfect exact environment, but to identify what is meaningful in order to be able to study it and to take it into account.

A good starting point is to identify the elements with what the system interacts and elements that are shared. The study should go much further.

The “**boundaries**” can be solid, impenetrable, penetrable to different degrees, clear or fuzzy. It may vary in time.

The better the insight of a person is, the better (s)he will be able to define the environment. The better the environment is defined, the more valuable insight can be acquired. It's 'definition doesn't need to be formal and can be adjusted.

The Environment should not be confused, let alone limited to a “Context Diagram”.

The **understanding of** the environment and the **relation and all interactions** between the environment and the present or future system is crucial.

Design a system that is valuable and that can survive in an environment is impossible without understanding that environment.

The **Minimum Viable System** (MVS) is the version of the system with the minimal set of features, qualities and capabilities which is still able to create value by performing a business activity or, for non-business systems, to fulfil a minimal meaningful role.

The MVS is equipped with ALL the **critical capabilities** (**critical features** and **essential qualities and capacity**) and is suitable for a basic usage.

An MVS can not work with partial logic, partial processes or incomplete set of features. Removing one single feature, capability or reduce a characteristic will render the system unable to function, unusable, unable to produce any business value and unable to play its role. It would make the system worthless. The MVS is thus the minimum that has to be delivered. All critical logic and features **MUST** be implemented and delivered. Anything less can not be operational.

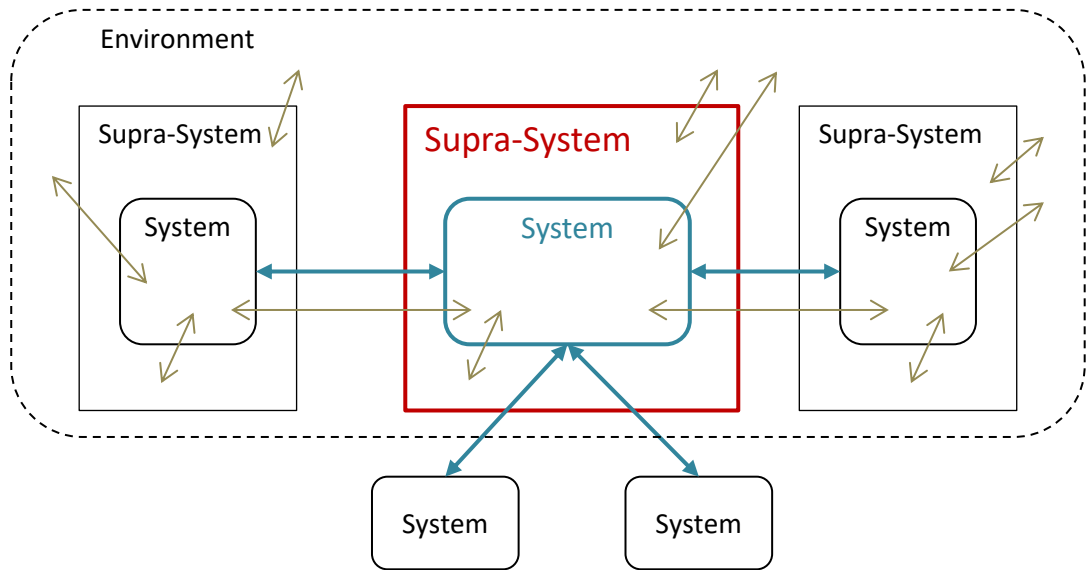
Examples:

- A car with an engine but without steering wheel.
- An airplane with take-off and flying capabilities, but which isn't equipped to land.
- A software system capable of accepting client data but which can't accept the products data or orders.

The MVS leads to projects with (relatively) fixed-scope !!

Possible Criteria for Determining an MVS

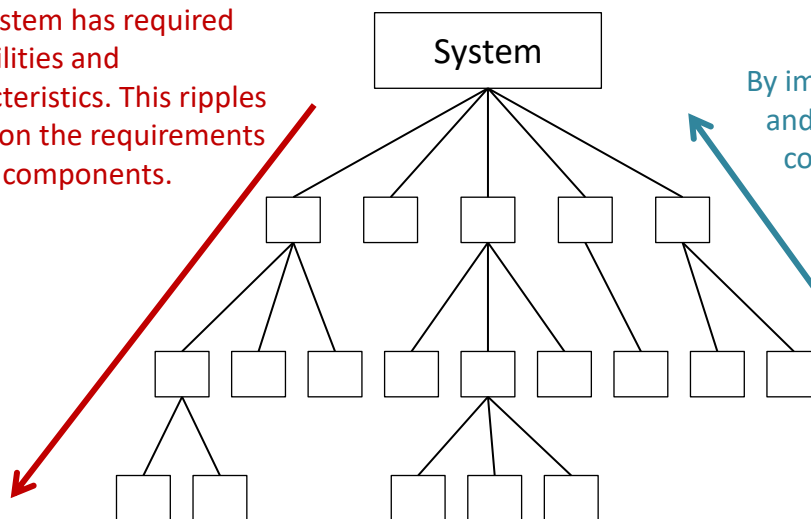
- Ability to perform a business function, to perform its role.
- Ability to perform an end-to-end process
- Ability to provide a service or to produce and deliver a product
- Ability to serve the supra-system or external customer



Systems and companies are/have:

- Connected
- Permeable boundaries
- Influences, interactions and exchanges among systems and between systems and environments and between companies and systems
- Consider that each system has its own environment (sub-environment of the company's environment or supra-environment). The environment not always / rarely a clear border. It is up to the practitioner to define its limits.

The system has required capabilities and characteristics. This ripples down on the requirements of the components.

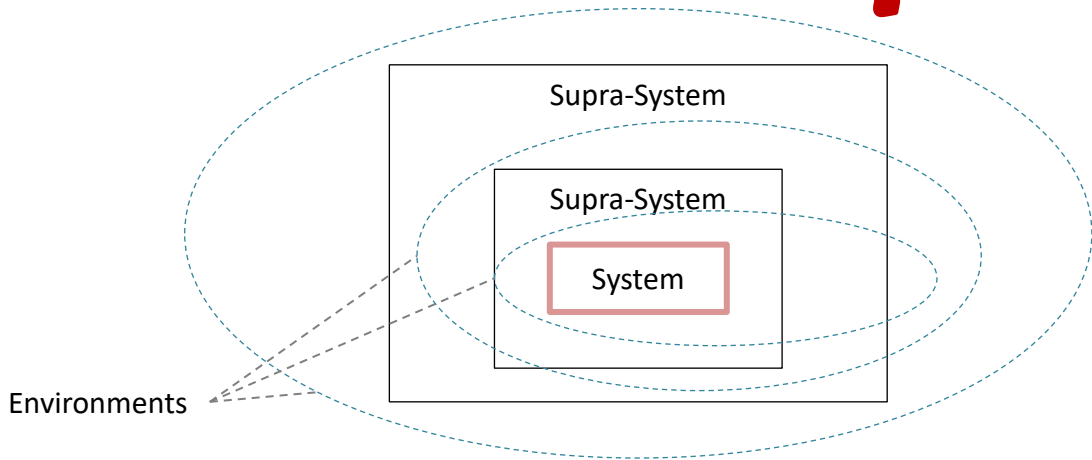


By implementing qualities and capabilities, the components create the behaviour, the capabilities and the capacity of the system.

Besides this model, connexions need to be included in this model.

5 KEY INSIGHTS

30/10/2018



1. System

- Purpose, structure, mechanisms, strengths, weaknesses, risks, prerequisites and limits. (This is valid for anything: process, methodology, system, ...)
- Every part of a (clean) system has a purpose, a function, reasons, ...

*Important to
understand this !!*

2. Supra-system

- Including the supra-supra-system, the supra-supra-supra-system, ...
- A system needs to be fully part of its supra-system (structural/systemic integration = integration by design).

3. Environment(s)

- must be determined properly (must contain everything influencing, even indirectly, in a meaningful way, the system), including those of the supra-systems
- the relations, interactions and influences between the environment(s) and the system(s).

4. Nature of the resources

- Resources and materials inputted, outputted and/or changed by the system

5. Internals weakening or that may weaken the system

- Internals: structures, mechanisms, dynamics and tendencies
- Among others, mechanism creating disequilibria endanger the system !!

For studying, analysing, engineering and managing systems
(like software systems, IS, IC, company, ...)

Analysis Approaches have to include this !!

CRITICAL PRINCIPLES in DESIGN of SYSTEMS (1)

Applicable to many types systems: political systems, governmental systems, economical systems, banking systems, organisational systems, rewards systems, system of (business) services, business models, information systems, ...

1. Systems amplify the human capabilities. Some systems are stronger than individuals !!!

- A system amplifies human strengths and qualities.
- Systems also amplify the effect of human flaws (weaknesses, negative & destructive qualities) !! (*beware when designing*)
- A system can help avoiding people to make mistakes (information, control & preventive barriers)
- A system can be misused and abused.

2. System's design must reflect the reality !!!

- System's design must respect nature, natural principles, first principles, ...
- through its structure, principles, relations, abstract concepts, mechanisms, ...

3. A system must fit in its environment !!!

- The environment interacts with the system.
- The environment as an influence on the design.
- The survival of a system depends of its environment. An environment can 'decide' and/or reject or terminate a system.
- The environment can't be ignored
- If a system doesn't respect the environment then it will be an enemy of that environment. Normally, the environment is stronger than the system.

CRITICAL PRINCIPLES in DESIGN of SYSTEMS (2)

4. A system has to implement ALL the functions necessary to be viable.
 - These are more than the functions dedicated to the purpose or to end-users. Examples: control functions, protective functions, management functions, ...
5. The system's lifecycle has to allow the system to deal with events happening during its lifecycle.
 - It deals with it or facilitates dealing with it, for example, by minimising the required adaptation.
6. A system has to respect key qualities and key capabilities.
7. The system has to respect spiritual values, universal values, moral values.
8. A system has to implement stabilising mechanisms, protective measures, safety measures, ...

Key Design Principles for Man-Made Systems^{18/12/2018}

Control and Protection

A system without control and protection mechanisms is at risk. Input, behaviour, performances, resource usage and outputs must be controlled.

Balance

Examples of balances are capabilities able to process approximately same amounts, win-win-situation, equity, and so on.

Modularity - Component-based

Modularity is achieved by having components that are loosely coupled (separation of concern) and high cohesion. Components can then easily be adapted and replaced. They create order. Components are also easier to support reuse.

Reuse

Developing for reuse can save a lot of time and effort. However, reuse requires a broader understanding and good anticipation skills. Components containing more general logic are usually potential candidates for reuse.

Internal Order

Internal order is, at glance of no concern of the business stakeholders. However, it reduces the time required to search for the concerned source code. It facilitates debugging, maintenance and adaptations. It decreases the learning curve. Order reduces complexity and complexity related risks.

Natural laws, Science, Rational Logic

Some laws, rules and principles may simply never been broken or the system won't work or it will lead to its decline. It is important to be aware and to understand these laws, rules and principles and where and how they apply.

Foundation

A system has to be based on a solid, well-organised foundation. The foundation can be organised in layers.

Knowledge – Insight – Truth

In large and complex systems or environments, it might be a challenge to acquire the required knowledge and insight. Often the truth may not be what we expect, want to hear or are ready to hear. But designing a system or solution with insufficient knowledge or a distorted mental image leads to problematic results

7 Main Survival Rules

30/10/2018

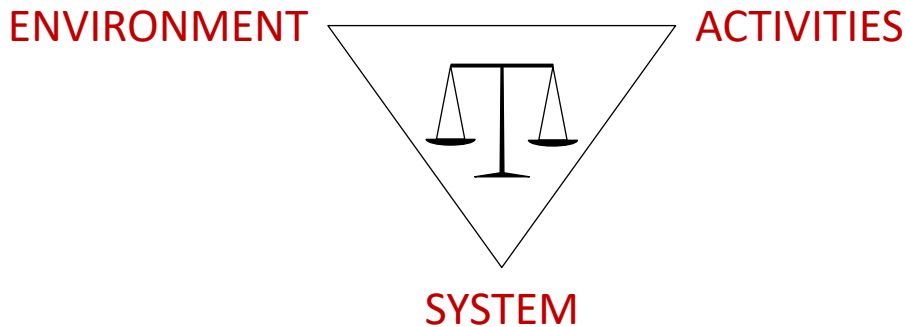
A **MAN-MADE SYSTEM** has to

- 1) have a **worthwhile purpose**, to function accordingly and to use available resources for it
- 2) be **useful**, produce (real) **value**
- 3) **fit** in its **environment** and be accepted (acceptability: like moral, respectful, ...)
- 4) fit into its **supra-system** and serve it by contributing to its purpose
- 5) have **internal harmony**: maintaining balance, reflecting real world, respecting natural laws, absence of conflicting logic, no mechanisms undermining the system itself, sound internal structure, ...
- 6) be able to **cope with events** it will or is likely to meet during its lifetime
- 7) **be better** than competing systems or alternatives

during its whole lifetime (thus being able to evolve).

And ...

- Each of these rules has many implications for the internals (the design).
- A broken rule can shorten the system's lifespan.



BALANCE AMONG THESE THREE ELEMENTS IS
OF VITAL IMPORTANCE

System

A system that is neglected will deteriorate. As a result it won't be able to perform activities.

Activities

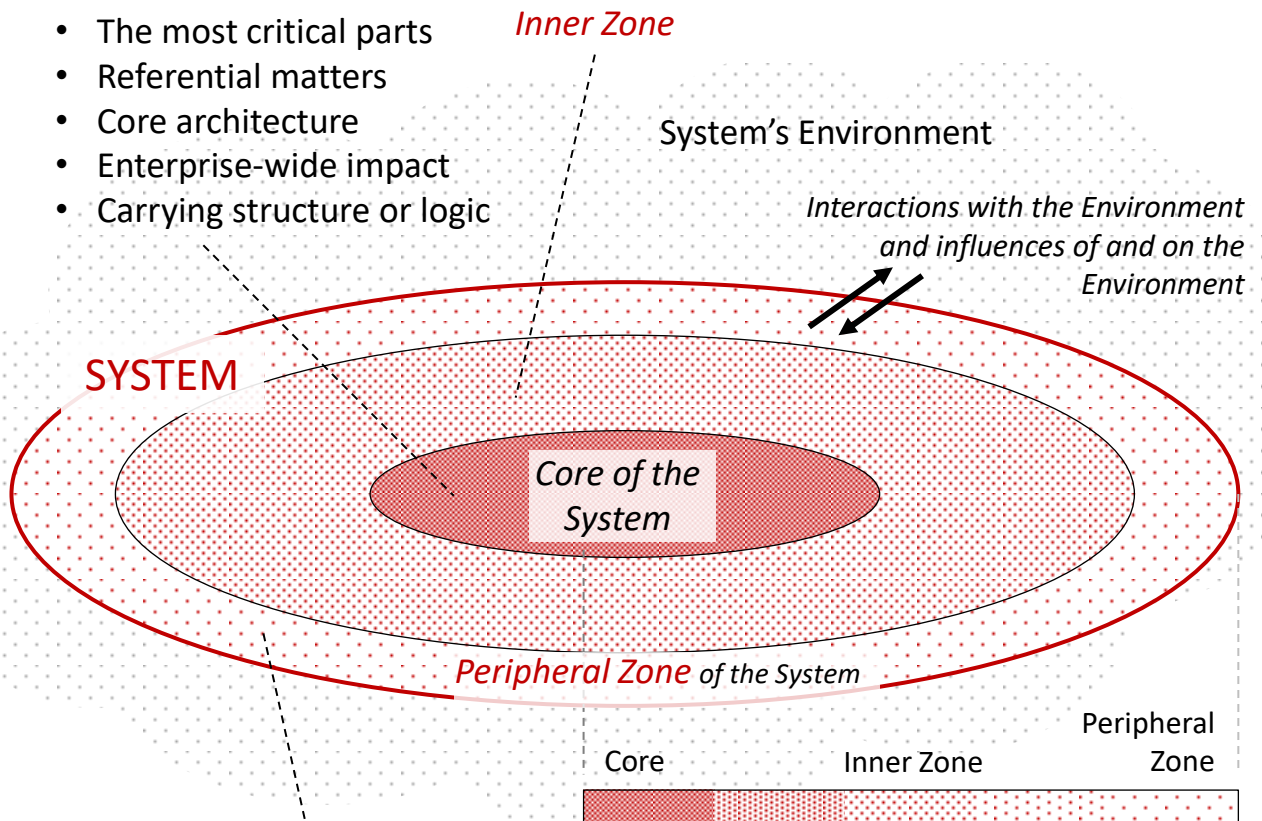
If the system doesn't fit with its intended usage or if it is not used (no activities are performed), then the system is useless. Some systems may 'rust' or die.

Environment

Each system exists in an environment. This environment is as important as the system itself. If the system is not aligned with the environment, if the system does not respect the rules of the environment, if the environment is not respected, then either the environment may kill, boycott or expel the system (like by stopping feeding the system) or the environment itself may deteriorate leading to the death of the system as well.

Core / Fundament of the System

- Core Logic
- The essentials
- The most critical parts
- Referential matters
- Core architecture
- Enterprise-wide impact
- Carrying structure or logic



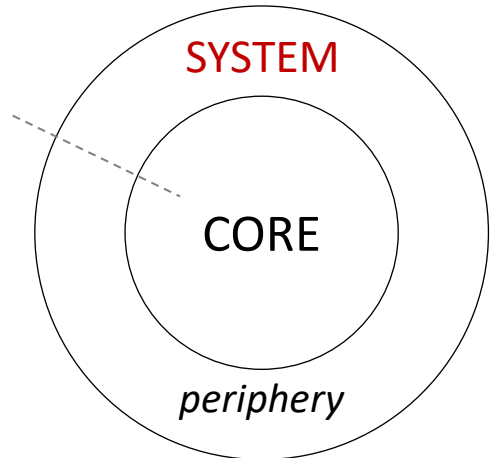
- Non-critical functions
- Functions built "on top"
- Functions with local purpose and importance
- Features with temporary value
- Extensions
- Whistles and bells

Notes:

The number and names of 'zones' don't matter. The concept does.

There are no real clear criteria or discrete boundaries between the zones. It's all a gradual scale and it's up to the judgment of people how to deal with it.

Everything that shapes systems, affecting the Structural Design and Core Mechanisms, usually strongly related to the system's purpose



CORE CONTENT

- Core Processes
- Core Mechanisms
- Core logic of key processes
- Key persons
- Role, functions, capabilities of components
- Architectural Structure
- Architectural qualities (integration, fragmentation, separation, duplication, scalability, interoperability, ...)
- Organisation of key resources (energy, matter, information, money)
- Components and Mechanisms required for survival and thriving

Adapting the Core requires huge changes, possibly structural changes. It is a lot of rethinking and reengineering. Often, it influences other areas across the system. The core should be well-thought and as stable as possible. (*How to increase the stability of the core?*). A system can be organised with inner-core, outer-core, foundation, ... These are just additional layers of the onion.

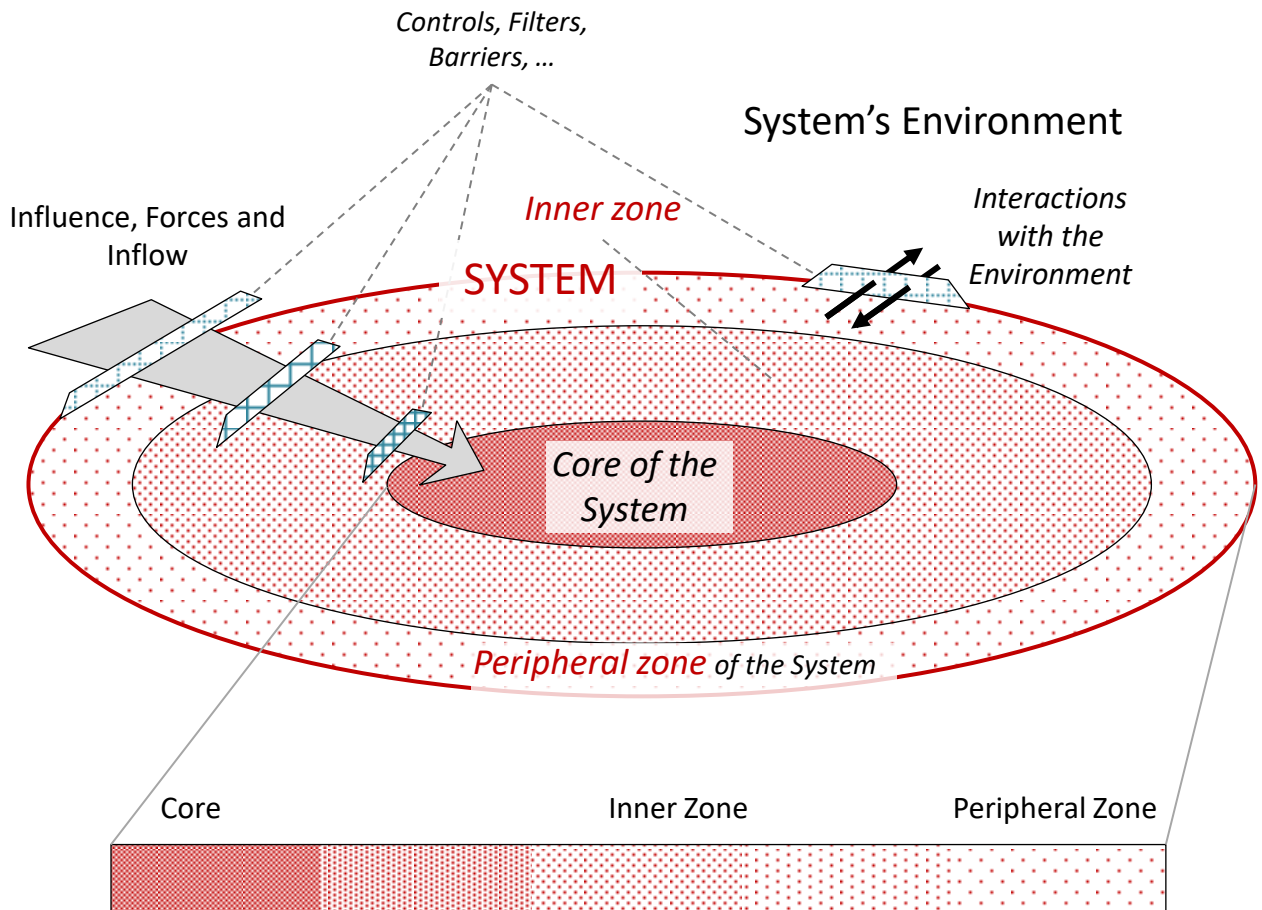
Periphery: (In software systems): features, user-interface, local logic, loosely coupled information, ... Changes in the periphery may affect other aspects of the periphery, but won't affect the core and are often simpler, local and easier to implement. More layers are possible (like an onion).

Not an end-user or stakeholder's issue: Something might be important or even vital for end-users and can be located in the periphery !! Or inversely, something can be of no importance to the end-users but be part of the core layer.

System's Controls (Tests)

10/01/2020

Inflows constitute a potential menace for the system.
Hence, they have to be verified.



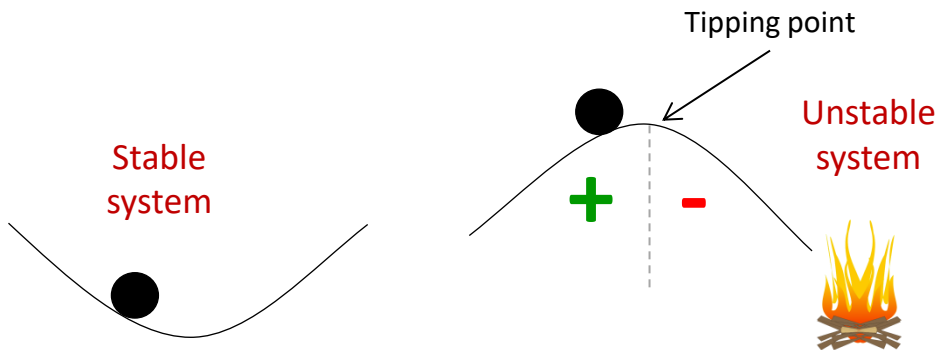
- More protected
- Verifications aimed to verify good functioning

- Protects the system from its environment
- Deals with the interaction between system and environment
- Verifications aim to protect
- Contains a lot of tests of input and output, but particularly of inputs
- Contain sensors monitoring the environment

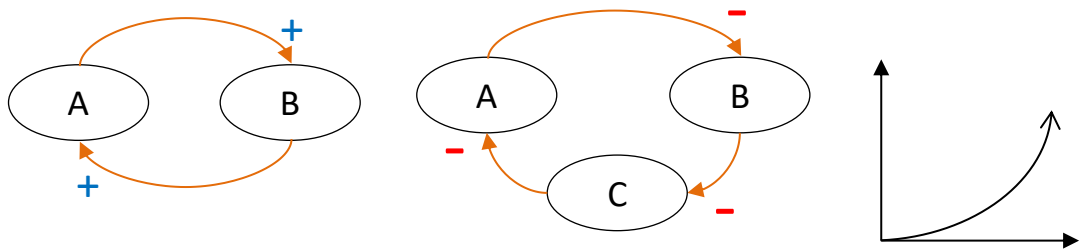
Notes:

Verifications, security, quality, as well as other characteristics may be dependent of the role and place of the part in the system. The organisation of controls and other systemic functions has to be defined per system.

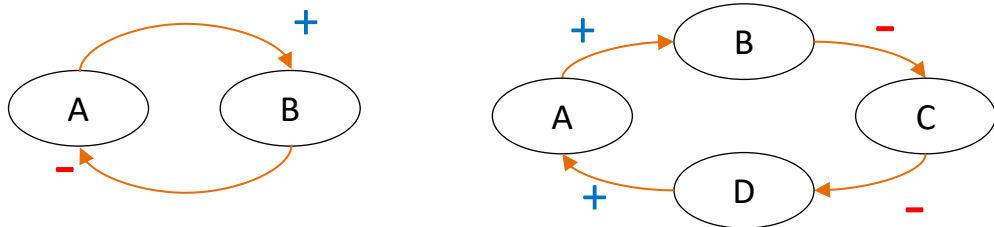
Stable and Unstable Systems - Tipping Point



Positive Feedback Loop – Reinforcing Loop

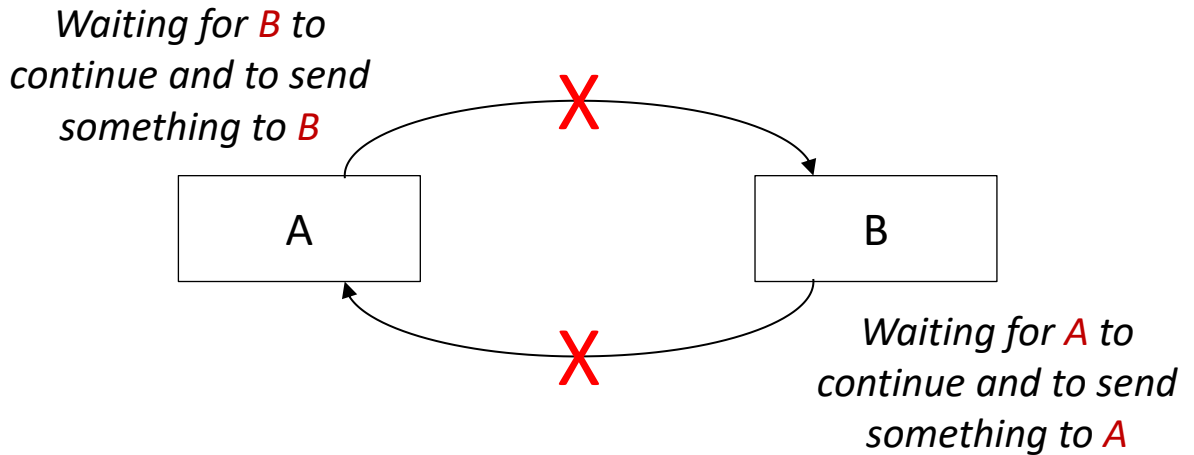


Negative Feedback Loop - Balancing Feedback



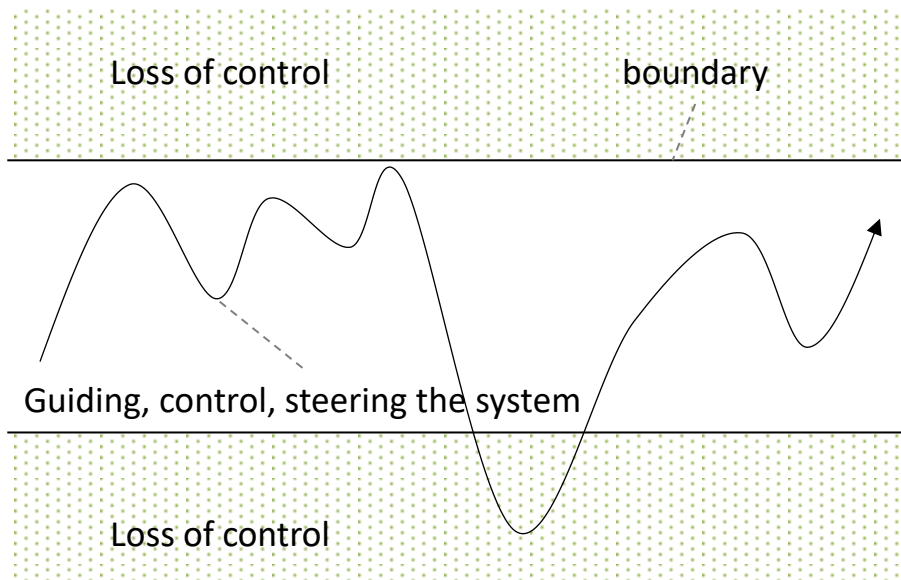
Positive and negative effects can be mixed in same dynamic model. The ensemble of these relations, their strength and the reaction of the components provides the final result.

Dead Lock



Critical Mass

Critical mass is the minimal amount of something required to bring about an unstoppable dynamic, one that is hard to stop or one that creates an acceleration



- Individuals are allowed to control a system within some boundaries. As long as the steering individual stays within these boundaries, he remains in control. If the individual crosses a boundary, (s)he some or all loses control.
- Control is a continuous dimension.
- A system has many boundaries (limits, thresholds, breaking points, ...).
- A boundary is not necessarily a precise line. It can be vague and broad.
- Crossing several boundaries may enlarge the effect. Crossing a boundary has not necessarily a linear effect on the system.
- Crossing the boundaries does not necessarily mean a loss of all control.
- These boundaries are not “fixed”. They are determined by the environment and by the system (thus defined by the designers and builders).
- Systems have to be respected. It is unwise for an individual to do whatever it wants with a system, unless it seeks to create havoc.

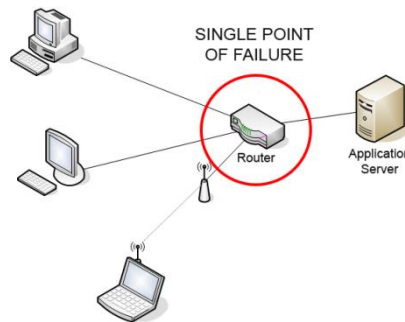
Notes:

Examples: human body, companies, information systems, climate system, financial systems, ..

Weakest Link



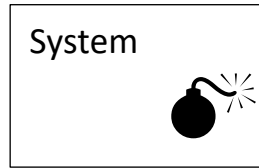
Single Point of Failure



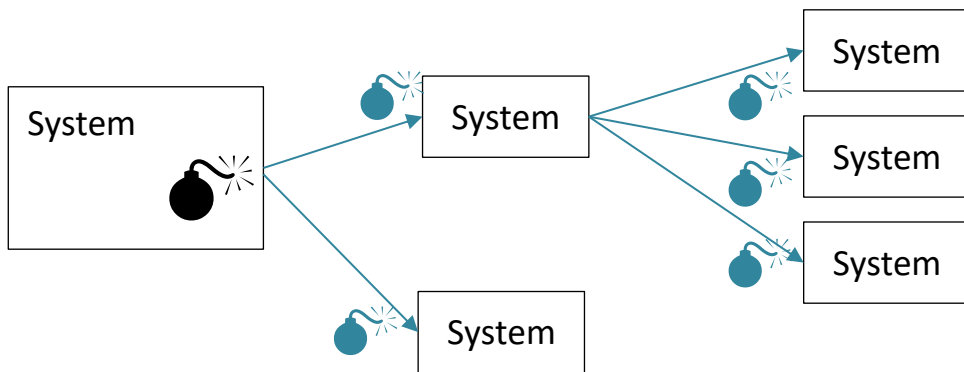
Definition: A single point of failure (SPOF) is a part of a system that, if it fails, will stop the entire system from working. (Wikipedia)

Applicable to any component, including users, interfaces, connections, services, external devices or tools, ... Components, tools and systems in other systems and organisations.

Single System Failure



Multiple Connected Systems Failure (or SoS Failure)



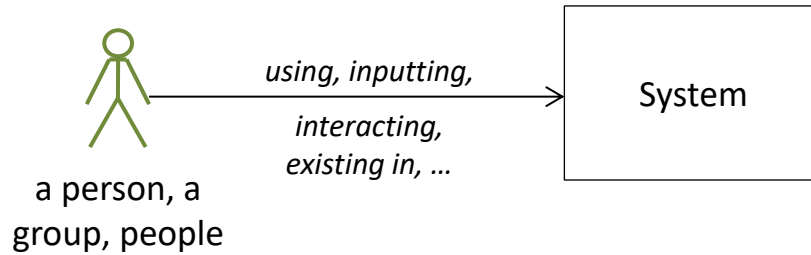
Domino-effect

- A company is a set of inter-connected systems.
- These systems might be connected to external systems in other organisations.

How to conceive systems resistant to the domino effect?

Note:

Domino-effect is commonly considered as repercussions on other systems leading to their collapse. This is an extreme form. One problematic issue on a system may influence the functioning of other systems. Or, they may contaminate them. This influence is not necessarily direct, immediate and obvious. Sometimes, it may even be immaterial, like a loss of reputation and trust.



A system

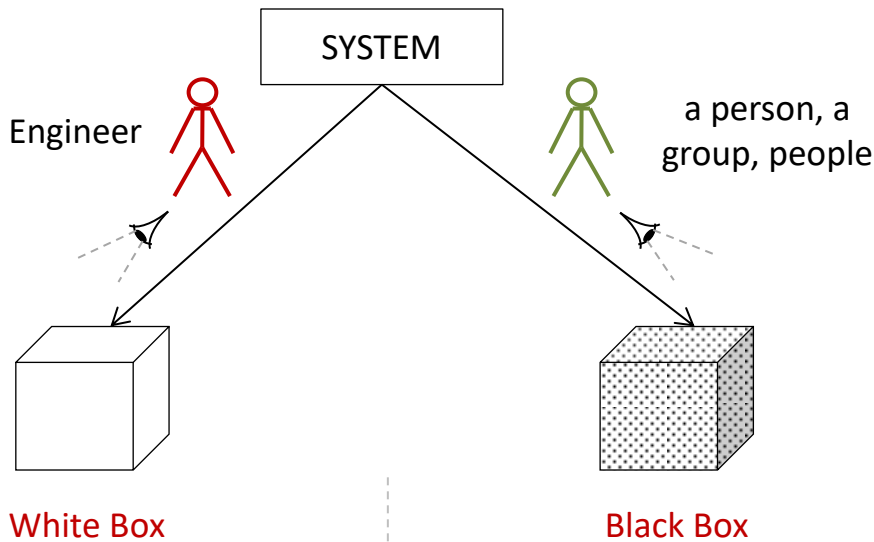
- responds to this input, usage, force, influence, ..
- accordingly to its own
 - Nature
 - Characteristics
 - Rules, Laws, Principles
 - Mechanism, dynamics
 - Capabilities
 - Limits
 - Constraints
 - ...

Notes:

- A system receives a trigger, a signal, a kind of input, forces an influence or its operational environment changes and it reacts to it accordingly to its own nature and to its inherent logic regardless of what people want.

Example: A car behaves well and follows the orders of the driver as long as crucial parameters remain within a certain range. If one or more exceed a limit, the car will act in a way not meant and not desired by the driver.

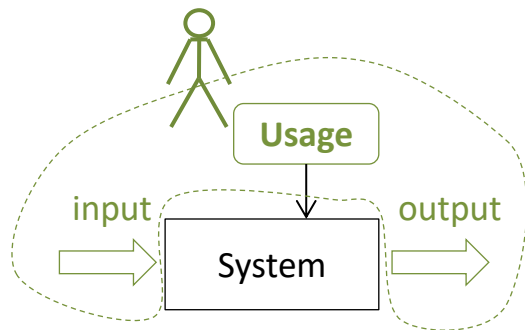
- The person (or the people) has to create the conditions, provide the right input and activate the right behaviour of the system. This will produce the desired output or result.
- Using or adapting a system that is only partially understood is very risky. One never know what new problems it introduces and how the system will react.



- Understands and has access to the system's internals

Limited understanding

- Understands the system through its inputs and outputs
- Understands the usage
- Some internal functioning can be understood or this insight can be assumed



Real insight in the system's internals and having access to them, makes it possible to manage the system. It greatly reduces the risks.

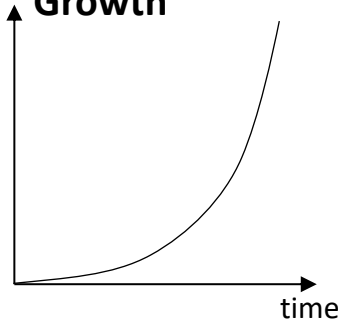
Managing the usage, input, output, but also the environment can be achieved with black box insight. Managing may be very risky depending of the system.

We can only manage what we understand.

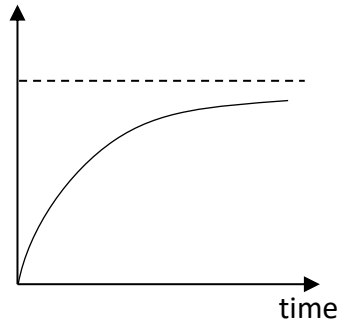
Exploitation and Growth of Systems

30/10/2018

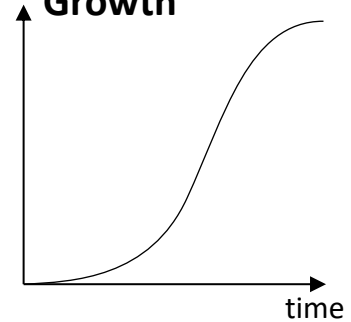
Exponential Growth



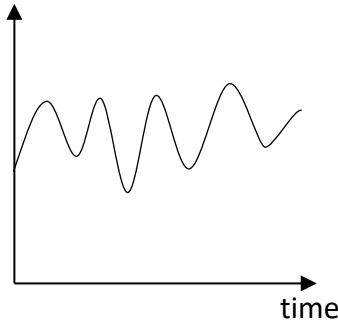
Goal Seeking



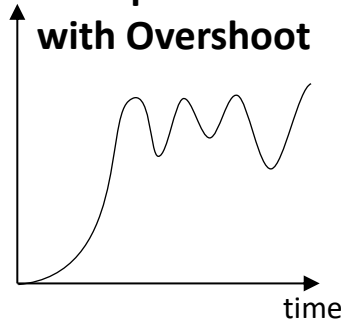
S-Shaped Limited Growth



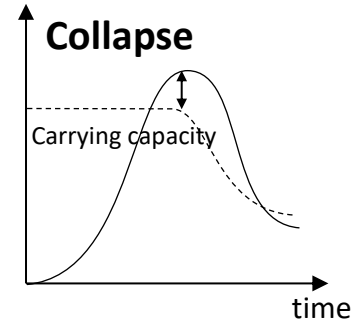
Oscillation



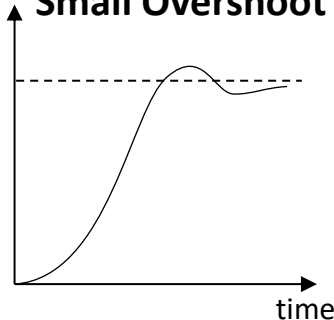
S-shaped Growth with Overshoot



Overshoot and Collapse

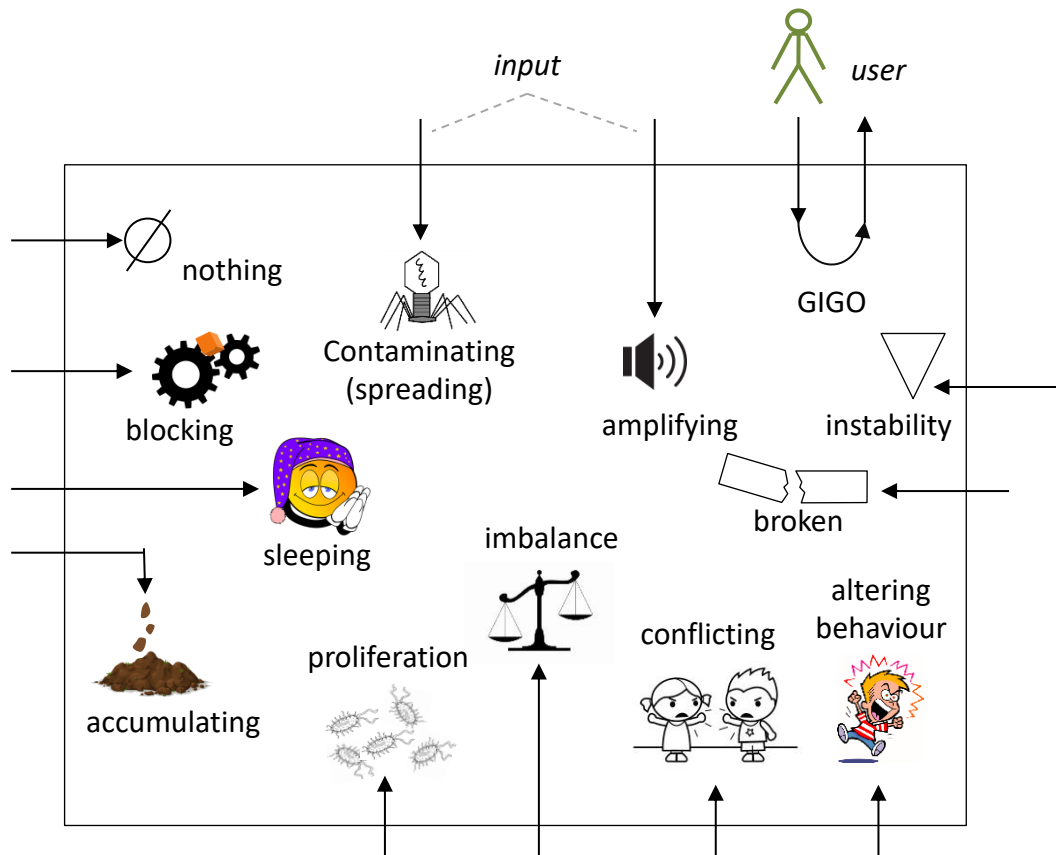


S-growth and Small Overshoot



Sustainability: Remain under the carrying capacity of the system.

Practical: Interesting to take decisions about capacity and scalability or for the implementation of mechanisms like buffers, triggers, alarms and controls.



Other effects:

increased pressure, wear and tear, frustration, demotivation, learning the wrong lessons, irresponsibility, lower values and norms, various damages, wrong drivers (motivators, reasons for doing something), uncertainty, confusion, increased sensitivity, negative forces (influences, ...), aging, degradation of matter, SPOF, domino, destroying, slowing down, speeding up, erratic behaviour, unreliable, wear and tear, ...

Note:

- Some issues occur in software systems, other in physical systems and other in human systems.
- Not all of the issues (applicable to software systems) can be detected through the user interface.
- Idea: Make a list of all possible types of problems that may deteriorate a system. Then indicate on what type of systems/matter it may occur (natural system, human system, industrial systems, business system, software system, ...)
- Idea: Identify who can identify them and how to identify them (required competencies).

TENDENCY TO ENTROPY

OVER TIME

SYSTEMS

TEND TO **DETERIORATE GRADUALLY**,
HEADING TOWARDS THEIR **COLLAPSE**.

MAJOR AREAS

- System's Functions
- Internal Organisation
- Usage / Input
- Resources

ISSUES

- Awkward to use
- Improvisation
- Patch and Pray
- Misuse, abuse
- Too many
- Illogically organised
- Vague
- Unreliable
- Missing
- Duplication
- Different forms and shapes
- Incoherent, inconsistent
- Conflicting
- Confusing, unclear
- Overly complex
- Ballast
- Unsuitable
- Not standardised
- Fragmentation
- Chaotic

TENDENCY TO ENTROPY

OVER TIME **SYSTEMS**

TEND TO **DETERIORATE GRADUALLY**,
HEADING TOWARDS A **COLLAPSE**.

How to prevent this?

- Guiding the Usage
- Clarify Norms
- Barriers against Abuses
- Limiting the Freedom
- Appropriate Control Mechanisms
- Appropriate Monitoring Mechanisms
- Never lose control or give control away
- Foresee corrective mechanisms

Notes:

The more something is critical

- the more crucial it is to understand it fully
- to put the best people on it with the best competencies
- To acquire and strengthen the required competencies

Controlling and Managing a system requires advanced skills.

Wrong Behaviour

Causes

- Structurally
- Logically
- Inappropriate materials, unsuitable application

Blockages, limits

Causes

- wrong interfaces
- inappropriate materials, materials don't fit
- inappropriate logic
- sometimes inappropriate structure

Rotting System

When a system rots, it often rots

- from the inside, under the surface
- gradually
- imperceptibly, unnoticed (surprise!)

Causes are inappropriate or lacking internal mechanisms.

Frequent causes: Lack of controls, incoherent logic, wrong incentives, wrong control, wrong objectives

Notes:

The main idea is to give **important clues to consider when designing systems** (not just IT systems or software systems, but also all kinds of organisational systems).

Therefore, the listed factors here may not be exhaustive.

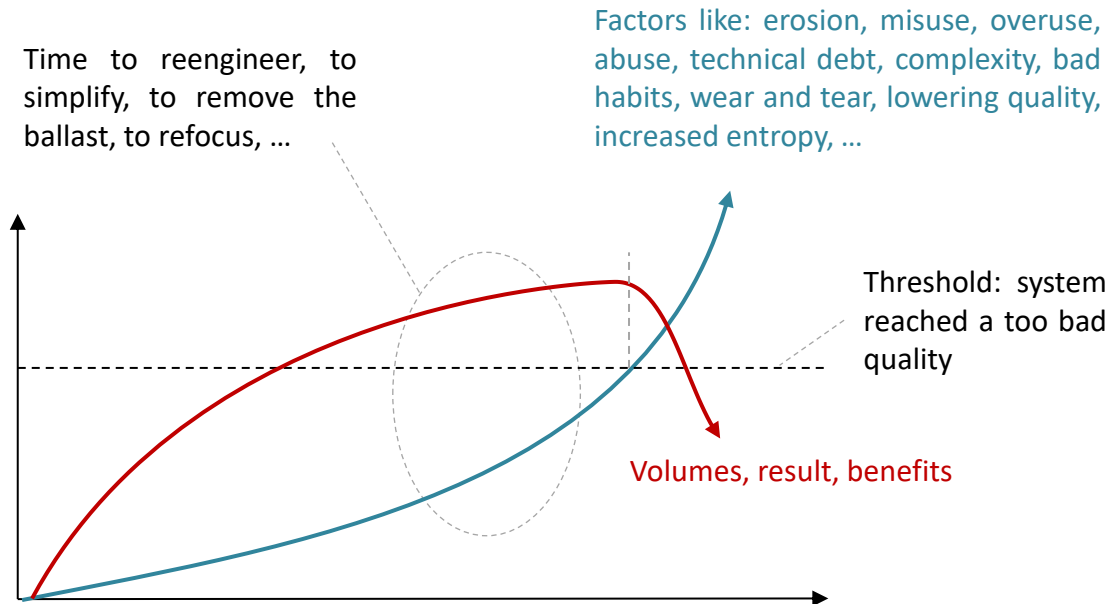
Main Cases of Dysfunctioning Systems

- Not the right **objective**
- Not inline with the **purpose**
- Not taking the **environment** into account
(insufficiently, underestimations, ...)
- Not **driven** by the **right** forces (wrong incentives)
- Lack of **controls** and other **protections**
- Lack of **manageability**
- Overly adding **complexity** or **oversimplifications**
- Lack of **integration**
(purpose, role, functional, logical, standardisation, ...)
- Lack of **training**
- **Abuses** of systems

Can be prevented
by DESIGN

System's Gradual Decay

10/01/2020



Gradually the system will decline. Erosion and other similar tendencies may undermine the system. Probably, this will first affects its performance and its results by a decline.

At a certain point, the cost/benefit (effort/result) ratio may become uninteresting. But an undermined system may also block or collapse. The decline may be sudden or gradually and predictable or unpredictable.

Beware: System's decay often happen internally, inside the system, under the surface. It can evolve for a long time under the radar. It takes some time before signs are perceptible and before the signs create awareness. Measures and analysis may bring these tendencies earlier to the light.

Note:

The lines do not represent the course of real evolution. They are only drawn for the sake of clarity, to explain the concept.

Self-Destructive Designs of Systems

14/02/2020

A design of a system may embed its own destruction

Self-sabotaging system

A System that is in the way of its own intent or own goals.

Unsustainable systems

A system may behave, by design, in a way that it is unsustainable.

Self-destructive

The design may implement mechanisms destroying the system itself.

Systems Failures in Time

14/02/2020

Short Term

- Inappropriate
- Missing capabilities or functions
- Incoherent Logic
- Defective
- Missing parts
- Integration issues
- Access issues
- Interaction issues with supra-system or environment (inputs and outputs)

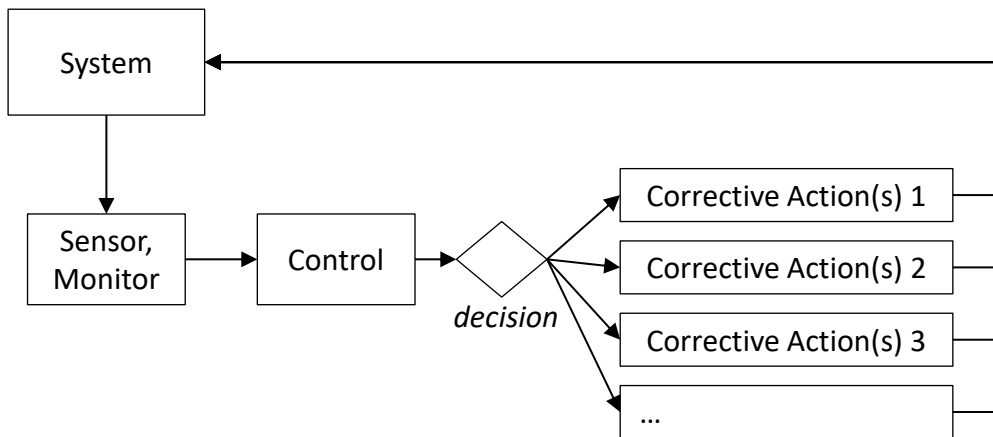
Usually
tested

*“Does it work
and
is it usable ?”*

**CAN BE ROOTED IN
BAD CONCEPTION !!**

Long Term - Survivability

- Gradual internal decline
example: quality of information declines
- Possibilities for misuses and abuses
- Flexibility & Evolvability
Adaptability, connectability, interoperability, extendibility, scalability
- Unsustainable input and output
- Evolution of behaviour
 - Internal behaviour
 - Creating internal imbalances and disharmony
 - Encouraging undesired behaviours
 - External behaviour
Behaviour unacceptable and unaccepted by the system or by the environment



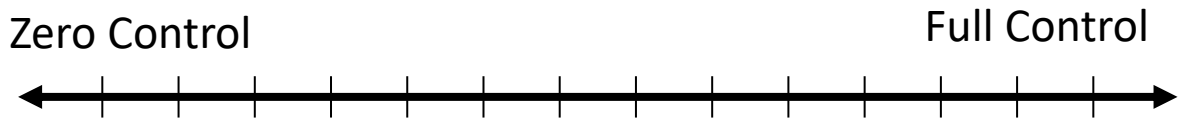
Controls are vital mechanisms for any system. Their aim is to ensure the right behaviour of a system, improving its efficiency, indicating when to adjust or adapt.

Controls are a matter of survival of the system

A system without controls or lacking of appropriate controls, and lacking of appropriate corrective mechanisms coupled to these controls, is prone to deterioration, abuse, fraud or any other form of decline.

Note:

- A team, an organisation, a company and even a society are also systems.
- Corrective actions can be, among others, adapting the system's behaviour, adapting the system, adding limitations, sending an alarm signal, changing a status, replacing parts, etc. A corrective action can be taken ad hoc.
- It is possible to control the controls. Questioning the efficiency of controls often happen only after a disaster struck. The question is then raised why the controls didn't function or how to improve the controls.



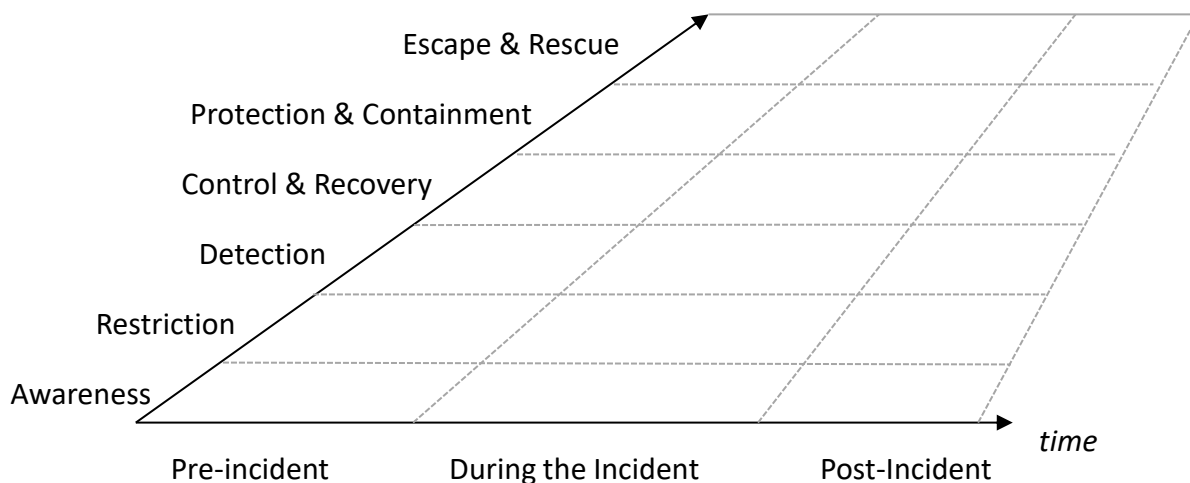
1. Ability to monitor the system
2. Understanding the interface
3. Possibility to adjust the behaviour and output of the system
4. Possession of the necessary competencies to adjust the behaviour of the system
5. Understanding the internals of the system
6. Possibility to adapt the internals of the system and access to these internals.
7. Possession of the necessary competencies to adapt the system

These aspects are **not binary questions** (yes/no). Their answer has to be expressed as a degree.

Barrier Functions in a System

10/01/2020

Prevention	Awareness	Deeper understanding of the matter; Insight and understanding of how system works, of risks and hazards.
Prevention	Restriction	Limitations of actions or imposing to establish necessary pre-conditions before an action or processing.
Prevention /Resolution	Detection	Verification/detection and warning mechanisms
Mitigation	Protection & Containment	Limiting further damage or losses being inflicted; Avoiding spread of damage, such as contamination
Resolution	Control & Recovery	Managing the incident, taking measures Recovering with minimal harm or loss



Control mechanisms, recovery systems, back-ups, emergency procedures, and so on have to be elaborated before an incident happen to be used either during or after the incident.

During an incident new restrictions can be imposed, such as access limitations. Awareness of the incident and of what and why actions can be taken is different from the awareness necessary for normal operating conditions.

Some key notes about Systems for Engineers

1. Man-made systems are engineered or can be engineered (majorly, not necessarily for 100%).
2. Characteristics are engineered into man-made systems. They do not always materialise as intended and some appear spontaneously. Some characteristics are necessary or desired, while others are not. It is desirable that the creation of capabilities and characteristics happened in a methodical, intentional and controlled way. This belongs to the essence of engineering. Systems Engineers seek to determine what characteristics and capabilities are required and to engineer them into the system.
3. Systems have capabilities, capacity, prerequisites, minimal and ideal operating conditions, constraints, tipping points, breaking points, limits, weaknesses, bottle-necks, deadlocks, SPOF's,
4. A system reacts to input, to forces and to changes in the environment in which it operates.
5. A combination of specific factors or exceptional circumstances may trigger a complete change in the system's behaviour. The change of behaviour can be (partially) unexpected due to lack of insight in the system, the circumstances and its environment.
6. A system reacts and behaves always accordingly to its own logic and characteristics. It follows always its own program. Piloting (or using) a system is in fact providing an input to the system. This input triggers a reaction and a behaviour of the system which will create an output or outcome. The pilot (user) loses control when the system exhibits a behaviour that was not intended or when the output is unexpected. Unsuitable circumstances in the environment or inside in the system or an incorrect input won't lead to the intended result.
7. Misuses of systems is more important when the user doesn't know how to use the system or when the freedom of the user is not under control.
8. When a system's purpose is changed by design, or when a system is used for another purpose than the one intended by its design; the system is usually either lesser efficient and effective. This usage may even weaken or deteriorate the system.
9. A system may implement principles, concepts and mechanisms which may undermine itself from within. It may exhibit self-destructive behaviours and tendencies.
10. Flexibility and freedom increase the complexity and risks of misuse and abuse. Freedom should go together with responsibility, competencies and wisdom.
11. A system is equipped with internal controls to protect itself from misuse, abuses, other external influences, inappropriate circumstances and input.
12. Having as goal meeting the criteria of "a working system" leads to mediocre systems. This represents only a single perspective. Meeting more and higher norms is imperative.

Some key notes about Systems for Engineers

13. Some consequences of a problem are visible. Some are undesirable effects that are experienced. The shortest and cheapest way to eliminate these annoyances is to mask or solve them, or sometimes by dealing with their direct causes. This is why consequences, symptoms, are often solved and not the real problem. We label them too easily as cause (cause of annoyance). We neglect to look to their cause, and then to consider this cause as a consequence and to look further upstream until we find the root cause.
14. Solving consequences allows the underlying problems to continue to exist.
15. Solving symptoms makes systems more complex. We don't solve problems, we add solutions, layers, exceptions and by doing so we create chaos.
16. A system forms a whole. Changing something may impact the whole system or may impact it on a different place, area or aspect as the one where the change occurred. This is why it is advisable, and sometimes even imperative, to understand the whole system before changing it or seeking to change the system through influencing it.
17. In mechanical systems the interactions are much easier to understand. Therefore, the causes and consequences are easier to detect. In more abstract or complex systems; like organisations, software systems or natural systems; the logic and the interactions are much lesser obvious.
18. Understanding the environment and the interactions, tendencies, forces, opportunities, threats, events and other influences between the system and its environment is essential to engineer suitable systems.
19. It is wise to get first the foundation, the carrier structure and core part, right when engineering a system.
20. An approach based on systematically accumulating parts and changing logic is more prone to result in a overly complex, disorganised, confusing, incoherent, ineffective, inefficient system. A system is more than the sum of some elements. Structure, internal organisation, order, coherence, interactions, alignments of parts are key to systems.

For man-made systems,
“Good” is “NOT Good Enough”



- COMPANY –

2 PERSPECTIVES

COMPANY as SYSTEM

COMPANY as ENVIRONMENT

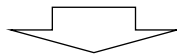
Looking Outwards

Looking Inwards

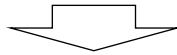


Why discussing the company in Information Systems Engineering ?

- The company forms the environment of the Information System. The IS has to fit into the company and must be able to operate in it.
- Information Systems are parts of the company. They have to be aligned with the company, its business model, its strategy, its culture, operating model,
- All information systems have to contribute to the company's objectives. The company is their supra-system.
- Information systems influence the company. They implement and influence its functions, capabilities and characteristics.
- Company and its environment influence aspects of the IS design and its functioning. They both play a critical role in the determination of objectives, role, purpose, functionalities, capabilities, characteristics, behaviour and constraints of the IS. They also determine or influence the logic and architecture.
- Information systems and information may drive the business and are a source of innovation.



The conception of the IS must respect the nature of the company.

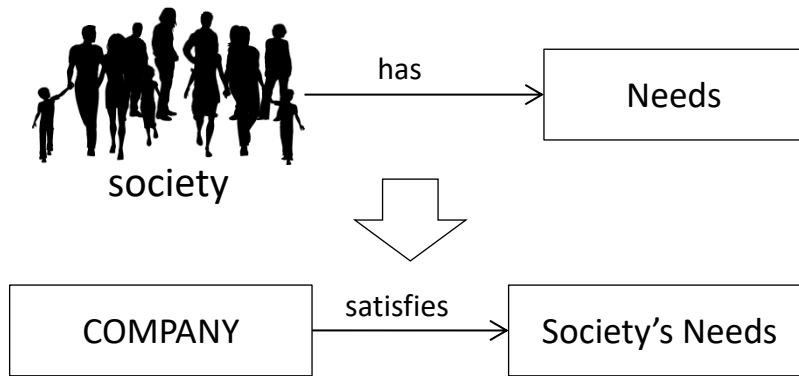


The IS Engineer must understand what companies are,
as well as the specific company and its environment,
and take them into account.

Notes:

- If the company requires to be flexible, can the IS be rigid?
- If the company requires to be manageable, can an IS be an unmanageable black box?
- If a company is a guided system, can ISE then be a directionless process?

What is a Company?



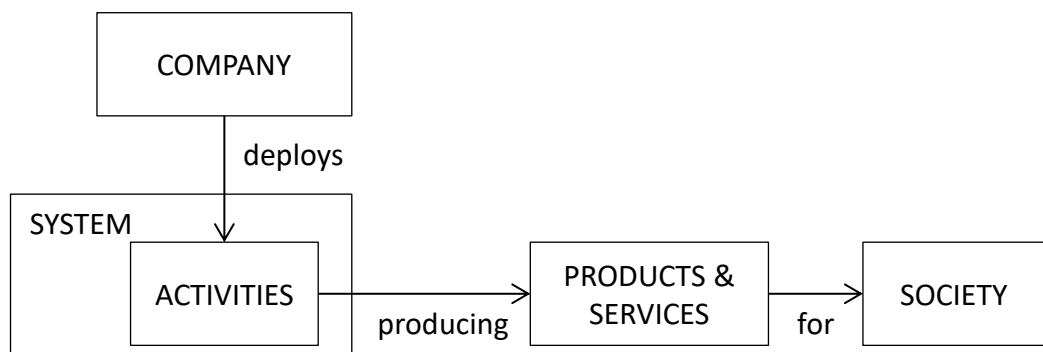
Company's Purpose

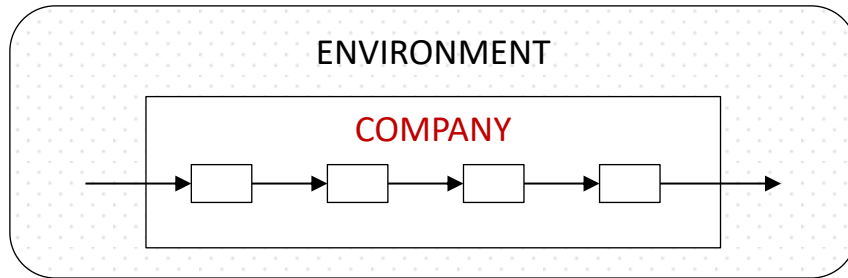
- Providing value to the society
 - Society: people, living beings, environment
 - How? Through its products and services
- Making society to progress

But, it is also

- a formalised entity consisting of people working together to a common goal
- a way for people to contribute to society and to make a living
- a way for personal growth

Achieving the Company's Purpose





A company is a system that produces products and/or services and sell them in order to prosper.

A company is a man-made open dynamic purposeful evolving socio-technical managed living system

- Man-made → understandable, manageable, ...
- Open → interaction with environment
- Dynamic → has a behaviour
- Purposeful → has a specific role, intend, objective, ...
- Evolving → changes, adapts
- Socio → containing people
- Technical → containing machines, computers, tools
- Managed → controlled, managed and guided
- Living → **complex system seeking to exist and to prosper**

These are goals, capabilities and characteristics that approaches, methodologies, methods, architecture and sub-systems have to take into account, to respect, to implement, or to contribute to.

What is a Company

30/10/2018

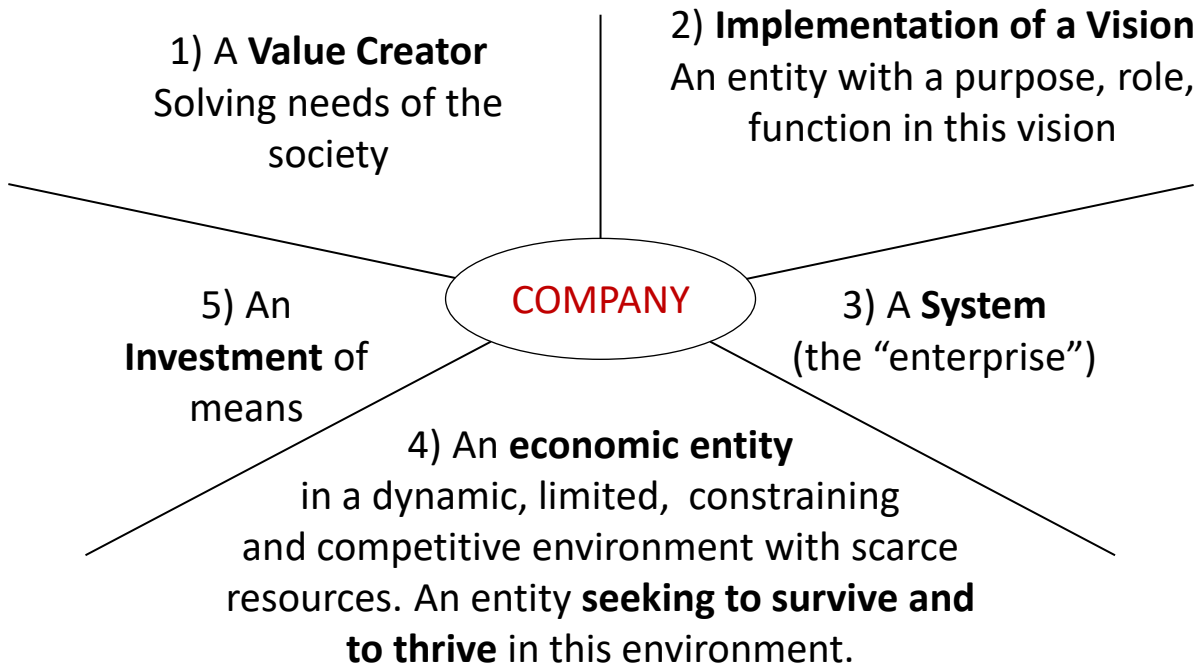
See also the few pages in section

“FOUNDATION – Core Ideas”

What is a Company ?

30/10/2018

5 Perspectives



Complex, man-made, dynamic, open,
heterogeneous, managed, guided,
economical, socio-technical system

- A perspective is only a limited view on a company.
- The company’s (sub-)systems have to comply and to be aligned with the whole ensemble of these perspectives.
- Each perspective brings its own understanding, priorities and requirements. None of these perspectives can be ignored.
- Each has its own far reaching implications in aspects like the development approach, priorities, design or even organisational and collaborative aspects.

The Analyst has (to understand and) to reconcile these perspectives.

Implementation of a Vision

Some companies are truly implemented to realise a vision. Others simply evolved without. They simply deployed activities, took decision and seized opportunities which appeared as the most beneficial at that moment. A vision provides guidance and focus. It helps to concentrate the activities on some type of work, on products and services, on needs and/or markets. The entire company has to build on a coherent whole and direct its effort to a common goal. This is mainly a top-down guidance. It avoids inefficiency due to development of unrelated activities in different directions.

Value Creator

A company that doesn't create value for the society is useless. It's a parasite. Value is created through activities. Competition forces the company to be effective and efficient. The company has to stay ahead by having better products and services. Customer relation and innovation are crucial. Innovation requires a propitious environment and advanced skills.

An Investment

An investment is the allocation of money in the expectation of creating or getting a benefit. Typically, the investor takes the decision to invest based on ideas, plans or studies demonstrating the likelihood of a creation a benefit. The investor seeks to know more about the initiative in advance. These plans have to be implemented and executed.

Economic Entity

An economic entity is about value creation through the production of goods and services in an effective and efficient way. It seeks to maximise its return on investment. Money is a critical mean for a company to survive, to function and to adapt.

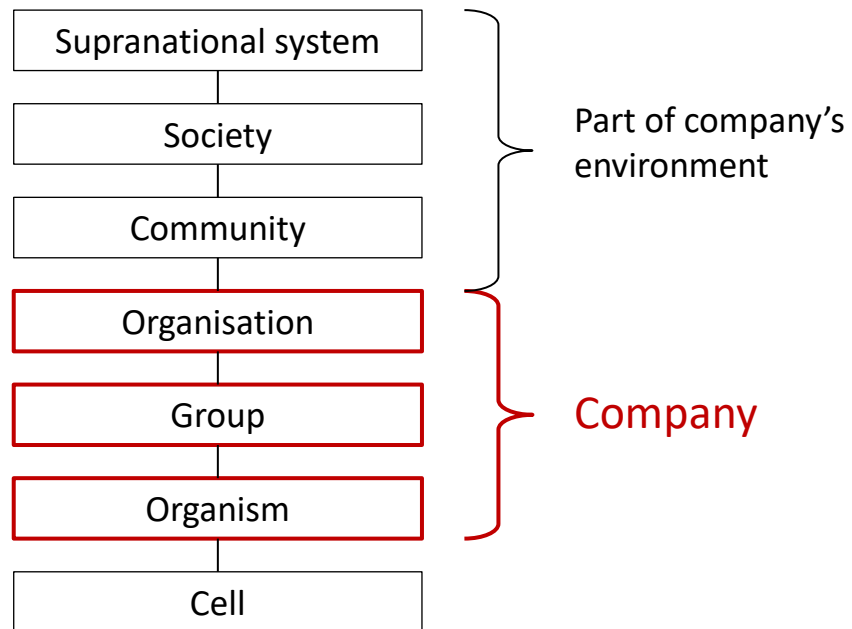
System

A company is a system. Some of its parts function in an undefined (free, ad hoc, erratic, variable, unpredictable) way. These parts don't suit the concept of 'system'. The system-like part of the company must be treated as a system. The company has to evolve, to renew itself. It has to engineer itself. Systems engineering (engineering of systems) is a key discipline.

A Company is a Living System

30/10/2018

Living systems are open self-organising living things that interact with their environment. (Wikipedia)



Living systems try to remain alive, to grow and to prosper

Living systems transport:

Matter

Energy

INFORMATION

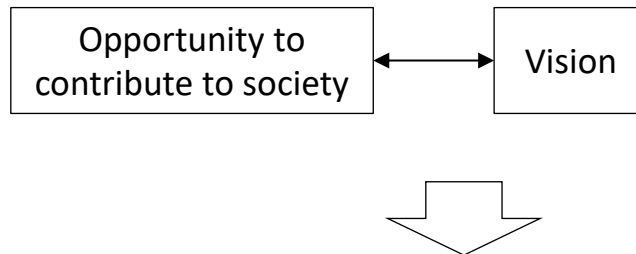
Organisations (including the architecture, subsystems, capabilities, ...) must be designed with these abilities in mind. What are the implications?

Living systems

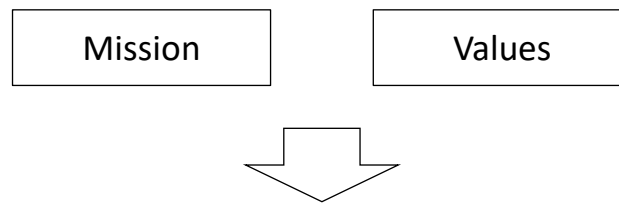
- maintain themselves
- organise themselves
- develop themselves
- adapt themselves

A matter of survival

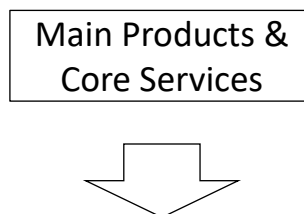
What does the society need? How does the society evolve? How do we envision a future society?



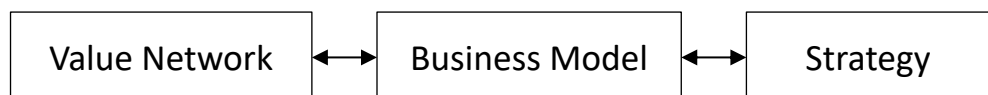
What role will our company play in the creation or evolution of this society and by respecting what values ?



What main products (or type of products) and (kind of) services will our company produce that will contribute or change the society in line with our vision, mission and values ?



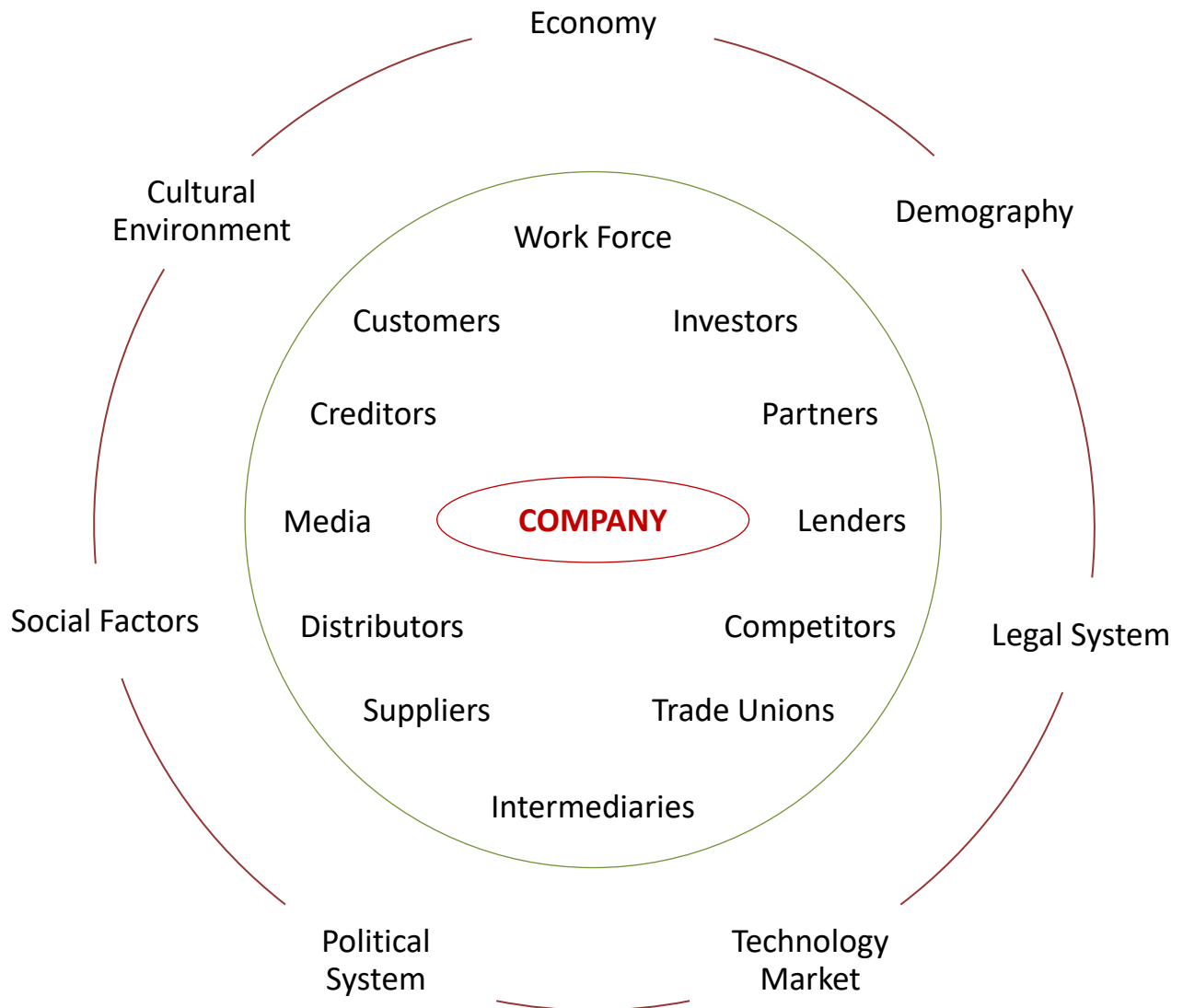
What high-level system has to be created in order to produce these products and services and this while being inline with the vision, mission and values?



Notes

Company's Environment

30/10/2018



A entity that can only exists if its environment allows it

Notes

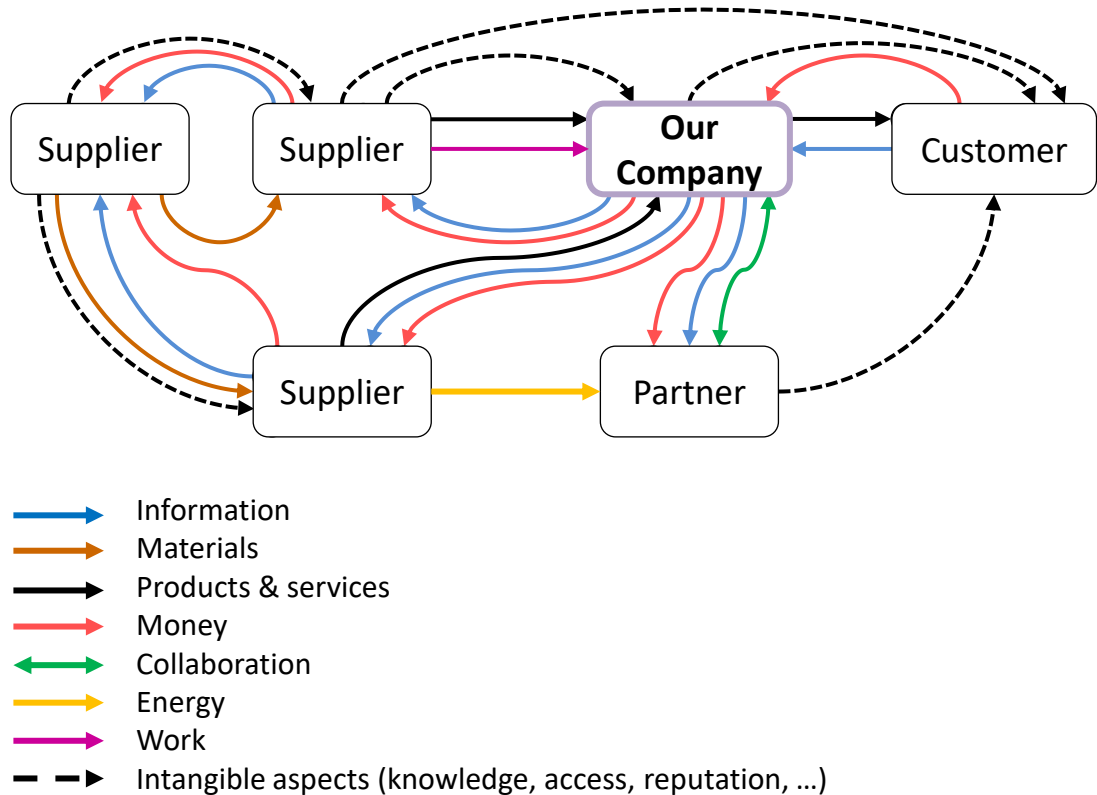
A company operates in an environment. This environment provides opportunities and threats. The company's internals are influenced by this broader environment. The company has to fit in its environment. → Understand the environment and take it into account when conceiving solutions.

See also

- Pesteli-Framework
- Michael Porter's Five Forces Model

- Society
- Community
- Country organisation
- Governments
- Political system
- Politics
- Law makers
- Legal bodies
- Economical system
- Healthcare system
- Financial system
- Recreation
- Ecology
- Technology
- Natural environment
- Climate
- Partners
- Suppliers
- ...
- Competitors
- Suppliers and partners of competitors
- Clients
- Prospects
- Work force
- Trade unions
- Academic Institutions
- Research centres
- Media
- Transport infrastructure
- Communication infrastructure
- Energy infrastructure – Energy supply
- Water infrastructure
- Service providers
- Raw materials
- Resources
- Waste disposal infrastructure
- Social Security

VALUE NETWORK



Notes:

- Represents
 - What party creates what value
 - Exchanges: Flows set up between different parties involved in the creation of value (final product) and for the deployment of business activities.
- Flows can be enriched with descriptions and other data (value, capacity, delays, ...)
- Suppliers, transporters and distributors are part of this system.
- Model can be enriched with plenty of information: amounts, problems, critical/optional flow indicator, positive/negative exchange, and so on... Own symbols can be used.
- We don't just set up a company. We create a much broader system beyond the company's borders with clients, suppliers, partners, ... An analyst has to look even beyond the company's borders.
- Example: [Supply Chains, Distribution Networks](#)

The **Business Model** describes the global business idea.
It answers the question: “How do you (plan to) make money?”

It provides an overall framework for developing a coherent and streamlined and focussed set of activities to create value.

More detailed questions:

- What value is created?
- Who are your target customers?
- What customer’s problems or needs will be solved?
- What are the products and/or services?
- Who are the customers ready to pay for them?
- How do you reach, acquire and keep customers?
- How do you differ from the competitors?
- How do you produce the products and services (from a business perspective, not from a technical perspective)? And how do you get them to the clients?
- What is the turnover?
- What is the cost structure?
- What is the profit margin?
- *And so on..*

Note:

The “**Business Model Canvas**” is a template for developing a new or documenting an existing business model.

Business Model ≠ UML model

Business Modelling ≠ Business Process Modelling (!!!)

A business model is a response to an opportunity and has to suit the company, be appropriate for the product and market and fit into the company’s environment.

Business Model Canvas

18/12/2018

Key Partners	Key Activities	Value Proposition	Customer Relationship	Customer Segments
	Key Resources		Channels	
Cost Structure			Revenue Streams	

LIST OF TYPES OF BUSINESS MODELS

Add-ons/In-app purchases	Low-Cost model
Advertisement (Ad-based) model	Make More Of It
Affiliate/Referral fee	Manufacturing model
Aikido	Mass customisation
Auction model	Merchant model
Bait and Hook model	No frills model (discount or budget model)
Barter or Swapping for products	Open Business model
Barter or Swapping for services	Open Source model
Brokerage model	Orchestrator
Cash Machine	Pay-as-you-go model (PAYG)
Commission-based model	Pay-per-use model
Cross Selling	Pay-what-you-want (PWYW)
Crowdfunding	Peer-to-Peer
Crowdsourcing	Performance Based Contracting
Customer Loyalty	Premium model
Data model	Razor and Blade model
Digitalisation	Recurring Revenue model (subscription model)
Direct Sales model / Direct Selling	Rent Instead of Buy model
E-Commerce	Revenue Sharing
Early Exit strategy	Reverse Engineering
Experience Selling	Reverse Innovation
Flat-rate	Robin Hood
Fractionalised Ownership	Self-Service
Franchise model	Shop-in-Shop
Freemium model	Single purchase model
From-Push-to-Pull	Solution Provider
Get-one-give-one model (G1G1)	Subscription model
Guaranteed Availability	Supermarket
Hidden Revenue	Target the Poor
Ingredient Branding	Tip jar/donation
Integrator	Trash-to-Can
Internet Bubble model	Two-Sided Market
Layer Player	Ultimate Luxury
Leverage Customer Data	User Designed
License fees	White Label
Lock-In	
Long Tail	

Note:

Not an exhaustive list

These business models are like templates. The chosen model is a starting point and has to be adapted and further developed to suit the company's specific situation.

A **STRATEGY** is a long term plan, 5 years or longer, for a company (organisation) to realise a vision or to achieve a goal.

A strategy is (often) meant to change the position of the company in the market.

A **STRATEGY** is

a **PLANNED PROCESS** of **TRANSFORMATION** of the company

- the identity (values, culture, reputation, ...)
- the Business Model
- the company's Value Network and partnerships (partners, collaboration, suppliers, sources, ...)
- the business activities
- the company's location
- the company's capabilities
- the company as system
- the products or services
- the policies, rules, principles, ...

A strategy **“predicts” the self-initiated changes (under own control)**.

To **increase the chance of a successful execution** of a strategy

- The strategy has to be communicate to all the company's employees
- The enterprise (the system) has to be changed in time
- Information is required to understand the company and its activities and to understand how the strategy works out.

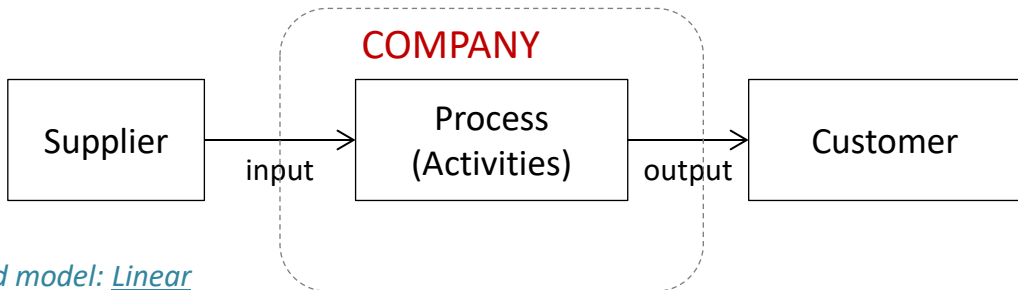
Notes:

This is the main idea of a strategy. A strategy may also be related to priorities, timing, policies, resources, leadership, company culture, competencies, internal organisation, and so on.

The timespan of a strategy concerns, by nature, years (like 3 to 10 years).

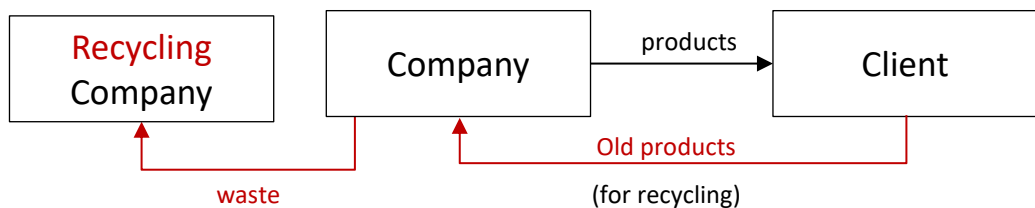
“strategic” is often confused with terms like ‘important’, ‘critical’, ‘key’, ‘fundamental’, ...²⁴⁶

SIPOC – Model

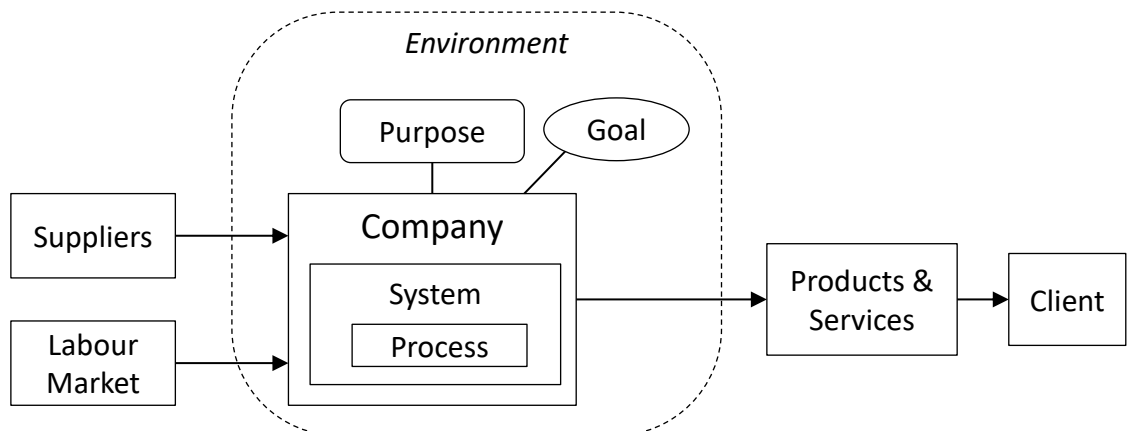


Limited model: Linear model. Not holistic.

Cyclic Model



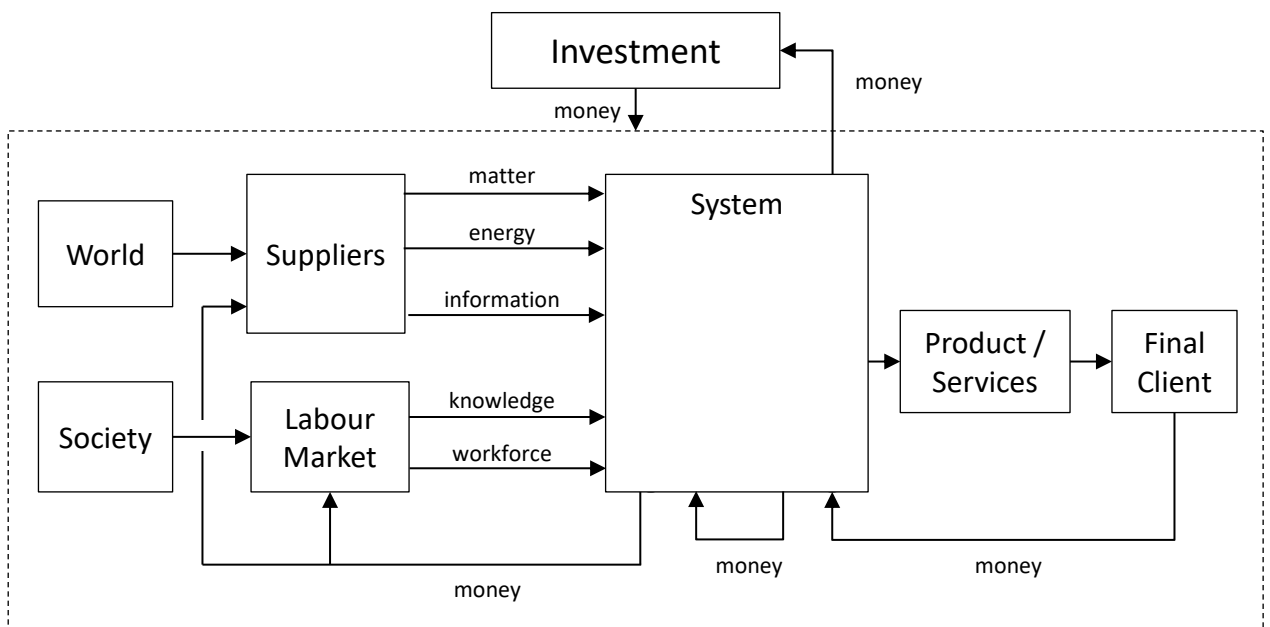
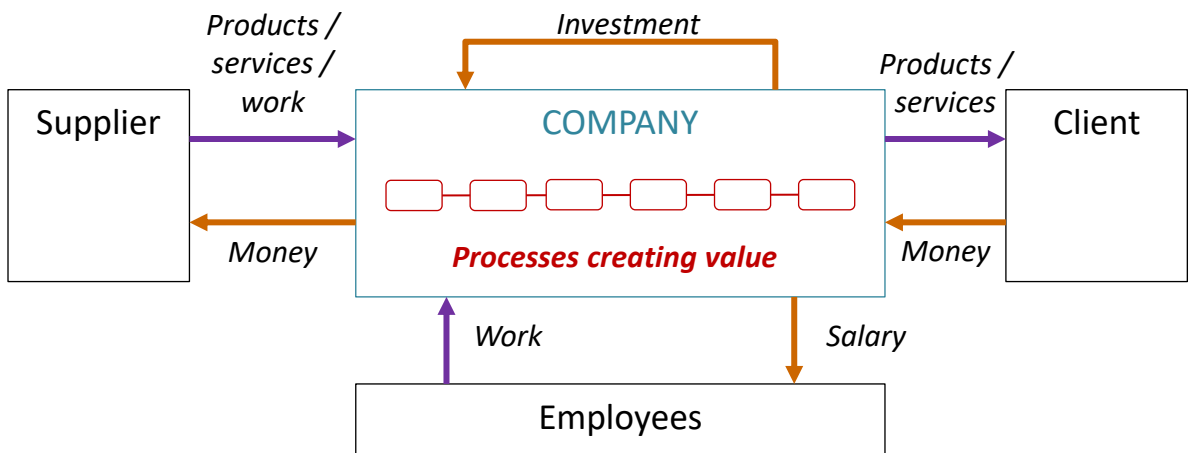
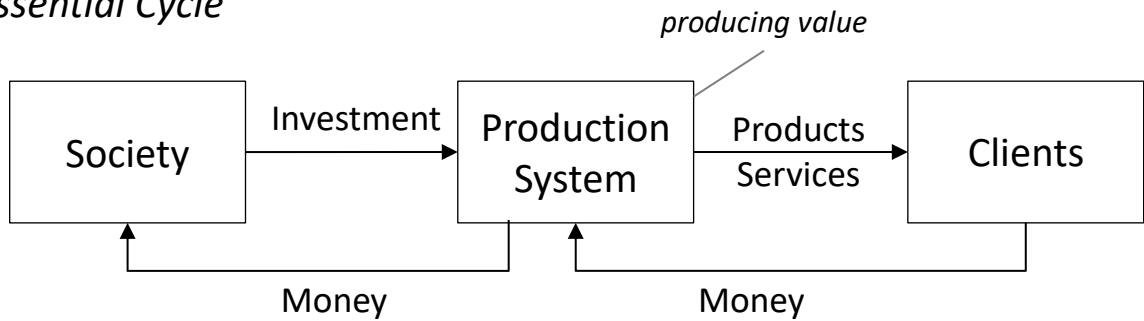
Remark: This is an example. Different cyclic models are possible.



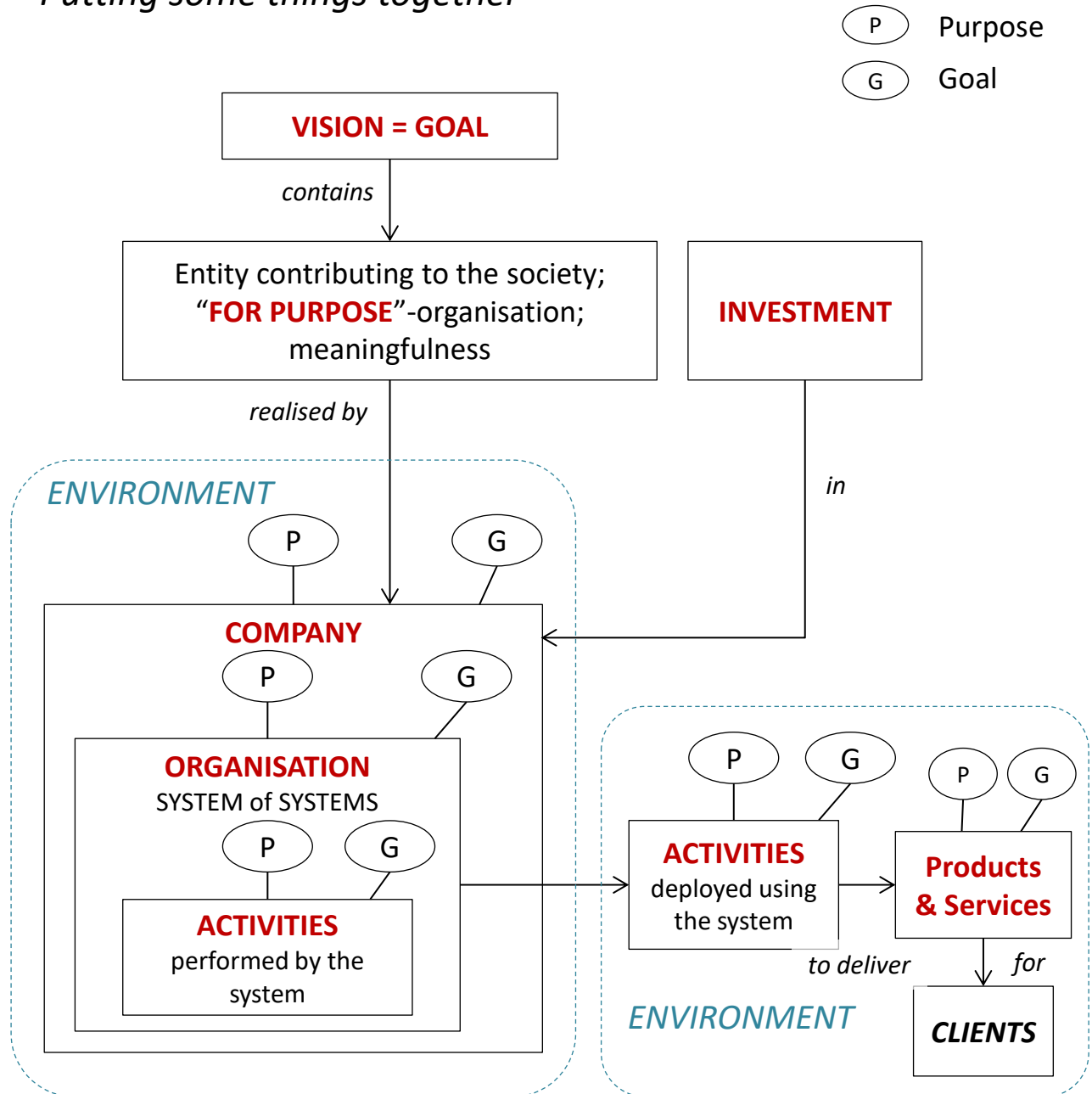
Basic Business Models

30/10/2018

Essential Cycle



Putting some things together

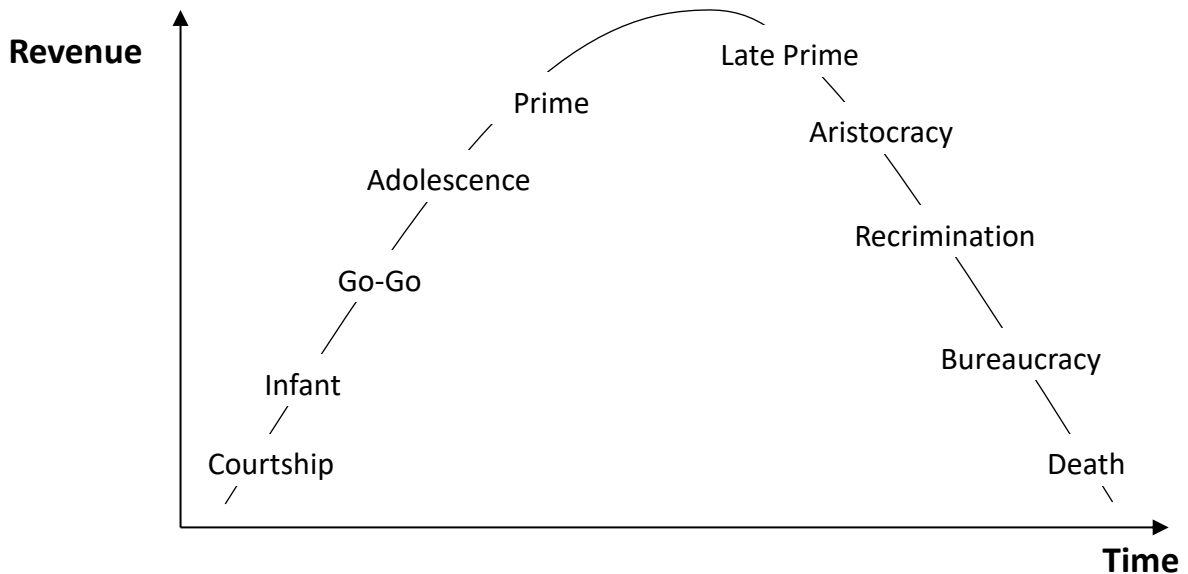


Notes:

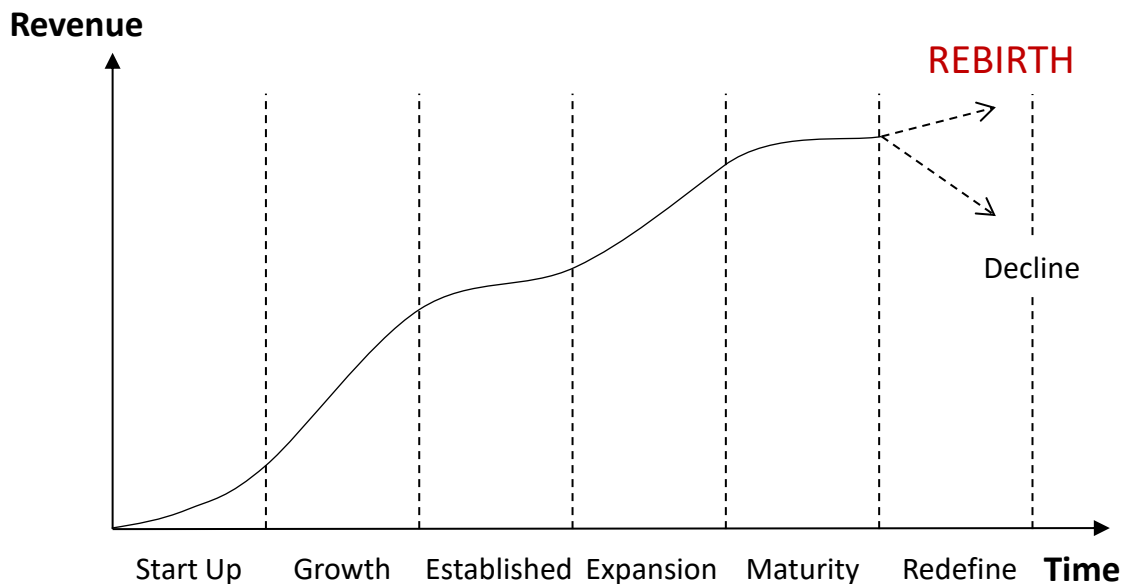
Everything has or should have a purpose and a goal (target, intention, reason, ...). Knowing it is important to understand what and why we do things, why they are necessary (or not), ...

Company's Life Cycle

30/10/2018



A more optimistic model ...



Notes:

Model may represent the maturity of a set of competing products on a market.

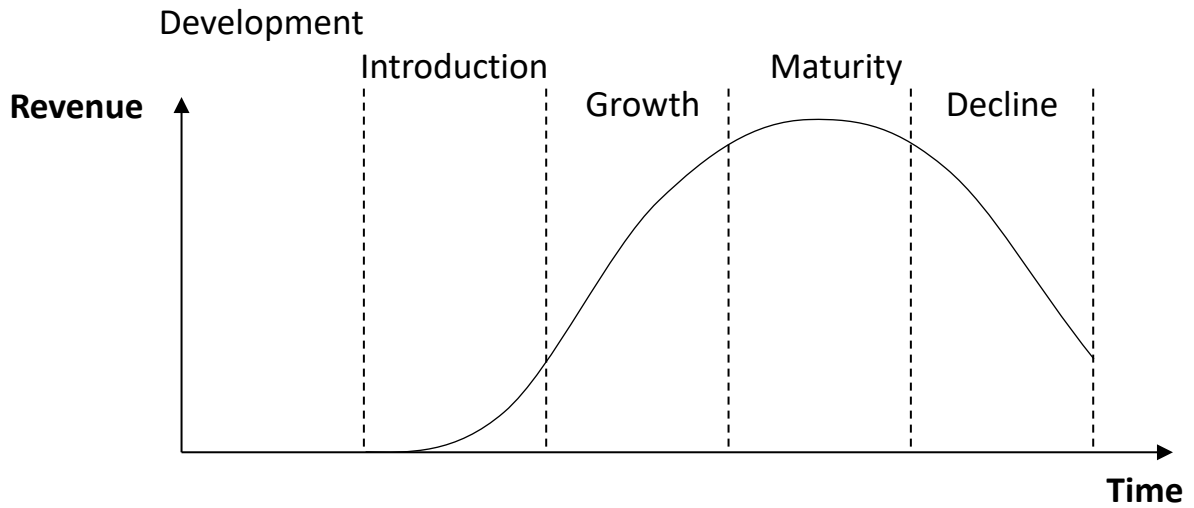
Cost of development, sales, sales revenue, profit, ...may be added to the graphic.

Why do companies end? To avoid at all cost. Sub-systems have to help to prevent this.

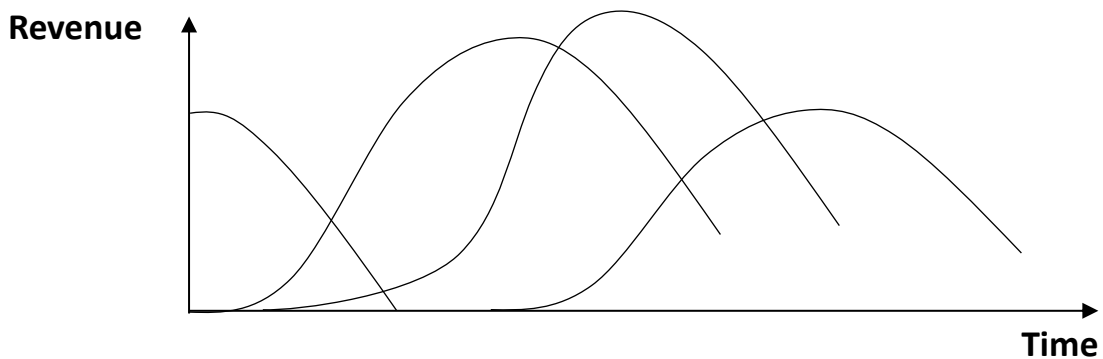
Simple linear model. Why? Can it be different? Why?

Product Life Cycle

30/10/2018



Company's product portfolio

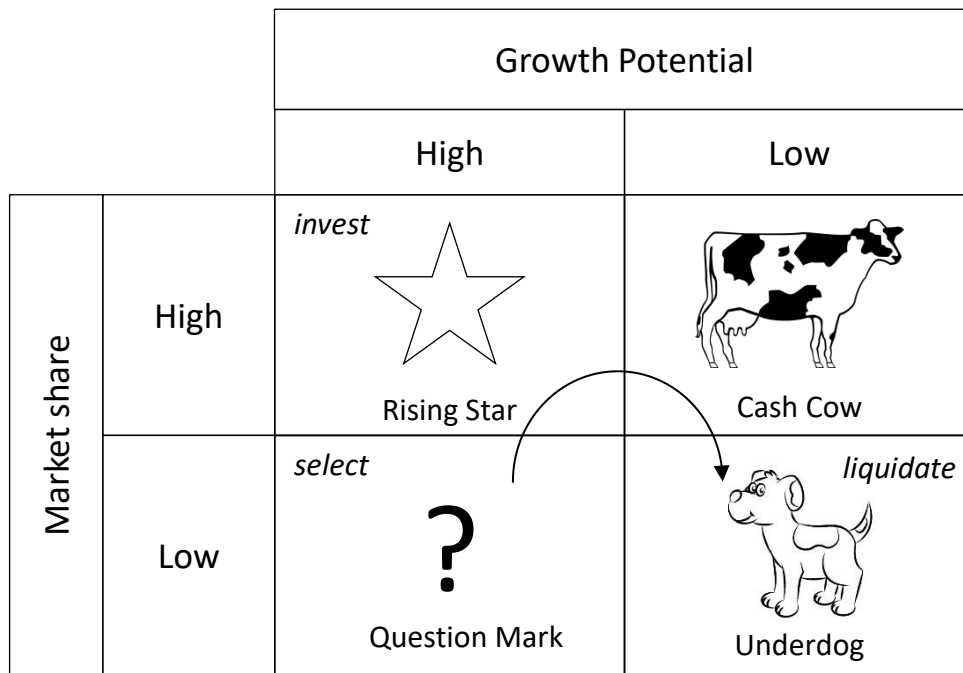


Impacts projects and systems.

A company strives to have always one or a set of growing to mature products to ensure permanent stream of income.

Product: BCG - Matrix

30/10/2018



Notes:

Question marks: Uncertain products, services, activities, opportunities

Rising stars: Selected good opportunities

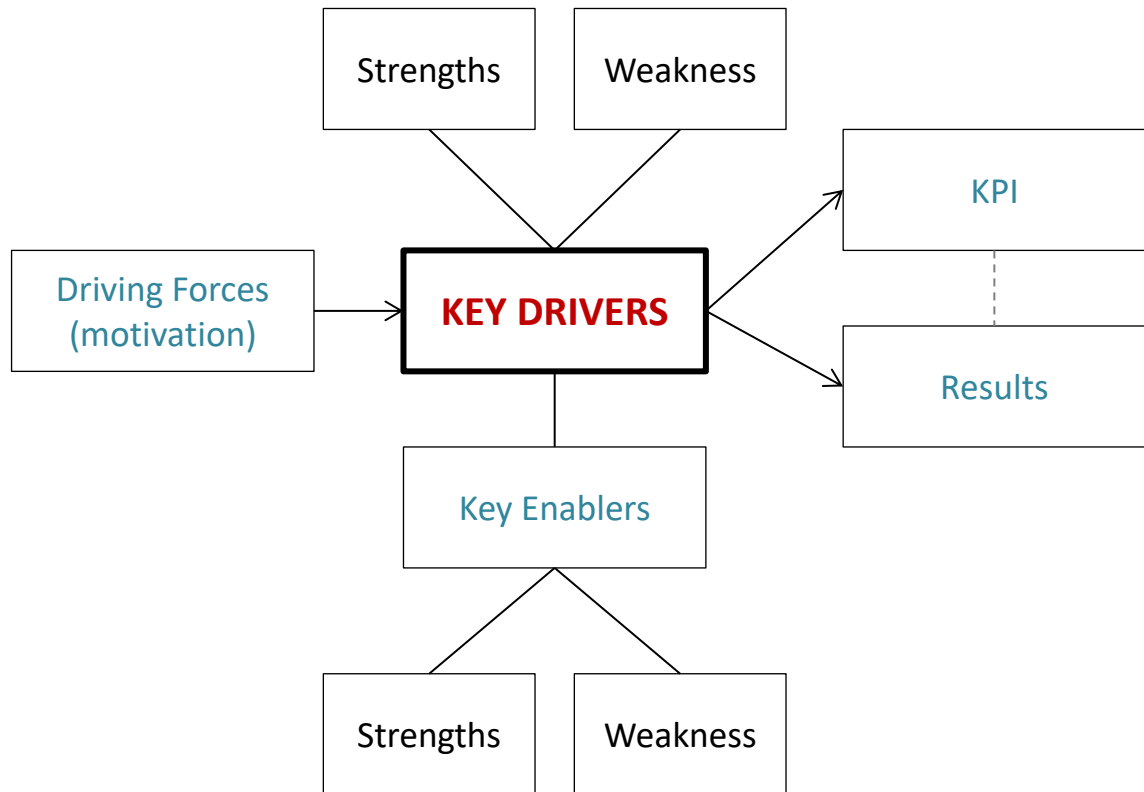
Cash cow : provide business volume, benefits, ...

Underdog: old products and services in decline. Revival (innovation) or disinvest?

The BCG-matrix must be considered in the light of the strategy and business priorities.

A company needs to have cash cows and rising starts that, later, will become cash cows.

Therefore it needs to invent new products that will evolve into new rising stars.



**Understanding quickly the essence of business
and where to look for improvements**

Notes:

Identify the few key drivers generating most of the business results.

Identify their main enablers allowing the drivers to function

Understand the driving forces (motivation, reasons, intentions, ...) driving the drivers?

Results are expressed as measurements of the KPI's (there are also PI's (Performance Indicators))

Know the strengths and weaknesses (+ risks) of the drivers and enablers.

The key drivers must be considered in the light of the BCG-matrix, the strategy and business priorities

Q: Why are some other drivers weak? Do they need to be strengthened and how?

Still need to understand the environment and the relation between this core aspect of the business and the environment.

Company's Evolution Drivers

18/12/2018

	Top-Down (mainly)	Bottom-Up (mainly)
External drivers for change		
• Markets	Y	
• Legal	Y	
• Clients	Y	Y
• Competitors	Y	
Internal drivers for change		
• Strategy, tactics, plans, priorities	Y	
• New activities	Y	
• Improvements	Y	Y
• Innovation	Y	Y

Analysis like SWOT, PESTELI (or PESTLE), Michael Porter's Five Forces Model and others help to identify external changes.

Implication: The question is not whether a top-down or bottom-up should be followed (dualistic thinking). Main directions are mainly coming from the top. Top-down is clearly the major approach. However, bottom-up approach should be use as well and should be integrated in the approach.

Events the company may have to deal with during its lifetime (not exhaustive):

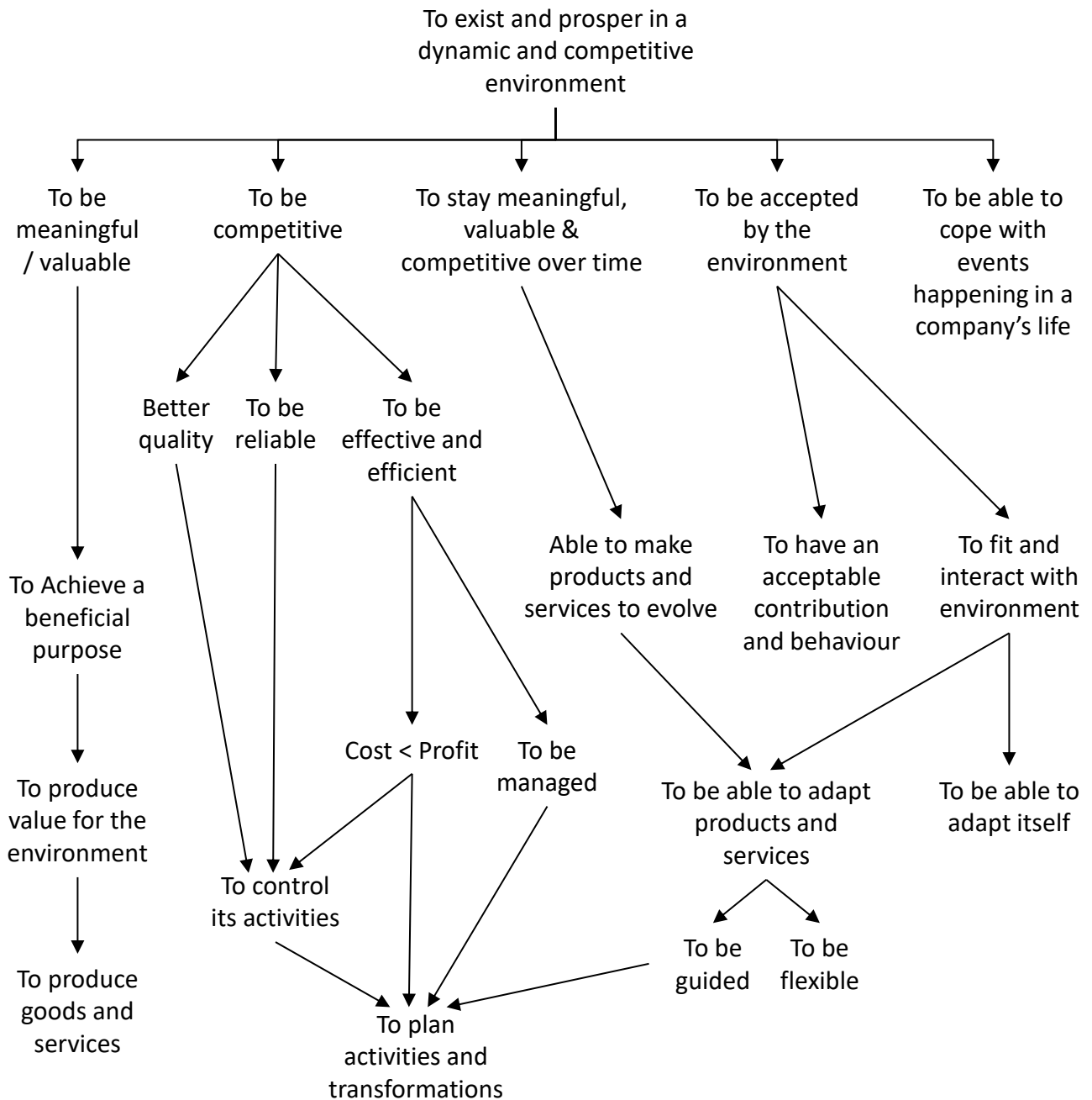
- Adapting the company culture
- Changing the business model
- Changes in strategies, policies, tactics or priorities
- Organic growth
- Maturing organisation (optimisation, formalisation, ...)
- Restructuring, rightsizing, downsizing
- Acquisition
- Merging
- Relocation of activities (like offshoring)
- Externalisation of activities (like outsourcing, privatisation, ...)
- Disinvestment
- Diversification
- Consolidation
- Takeover
- Bankruptcy
- Relocation of HQ or plants
- Changes in product portfolio or in service portfolio
- Changes in customer base
- Change of geographical market location
- Changes in investment and sources of revenue and profit
- Changes in distribution and transportation network
- Changes in partnerships and inter-organisational collaborations
- Changes in work philosophy, approach, methods ...
- Technological evolution
- Evolution in automation and computerisation
- Digitization
- ...

Notes:

How well can the architecture and systems cope with such events? Does the implemented system hinder these events? How to prepare for them? How well can the engineers deal with such an event?

Essential Requirements for a Company

30/10/2018



Notes

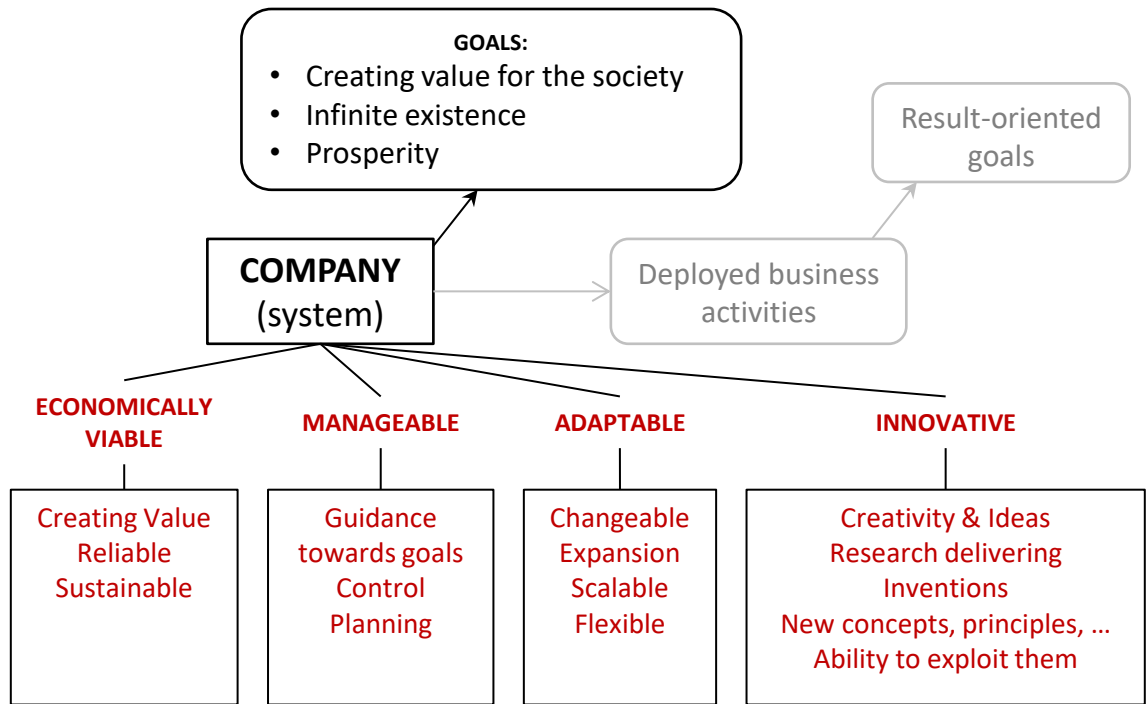
This reasoning can be further detailed.

Since a company is also an investment → goal, guidance, predictability, control.

Control, predictability and planning are required, as well as flexibility and adaptability. A balance must be achieved. Sometimes this will lead to a trade-off. Sometimes these are not conflicting and both can be achieved.

Company's Critical Characteristics

30/10/2018



The sub-systems and components have to respect these requirements.

These required qualities have to be engineered in the enterprise. They can be found in the principles, capabilities, structures, concepts, features, mechanisms and characteristics.

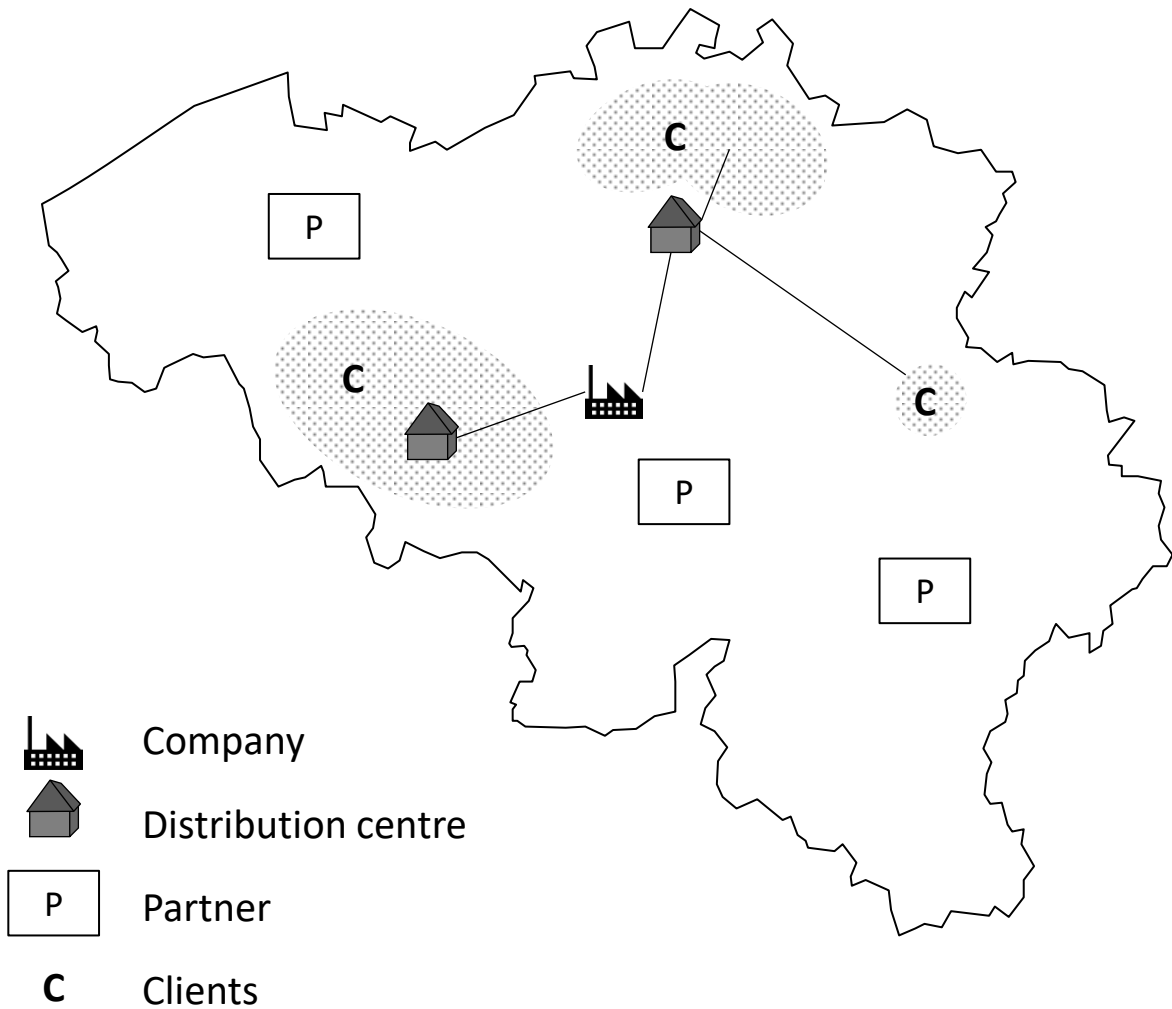
They influence the **systems design**, the **engineering philosophy** and **methods**, the **development processes**, the **transformation initiatives**, ... but also **roles** and **competencies** of people (authority, responsibilities, activities, interpersonal relations & collaboration).

Notes:

- The listed characteristics are critical for the company "as a system" (for the engineers). From a broader perspective, other characteristics like leadership and culture are critical as well.
- These characteristics don't appear by themselves in the enterprise and its systems. They have to be engineered in it.
- Each characteristic can be further detailed:
 - Sustainable: effective, efficient, legal, ethical, planet-friendly, socially respectful, ...
 - Flexible: elegant architecture, modular design
- None of these characteristics or deduced characteristics can be ignored or neglected, at any level of engineering.

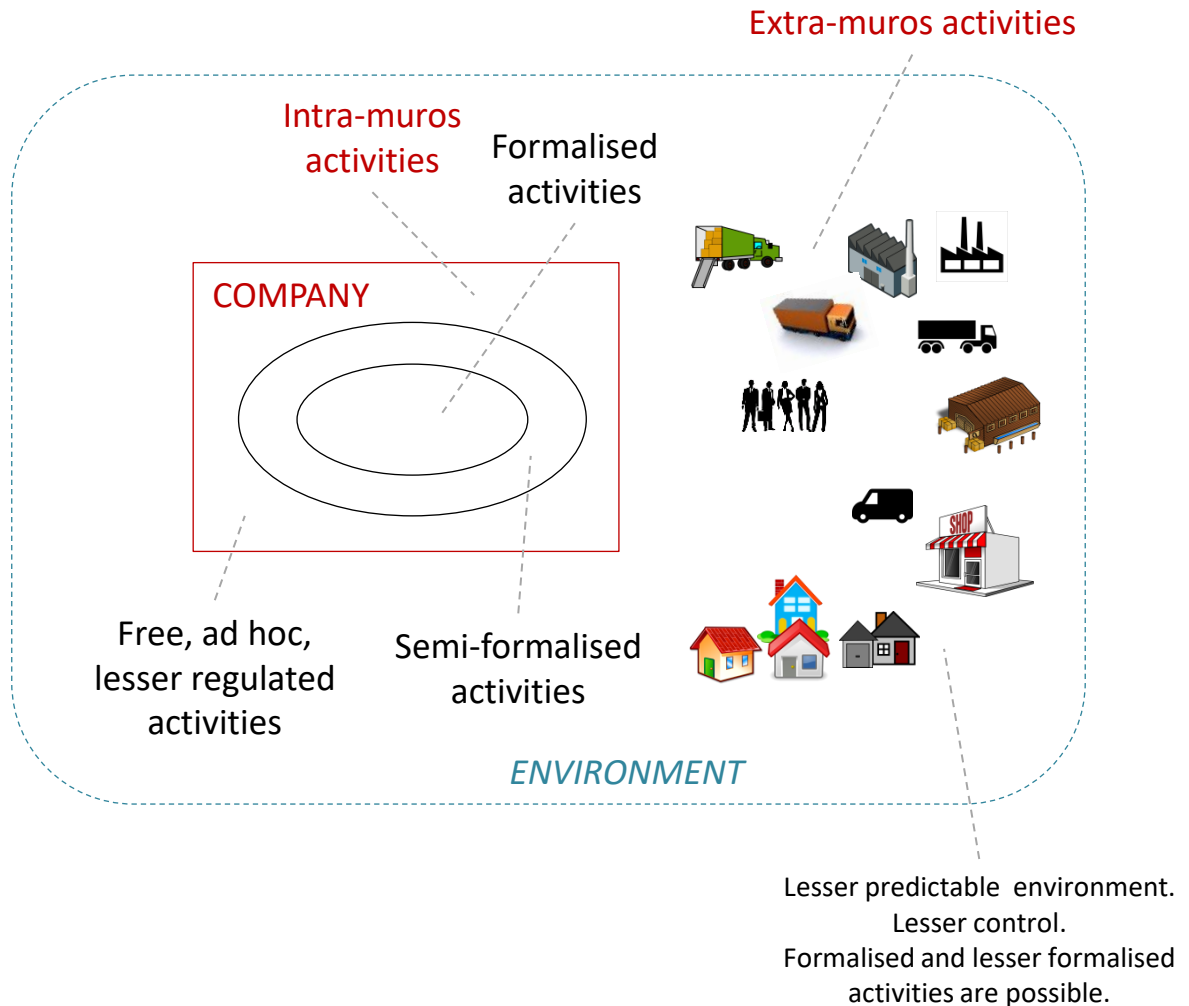
Geographic Organisation

30/10/2018



Notes:

The geographical situation often plays a role in the organisation of the company and in the architecture of processes and systems.



Notes:

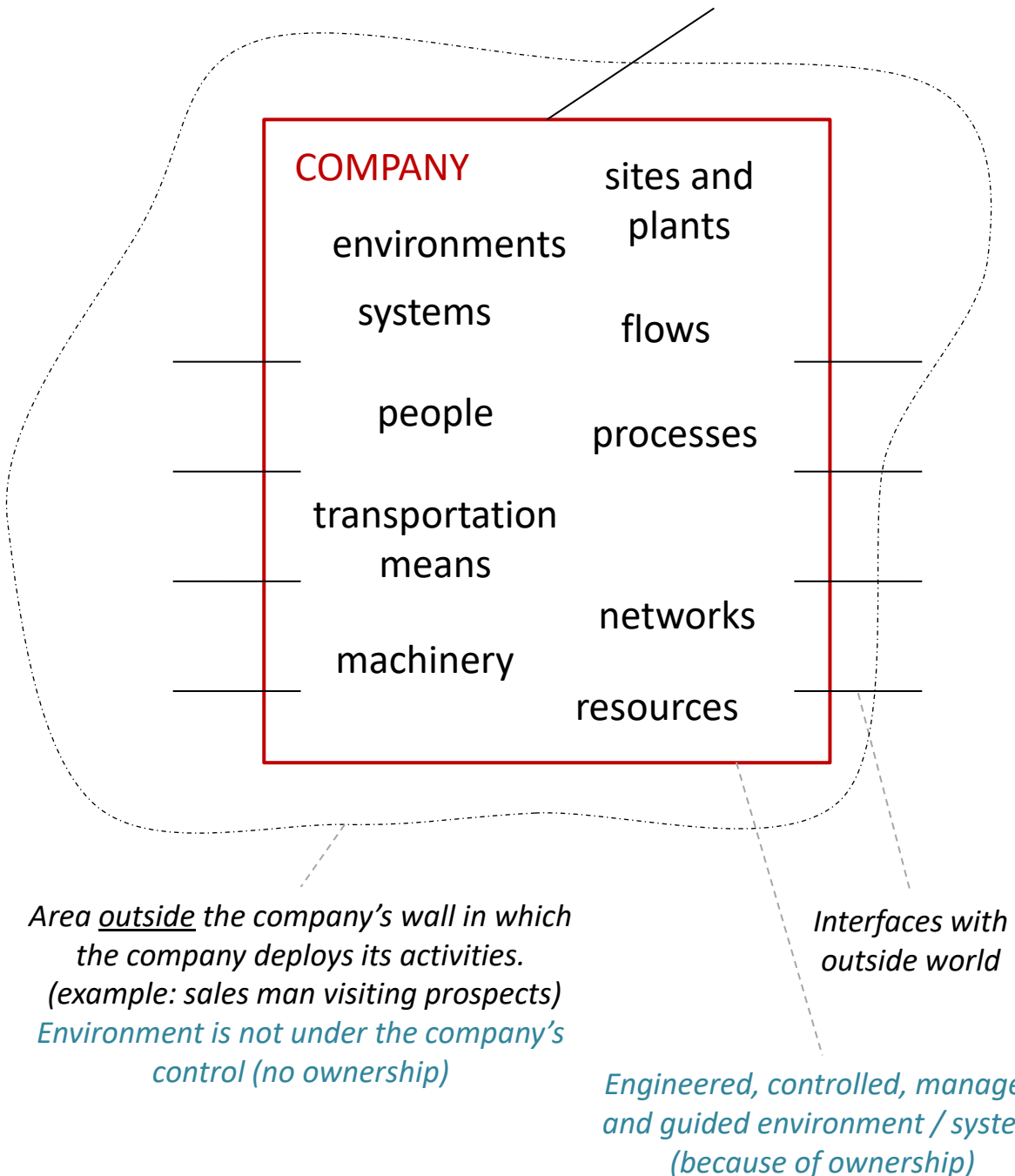
Inside a company a system can be more cohesive, more tight. Outside the company the systems tend to be more tentacular and lesser bound to a location (web applications, Apps, ...).

Inter-organisational information systems, like supply chains, are somewhere in between.

Unfortunately, often the term Interorganisational Systems (IOS) is used. This term is too vague and should specify that the system has to be an information system. A energy distribution network, a transport network between organisations, a close collaboration between two or more companies are all systems between organisations.

The Company – “Intra Muros”^{30/10/2018}

*harmonious and managed system,
seamless collaboration*



*Area outside the company's wall in which
the company deploys its activities.*

(example: sales man visiting prospects)

*Environment is not under the company's
control (no ownership)*

*Interfaces with
outside world*

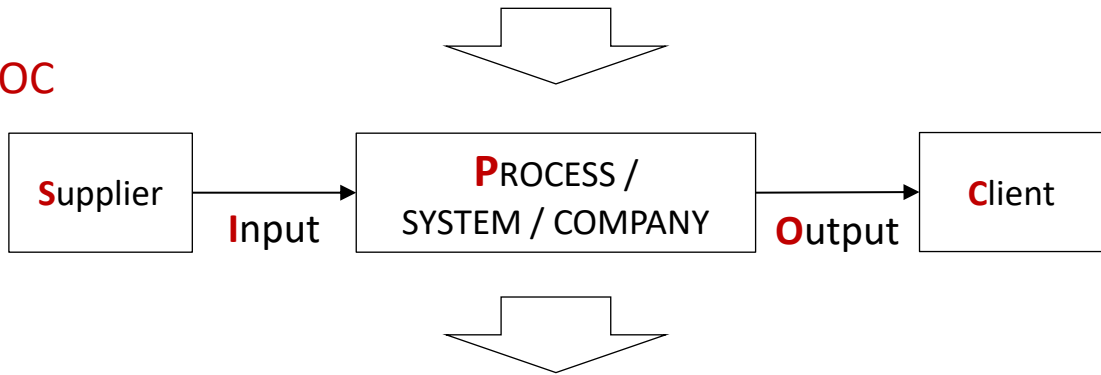
*Engineered, controlled, managed
and guided environment / system
(because of ownership)*

Business Operations Model (Operating Model)

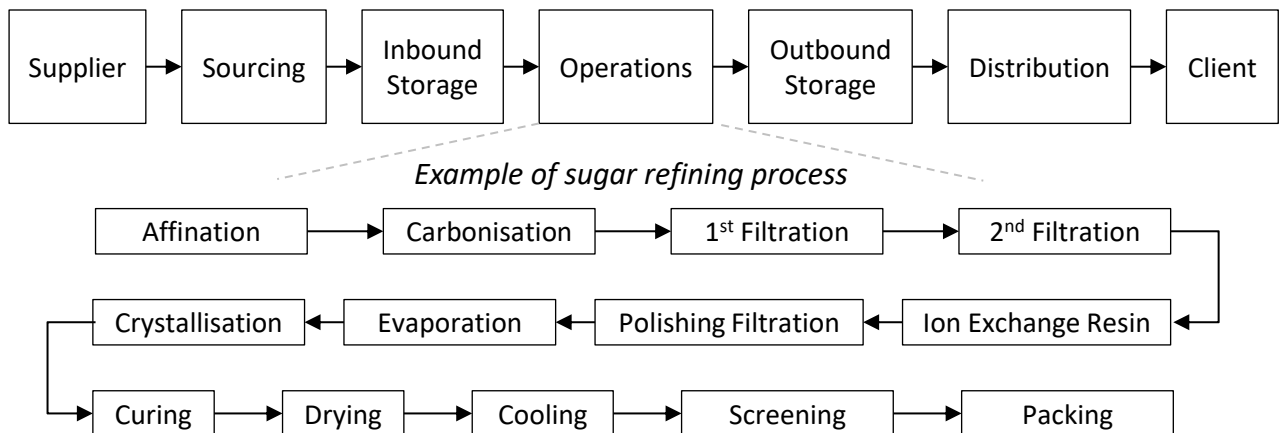
30/10/2018

Business Model & Value Network
(value creation and value exchange models)

SIPOC



Value Stream: level above the Business Processes

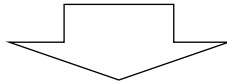


Methods: Value Chain Model, Value Stream Model, Rich Pictures, Flow Diagrams, ...

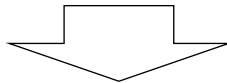
Notes:

The Business Operations Model describes the way the structures, processes, flows, assets, roles and competencies work together allowing the company to function and to achieve its goals.

SIPOC

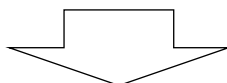
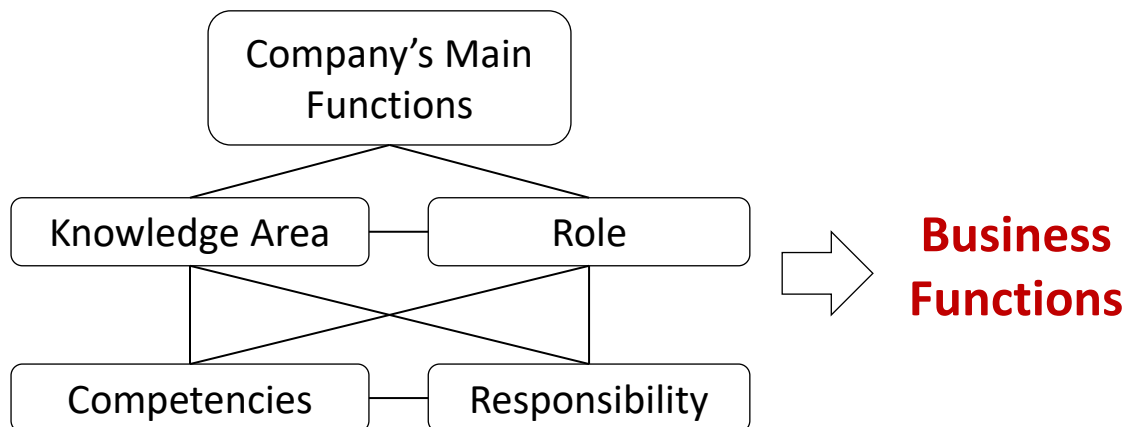


Value Chain Model / Value Stream Model



Business Functions Model

Model organising the company's main functions

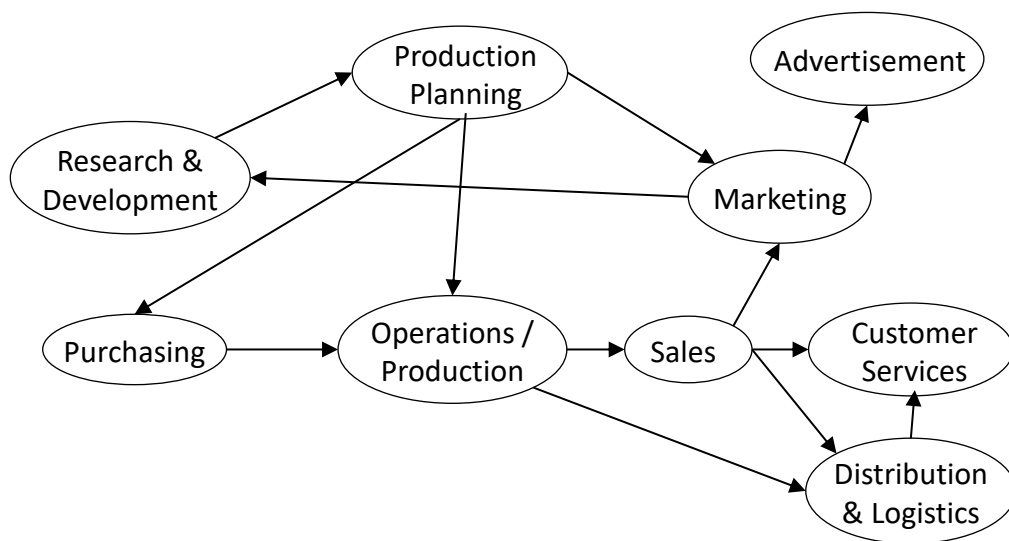
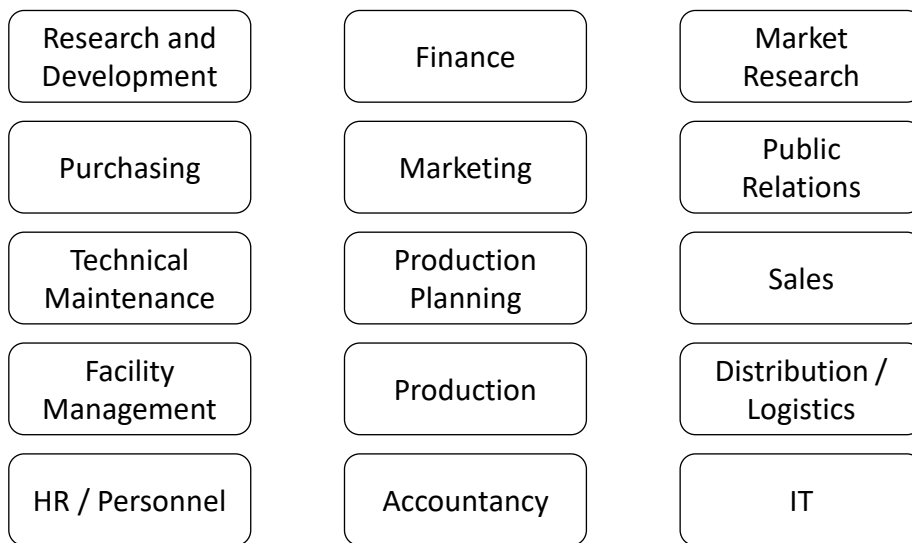


Organisational Units

Business Functions
≠
Organisational Units

Business Functions

30/10/2018



Notes:

- Organisation of activities.
- Organised per business knowledge area and main expertise, per purpose, per goal, per responsibility and authority,
- Types (production, selling, support; internal, external) can be indicated.
- Can be enriched with main relations or flows among business functions.
- Business functions often coincide with divisions and/or departments. However, business functions \neq department or other organisational units.

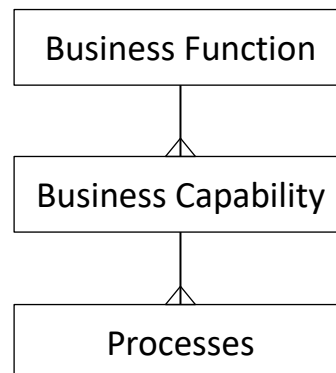
A **Business Capability** is the **organisation's ability** to perform a unique function or **unit of business activity**. This ability is the result of the **intentional** implementation of a defined **set of necessary means**.

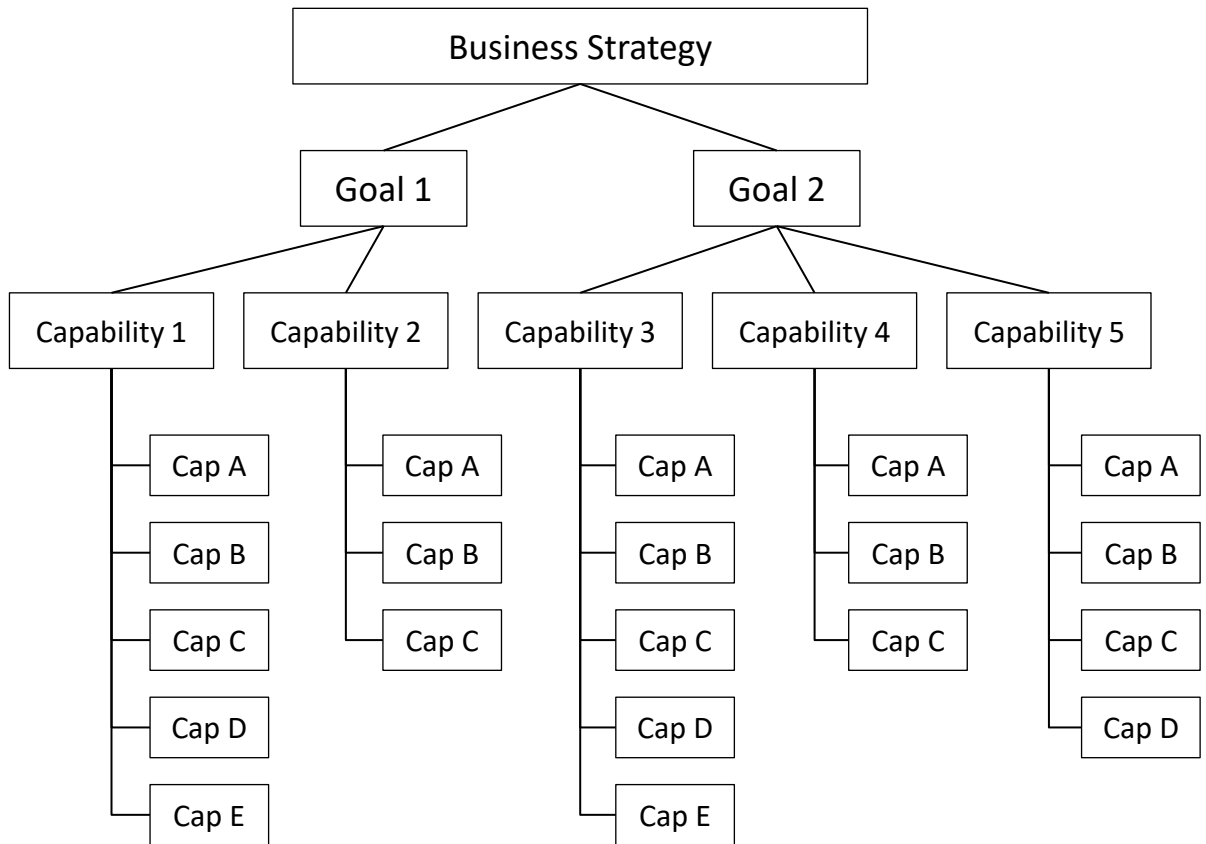
In management: allows to manage functional strengths.

In enterprise architecture: a building block to define the business architecture and to translate it into the enterprise architecture.

Business Function - Business Capability

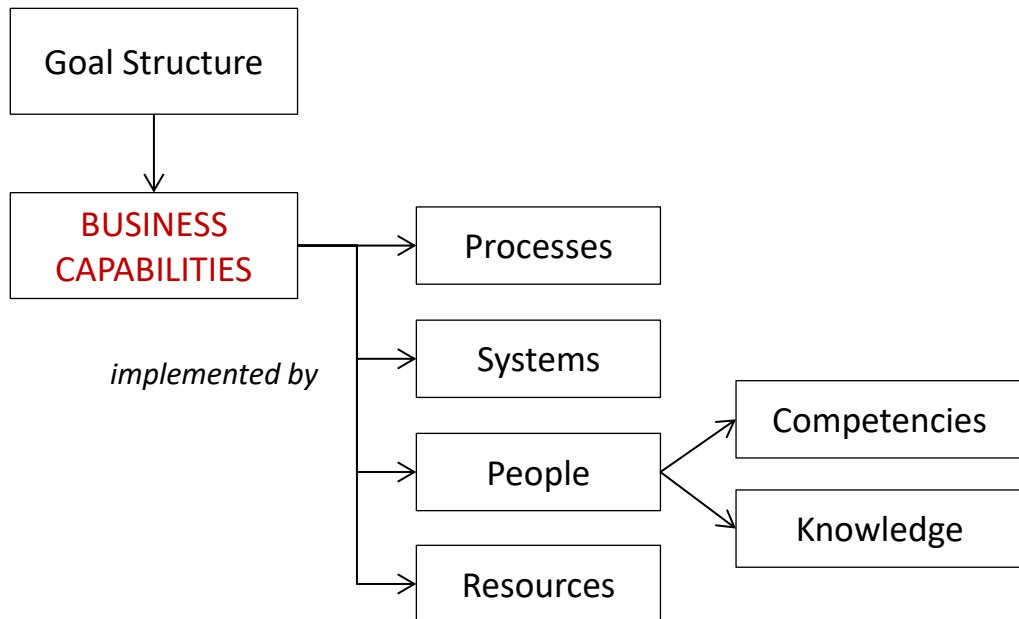
Business Capabilities can be grouped under a Business Function. A Business Capability is a sub-function of a Business Function. A Business Function is rather an abstract grouping used in models. A Business Capability is, when implemented, more concrete since it links between means and ability.





Related:

- Business Capability Map
- Capability Management

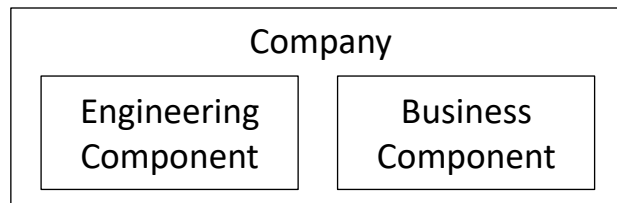


As example, an approach can be deduced:

- Definition of goal structure (for example, deduced from the strategy)
- Definition of the required business capabilities necessary to reach the goals
- Define the hierarchy of capabilities (key indicators and targets)
- Link the business capabilities to the business functions. Are they present in the business functions? To what is their strength? Is there a gap?
- Determine the required/involved processes
- Determine whether the processes are aligned with the required capability.
- Determine the required/involved systems.
- Determine whether each system is aligned / implement its part of the capability.
- Same for organisation (people), competencies and resources.

Notes:

- The process can be adapted for listing the available capabilities, for verifying if a plan or forecast is feasible, to identify a gap, weaknesses, ...
- This model can be enriched with risks, limitations, links to on-going projects, ...



BUSINESS CAPABILITIES

Core Capabilities

Essential capabilities required to fulfil the company's purpose. These capabilities should be strong, solidly embedded and optimised. They should be better than those of competitors. Innovating here is certainly useful.

Strategic Capabilities

Critical capabilities especially developed to enabling the execution the strategy and which are developed. They are developed to execute the strategy. Innovation in these capabilities is very important. Over time, they may become a core capabilities.

Common (Business) Capabilities

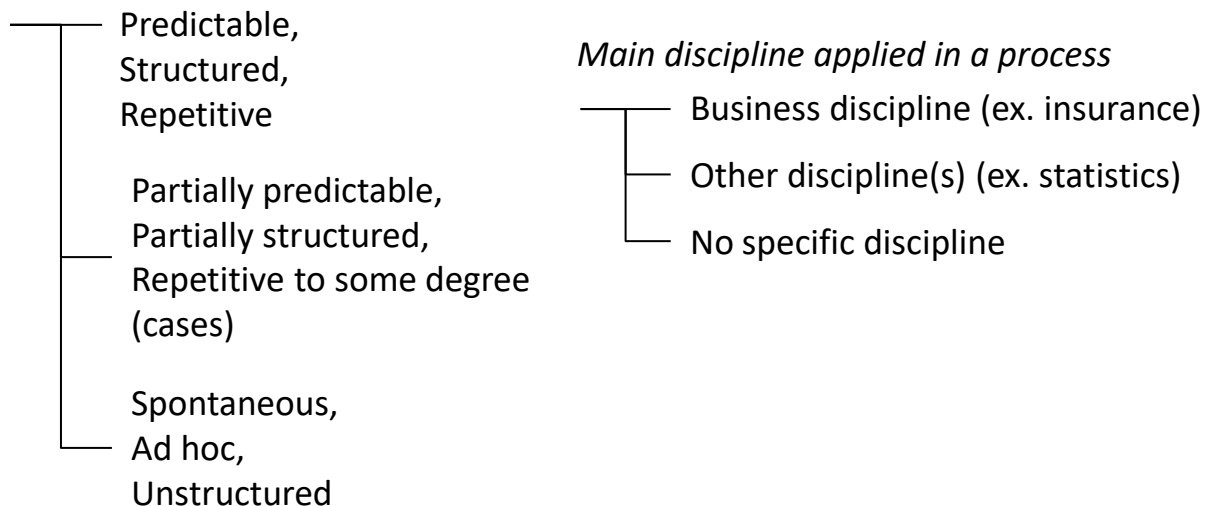
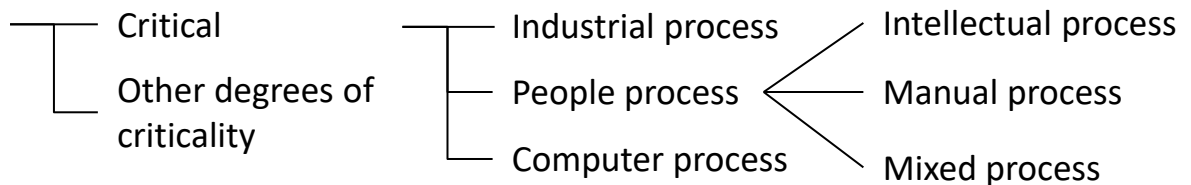
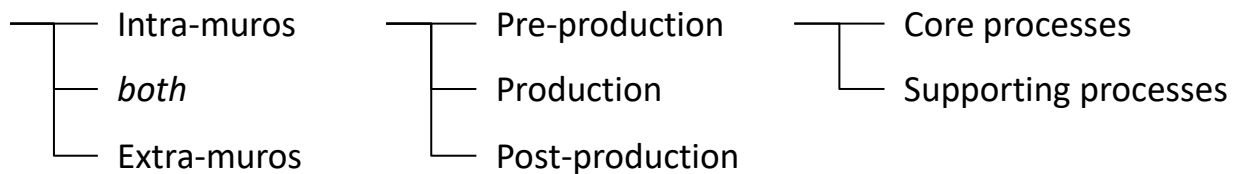
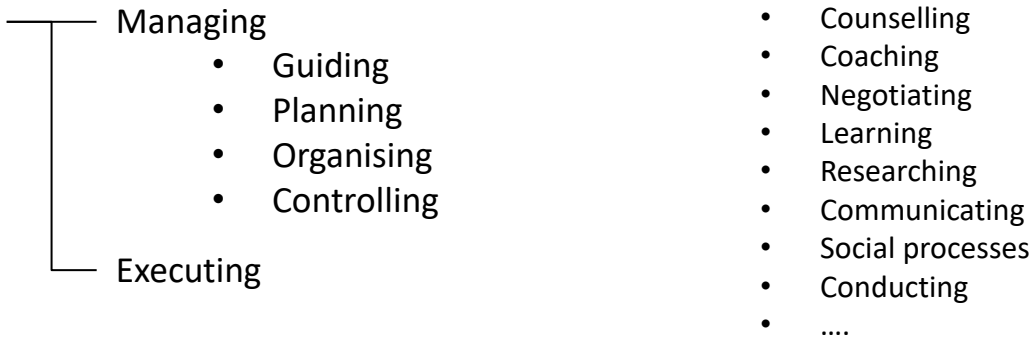
These are usual capabilities.

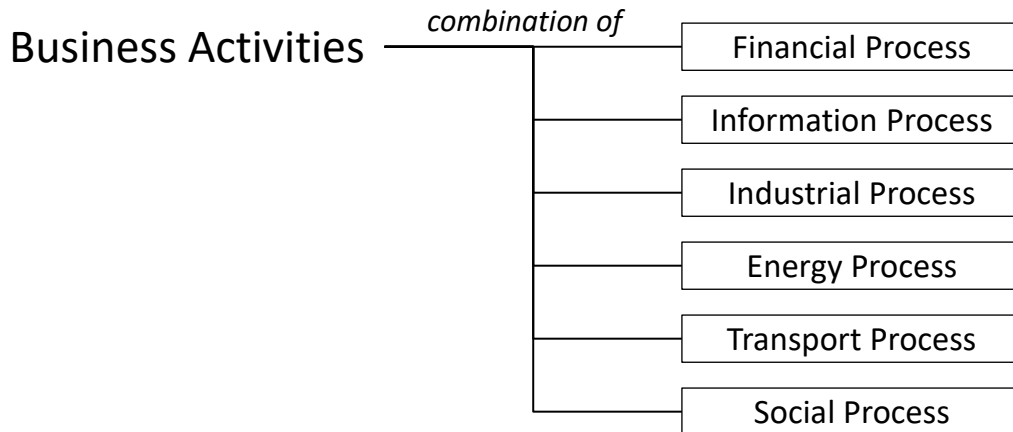
Supporting Capabilities

These capabilities provide support to all the three other capabilities. Examples are HR and warehouse management capabilities.

ENGINEERING CAPABILITES

To be developed: like Engineering Capabilities (R&D, Industrial Engineering, Product Engineering, Service Engineering, ISE, ...), Common Capabilities, Supporting Capabilities

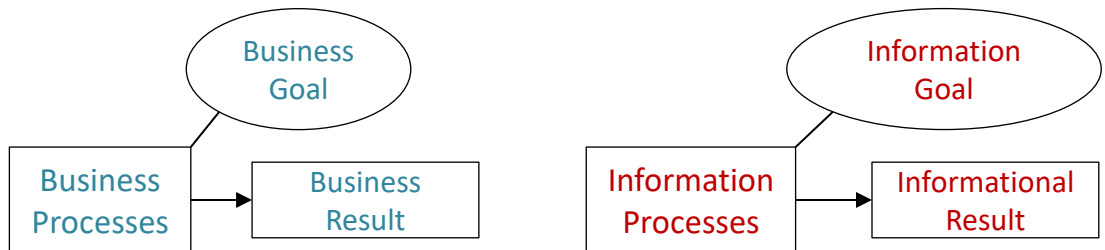




Business Processes	Financial Processes	Information Processes	Industrial Processes	Transport Processes	Energy Processes	Social Processes
Information Processes						

Some information processes may support other processes.
And, all process types contain some information processes.

Business Process vs Information Process



A Business Process structures the process. It may process matter, money, information, .. It may include transport or social processes. Business processes implement always some information logic. They are expressed only in business terms (business vocabulary only).

Information processes process only information. They **may** implement business logic. Their concern is how to capture, processes and/or distribute information to perform business activities or in order to support them.

Not all processes executed by the business community are Business Processes. A Business Process can be decomposed into more detailed processes. These more detailed processes are not Business Processes. Some other detailed processes exist which aren't Business Processes either.

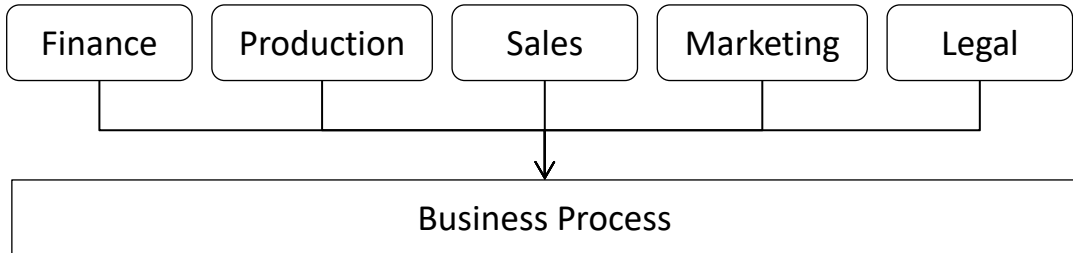
An Information Process can be a Business Process. It is advisable to consider the business perspective of the process separately from the information perspective.

We may not simply label every software feature or software process as '*Business Process*'.

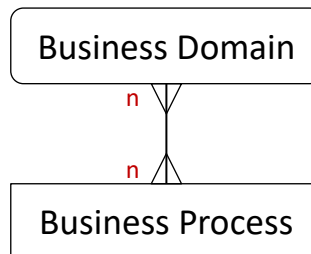
Business Departments, Domains and Business Processes

20/10/2018

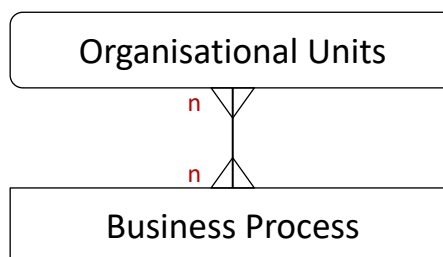
Business Processes & Organisational Units



One business process may implement logic of different business domains



Business Processes & Organisational Units

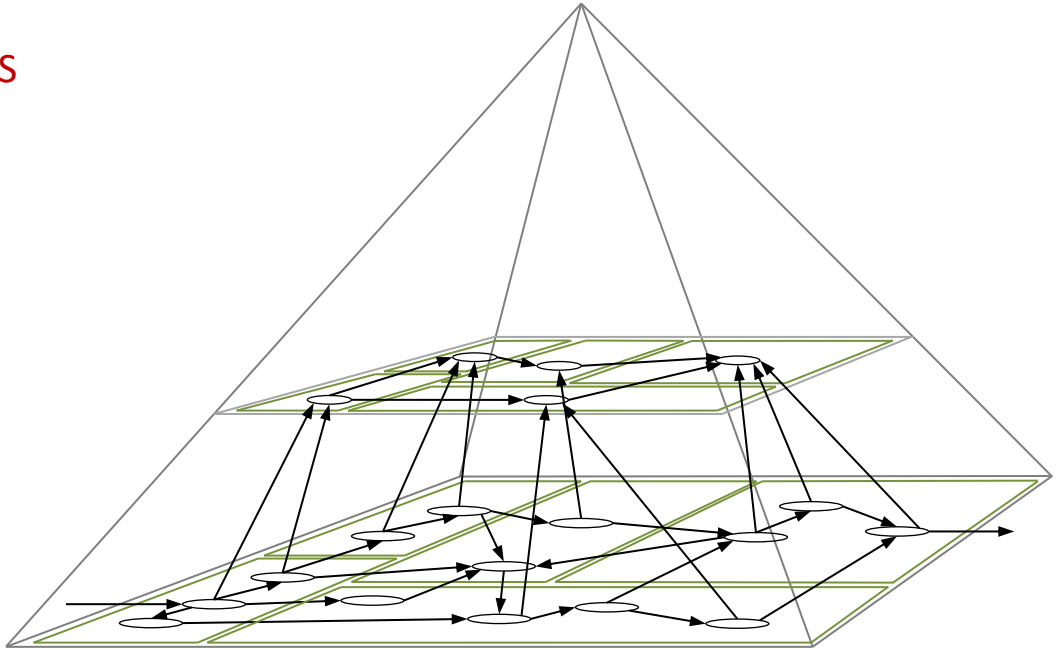


Each organisational unit has its own goals, priorities, plans, needs, intentions, rules, ...

Notes:

Business processes are (often) a collaboration between business domains and between organisational units (like departments, services, ...). Analysing or designing a business process may require consulting different experts and taking into account the variety of goals, contexts, priorities, ...

PROCESSES



PROCESSED RESOURCES



matter



information

EXECUTORS



machines



people



Computer / software
systems

FLOWS inside the company



matter



energy



information



Knowledge
(information)



Money
(mostly information)

System / Company

- Structure
- Mechanisms
- Channels
- Storage

Flows of

- **Information**
- Knowledge
- *Money*
- Matter
- Energy

- People
- Machines
- Computers

Robots (Computers+Machines)

- Infrastructure / Networks

- Buildings, Terrains

A Company as Heterogeneous System of Systems

System of Systems

- People organisation
- Industrial systems
- Accounting system
- Cost system
- Communication systems
- Transport system
- Software systems
- ...

Heterogeneous

Systems are of different nature

Example:

- people organisation \neq software system
- financial system \neq industrial system

Question:

- Have to fit, to collaborate and to function harmoniously.
- Requires different domains of expertise and engineering skills and methods

Difficulty!
How to solve this?

DIFFERENCE BETWEEN COMPONENTS

30/10/2018

COMPANY			
Executor	INDUSTRIAL COMPONENT	HUMAN ORGANISATION	INFORMATION COMPONENT
Way of execution	Automated	Manual	Automated
Processing	Material, Energy	Material & Information	Information
Nature of processes / processing	Pre-defined processes	Interpretation, variable logic, ad hoc decisions, explicit and implicit knowledge	Pre-defined processes

1. People are able to deal with higher level logic. They have the capability to interpret, to take ad hoc decisions and to adapt. They can deal with high-level more vague instructions, with non-formalised logic, variable process execution and with intuitive knowledge. Not everything is or need to be pre-defined, formalised, ...
2. The Industrial and Information component contain pre-defined, hardwired (even if configurable) logic. Faster and more reliable process execution.

People organisation and automated components are very different.

May lead to misunderstandings, conflicts, ...

Notes

These are the three most common main “executing” components in a company. We can also have an “energy component”, “transport component” (doesn’t matter). Automation is done either by moving work to the industrial component, or by moving work to the information component (reduction of the people organisation?)

People can learn. Able to acquire new skills.

People can deal with fuzzy situations.

The mind activities and products of the mind are partial, distorted, subjective and fuzzy. These are lines of thoughts, ideas, visions, beliefs, motivations, intentions and expectations.

Difference between the talk, thinking and actions.

Intellectual activities are not linear. Their pace is highly variable.

People are creative.

People can inform themselves, investigate, research, ask, interpret, evaluate, take decisions and adapt their actions accordingly.

People's mind is variable. The understanding changes over time. Ideas may pop up. Preferences may change. Different decisions can be taken.

People tend to create a set of beliefs and an understanding that makes sense to them and which is coherent.

People have social relations which are dynamic, temporary and of different intensity and quality. This impacts collaboration.

People influence each other for the best or for the worst.

People tend to copy each other.

People like simplicity and dislike complexity and problems.

People are bad problem solvers.

A person can get tired or sick.

Many factors influence the speed, efficiency and quality of activities of people.

People need to relax, take a break and have some rest, days off and vacation.

People are different, have different qualities and require different work environments.

People may use freedom well or abuse of it. With freedom comes responsibility.

People react on fear, uncertainty, lack of information, threats, and so on.

And so on... !!

The strengths, weaknesses, limitations and other characteristics of people have to be taken into account when designing IS's, but also when working in projects or elaborating methodologies.

SOFTWARE SYSTEMS

- Designed to deal with information
- Implements information capabilities
- Implements information processes
- Solves (or help solving) **information problems and information needs**
- Man-made
- Have purpose
- Process information incredibly faster and way more precise than people and can deal with huge amounts of information
- Information is stored more reliably (lesser volatile than people's memory)
- Work defined by predetermined logic (with AI, things can change)
- Fully formalised systems
- User interface is only a very limited and superficial view on the software
- Expresses high-level logic as well as extremely detailed logic
- (Pure) Business logic is only a part of the implemented logic
- Weaknesses: interpretation, improvisation, ad hoc decision making and thus ad hoc adaptation (because of prior detailed definition of the logic (slow and tedious))

MACHINES

- Man-made
- Have specific purpose
- Transform or transport materials, usually faster, more precise and at lower cost than people
- May reduces risks for people
- Machines produce repetitive work with similar results (good or bad)
- Work is predefined
- May need information; may contain software (robots)

The company consists mainly of people, processes, industrial systems (not in administrative companies), information systems and information. They are organised in architectures. These architectures need to be aligned. They are inter-dependent.

People Organisation

Best known. More on next pages.

Industrial Component

Only applicable for industrial companies. Out of scope in this paper.

Process Organisation

- Structure of the execution of the processes
- Structure of the logic of the processes

Information processes are executed by people, information systems

Processes dealing with matters are executed by industrial systems

Information Architecture

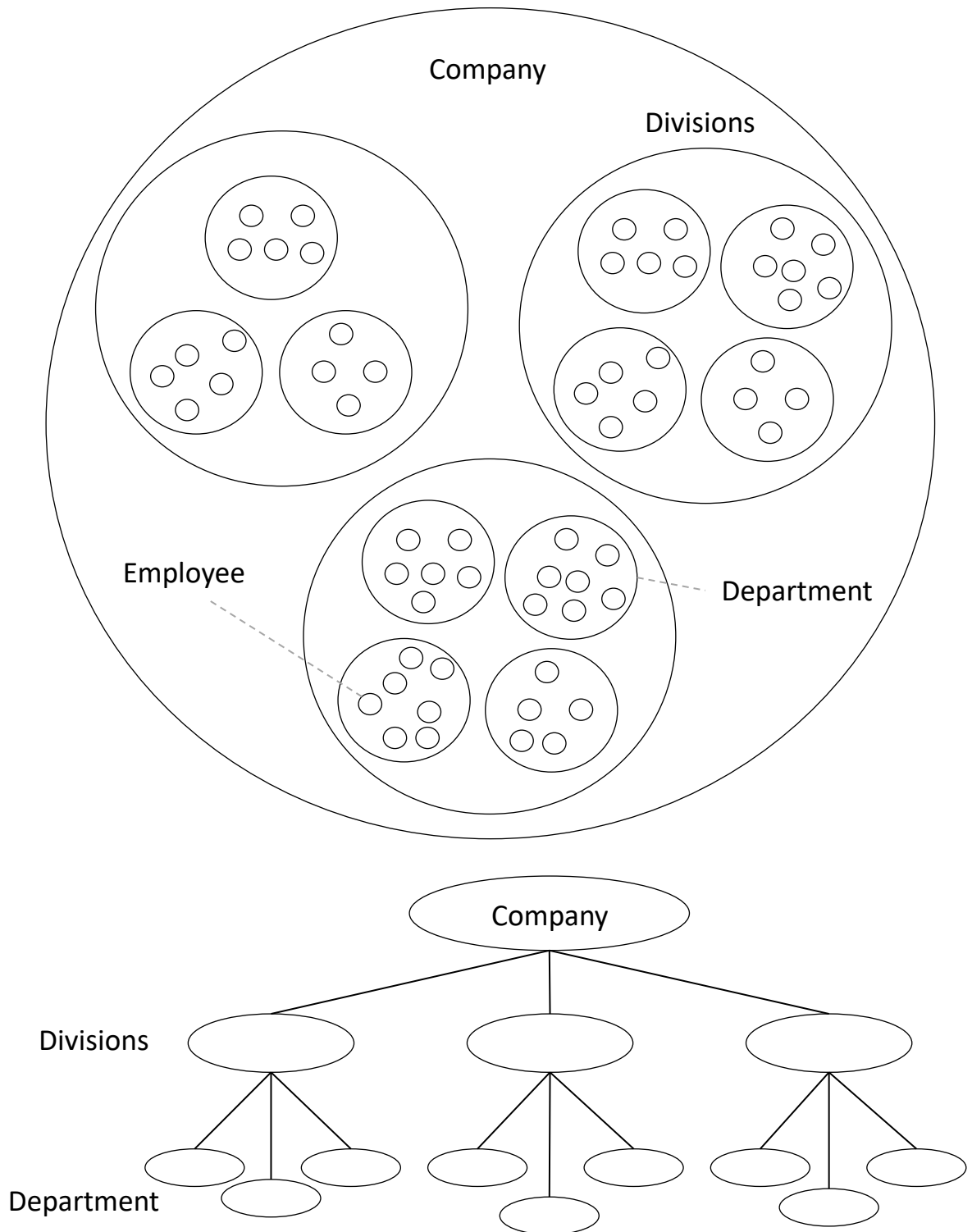
The information in a company is not a collection of databases. It should be organised around a global high-level information model.

Information and Software Systems Architecture

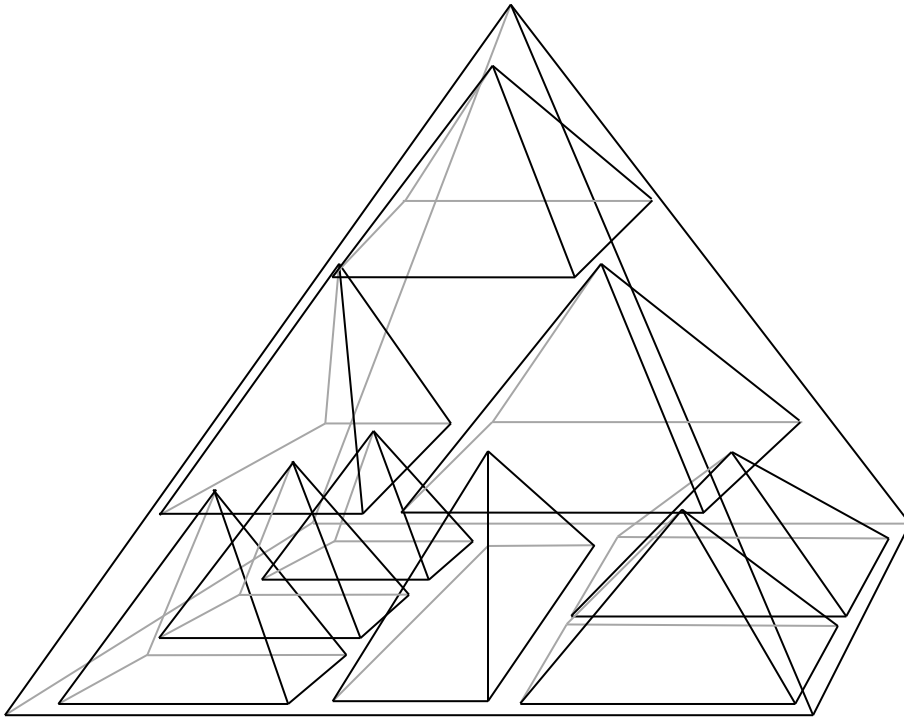
Organising information systems and software systems around an architecture is likely to be more effective and more cost-efficient than having simply a collection of systems.

Organisational Units

30/10/2018



These models show, as example, a common basic idea of dividing a company into organisational units. Other organisational structures and names are possible.



The company can be seen as a pyramid of pyramids. Flat organisations contain lesser layers of pyramids, while larger more hierarchical organisations are based upon more superposing layers of pyramids.

A “pyramid” can be temporary (like for a project initiative).

Organisations executing a lot of project may chose for a matrix organisation.



Notes:

Organisational units often receive the names of business functions. The relation between both is often very strong.

Traditional and common way to represent a company

Represents:

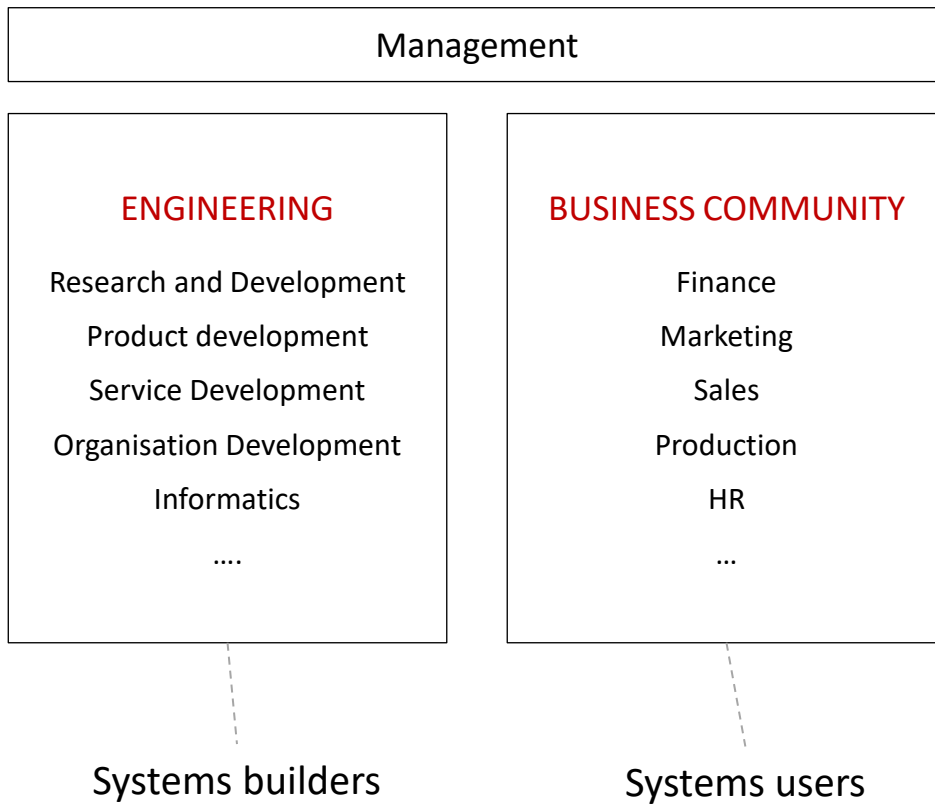
- units, departments, services
- or management hierarchy (names and roles)
- or both together

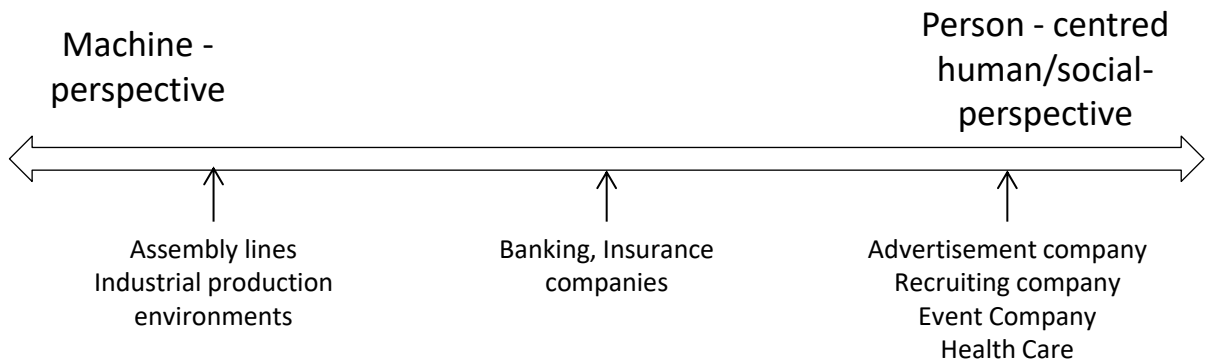
Reflection:

Is it beneficial for the company as a whole to allow the departments to define themselves how they work? Why or why not?

A Logical Company Organisation

30/10/2018





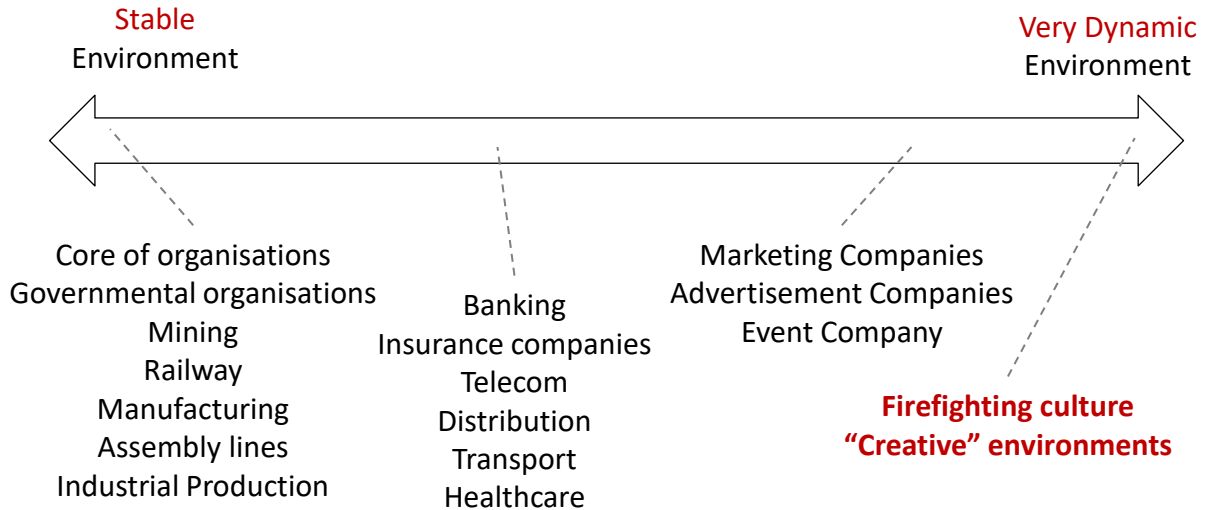
All organisations are partly ...

- organised like a **machine** (formal structure, procedures, processes, controls, specific skills, cost structure, ...)
- a **human** and **social** aspect

These aspects are present to different degrees depending on their economical activity but also depending of the part of the organisation (product development, customer service department vs industrial production line)

Notes:

Engineering IS, or larger parts of the organisations, has to take both perspectives into account. The weight and priorities must match the specific nature of the organisation.



Essence:

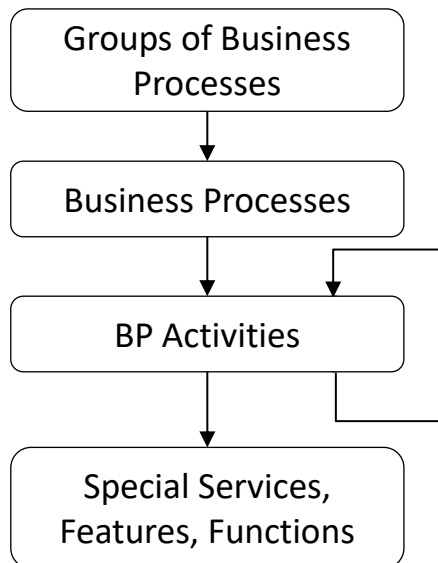
Some industries, some parts of a company are rather **stable** environments. Other industries or departments are **very dynamic**.

Equipping **dynamic environments** with a solid Information Component is **much more challenging**

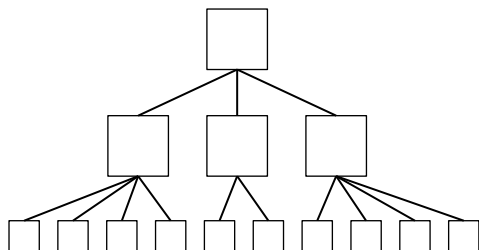
Remarks:

- Only rough estimations (can be inaccurate, subjective, ...)
- Depends on the department
- Depends of the company culture
- Every company is different
- Know your environment

Process Architecture (Execution Perspective)



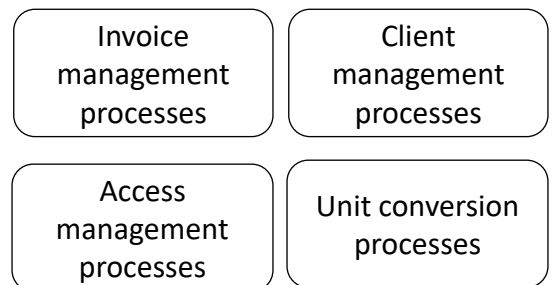
Decomposition of the processes as it appears when executed. Function and services are called and executed as needed.

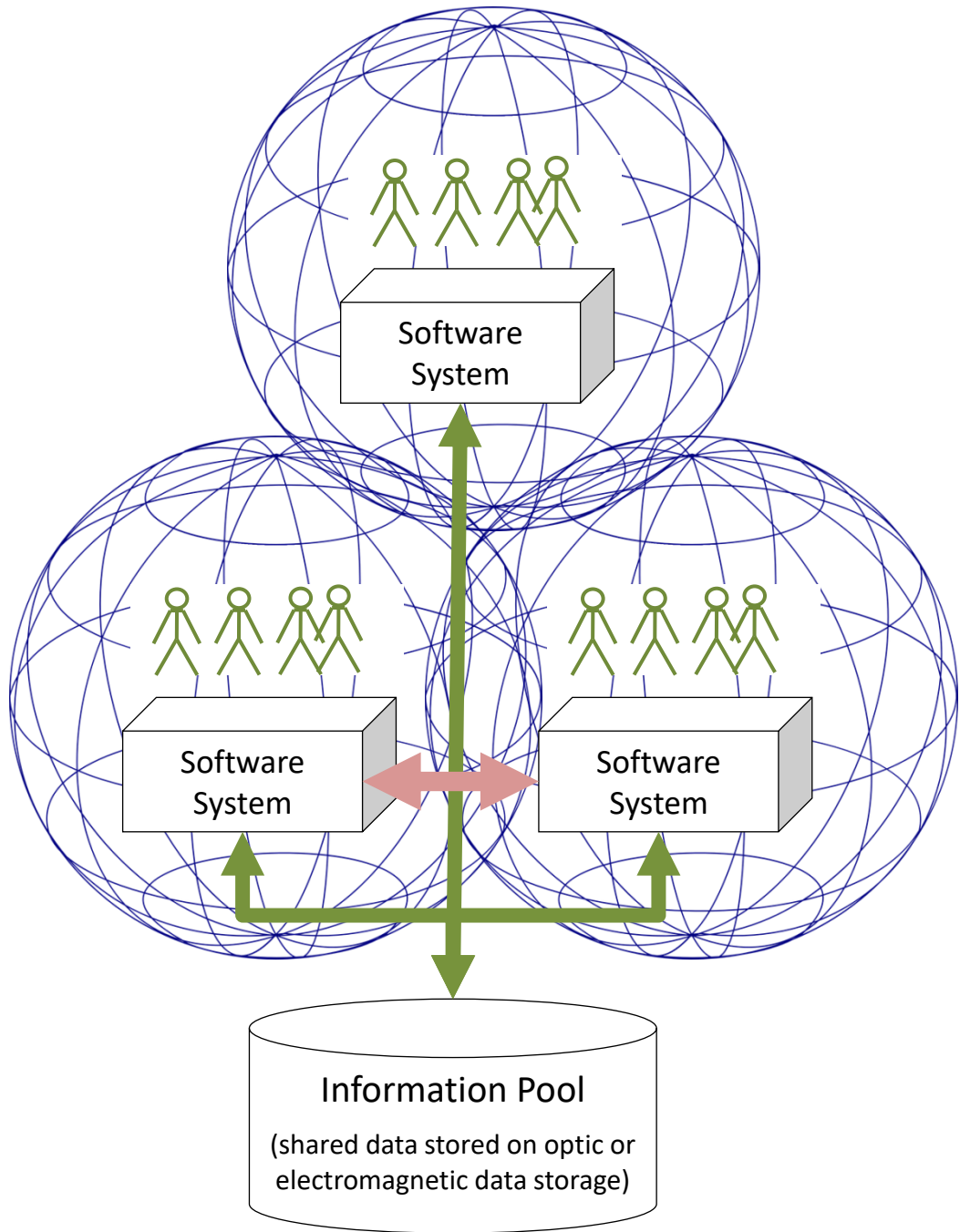


Process Logic Architecture

Organised in systems, libraries, components, modules, service software agents, ...

This architecture organises the process logic, like the source code. They are usually grouped by purpose or by information domain to be accessed.





Notes

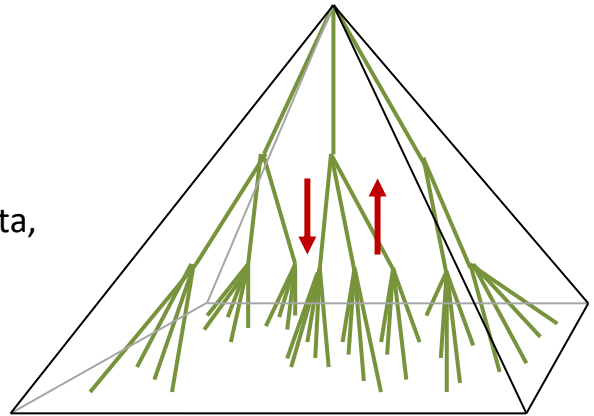
- IS's share information
- IS's communicate with each other. IS spheres overlap.

VERTICAL FLOWS

These flows concern mainly **budgets**, **information** for guidance and management, like plans, guidelines, policies, authorisations, performance data, results, ideas, ...

These flows move up and down through the organisation.

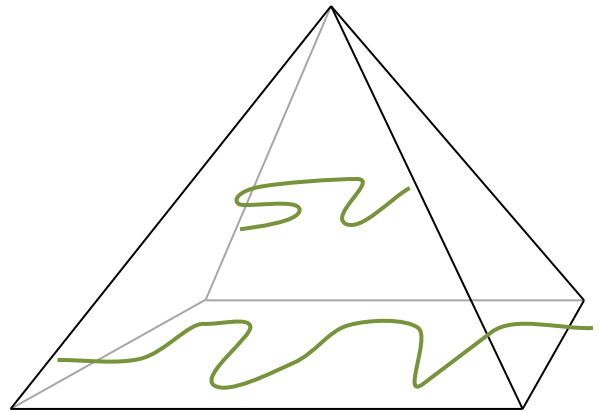
It often concerns **selected**, **filtered**, **aggregated**, **global** data, or in the other direction, a **refinement** occurs as the flows goes downwards.



HORIZONTAL FLOWS

Horizontal flows are often embedded in business processes or concern **sharing** and **collaboration**.

Information, **material**, **'work'** and **knowledge** tend to flow horizontal. Horizontal flows happen at different levels (in layers). Detail of information across the flow remain the same.



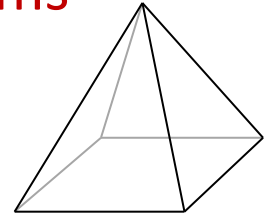
Flows happen not only inside systems, but also between systems. A flow represents a **dependency between systems**. Flows help to achieve final outcomes. Today, no system can work in isolation anymore.

Flows are critical and should be engineered

We tend to engineer architecture and systems, but we tend to forget the flows.

Electricity, water, gas, computer networks, telecom networks reach to all corners of the company.

Very important internal global systems



Goal Hierarchy System

System for analysing, decomposing, deducing, aligning and spreading goals across the company

Planning System

Elaboration and alignment of the various strategies, tactics and plans across the company

Communication System – Communication Channels

Established communication channels or processes across the company

Decision Making System

How decisions are being made and spread

Incentive System

Gathering and aligning all the incentive policies or mechanisms across the company

Creativity System

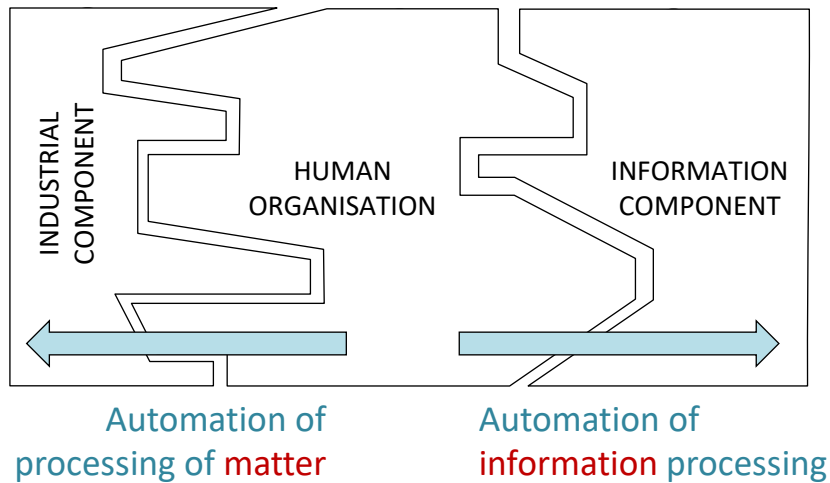
System to capture ideas across the company, evaluate them, and so on.

Other interesting/potential systems: knowledge system, collaboration system, network system (people), investment-profit system, priorities.

Notes:

- The key here is to establish first about the mechanism, the logic of the system. Thinking about software comes later as a way to implement the solution. ('system' ≠ 'software system')
- Each or most system should include evaluation, control and feedback mechanisms.
- These systems are not independent: goal, decision making, planning and incentive systems.

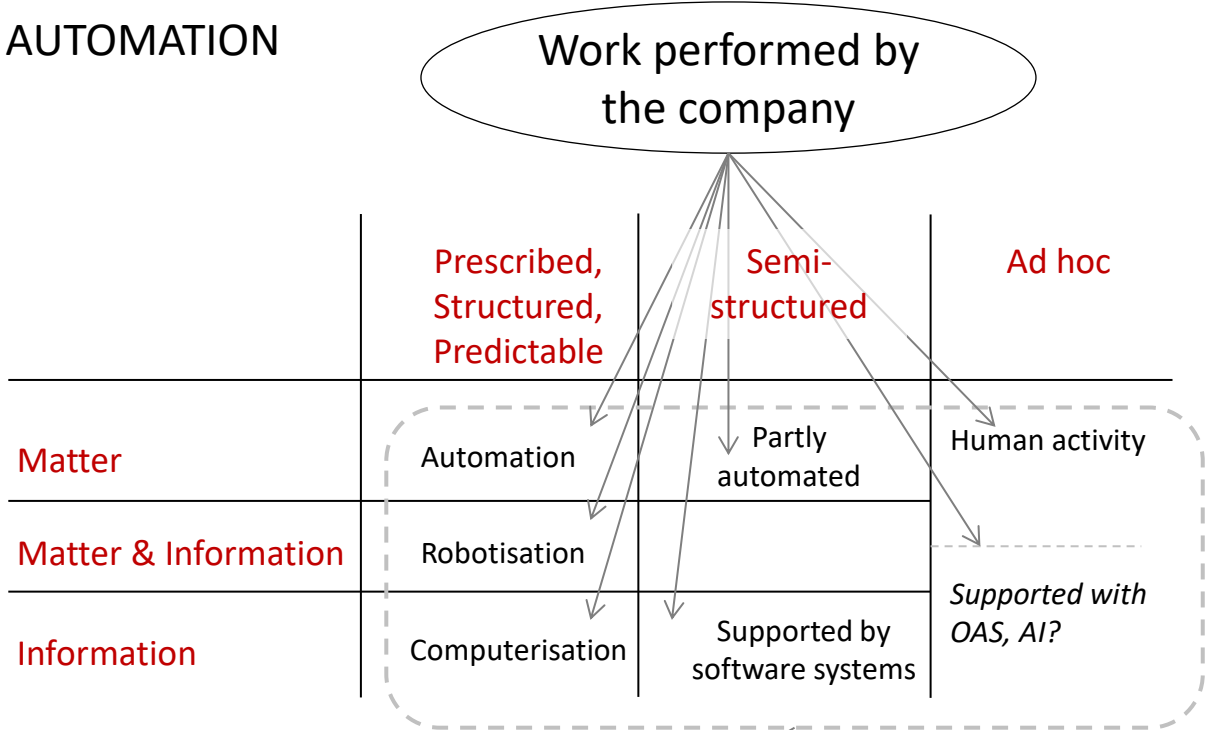
Automating the Three Components 30/10/2018



Three Executors

- have to fit into each other
- are intermingled
- have to collaborate seamlessly to achieve a same goals

AUTOMATION

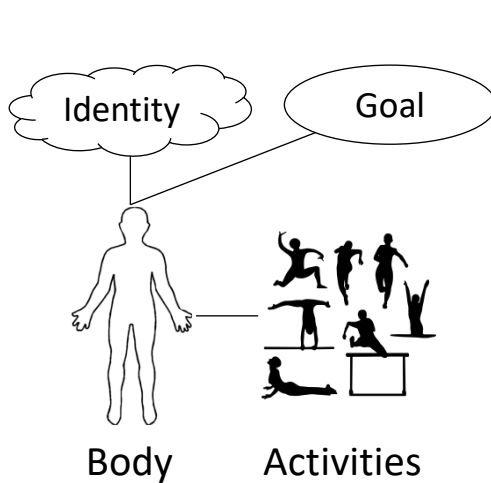


Ideally, forming a coherent single set of activities working towards some common goals.

Person \leftrightarrow Company (1)

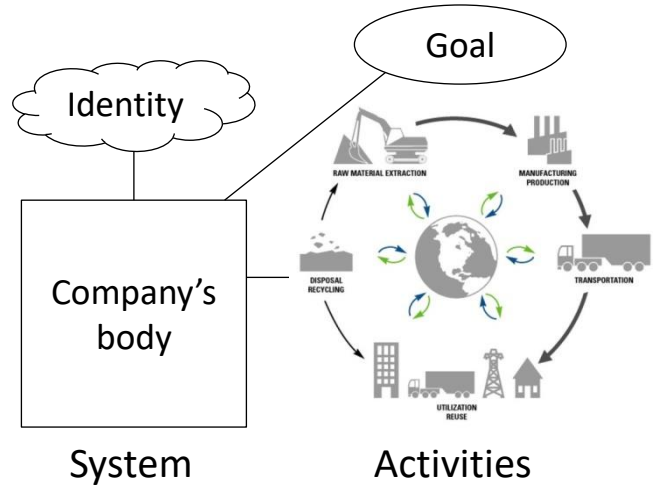
30/10/2018

PERSON



- Seeks to remain alive, well and happy;
- Personal development and self-realisation;
- to achieve the full potential and dreams;

COMPANY



Seeks to remain in existence;
to flourish and to prosper
(too often: to be profitable now)

Person ↔ Company (2)

30/10/2018

PERSON	COMPANY
Name, Identity	Name, identity, brand names
Personality, personal values, culture, belief system, thinking patterns, mental strength, norms, habits, autopilot, maturity	Company culture, values, leadership, CSR (Corporate Social Responsibility) , world view, belief system, way of thinking, habits and common practices, automatic responses, maturity, market position, ...
Relationships, social network	Relationships, collaborations, partnerships, network
Reputation	Reputation
Activities	Business activities, production
Skills, capabilities, limits	Skills, capabilities, limits
Body	Company as a system
Skin, guts	Boundaries: walls, gates, entries, firewalls, ...
Skeleton	Structural elements, like human organisations, organisation in business units, business functions, ...
Brain, guts	Management, guidance of operations
Nervous system	Information network
Blood vessels system	Matter and energy transportation network
Digestive system	Pre-processing of matter
Senses: vision, audition, taste, smell, touch (environment, input, ...)	Input, monitoring
Liver, kidneys, lungs	QC, Filters, cleaning mechanisms
Lungs, lymphatic system, sweat glands, excreta	Waste disposal system
Liver, Fat	Storage
Immune system	Recruitment (selection), Security
Healing mechanism	R&D, Engineering, Change Management
Pain, symptoms	Alarm systems, control agents, ...

Person \leftrightarrow Company (3)

30/10/2018

- A person has a private and public life.
- Organs perform each a function for the body (specialisation).
- Organs are placed inside a structure.
- Vital organs are better protected.
- Organs collaborate with each other.
- Organs are semi-autonomous. The conscious mind can focus on what matters.
- The body is hyper-efficient and has survival strategies built-in.
- The body and brain learn. We learn (also) from mistakes.
- The body and brain can be trained to acquire skills and capabilities. Not training and learning keeps the body weak.
- The body has the ability to heal itself.
- The body needs rest and sleep to restore.
- Prolonged period of heavy activity or stress is bad.
- If one organ fails, another organ may take over its role, be it partly or temporarily.
- Not treating a body and mind properly and abusing them leads to problems (over time).

Notes

This comparison provides lessons useful for designing systems.

Probably, other similarities can be found.

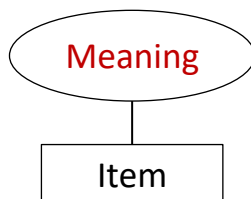


- INFORMATION -



Information

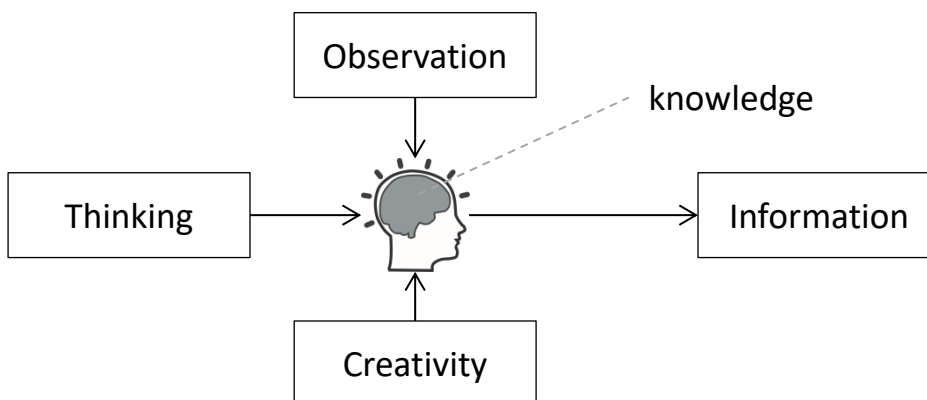
Information is the **meaning** (or knowledge) attached to something produced by humans.



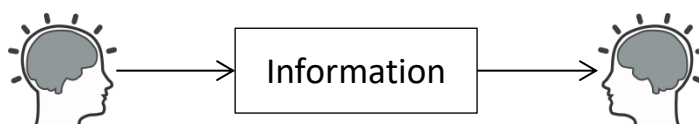
Example: A letter: Nor the letter, nor the words or letters (symbols) are the information. Information is the meaning attached to the letters and words.



Information is the **expression** of a knowledge.



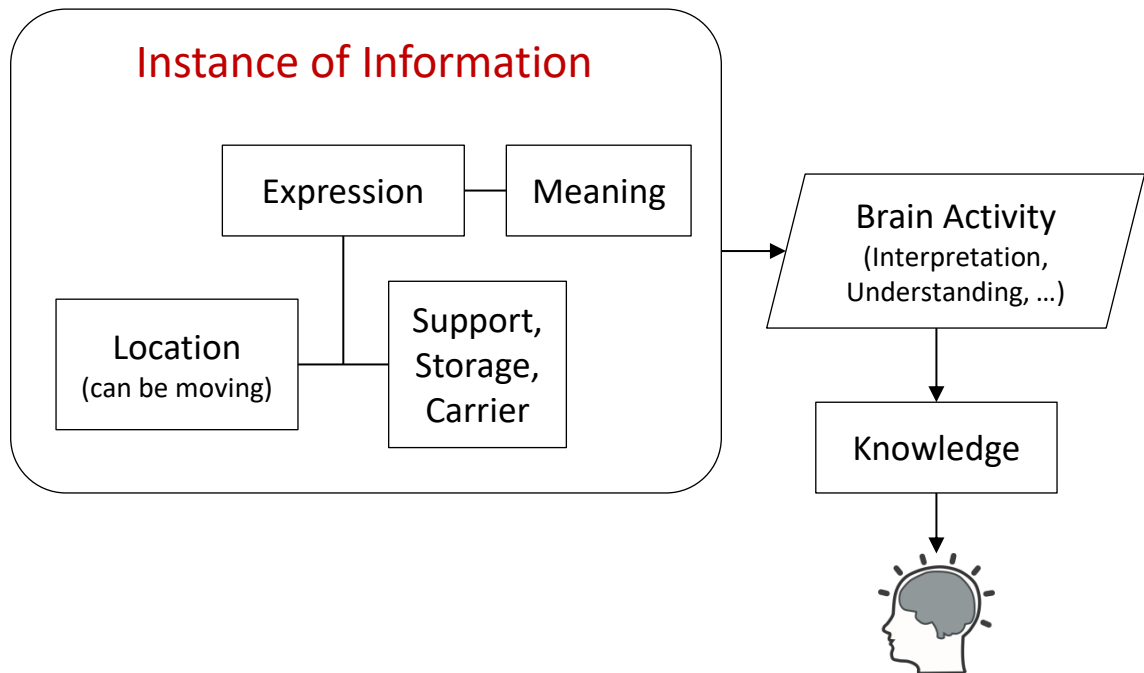
Message producing a **change** of a knowledge.



communication

*Captured information produces a change
(can be very small) in the knowledge*

- Information can be “produced” by man-made systems.
- The information doesn’t need to be picked up to be present (books, traffic signs, bottle with a message thrown in the sea, sign in the desert, ...)



Information IS the meaning. However, information can not exist without its expression.

Metaphor: A person is not simply the human body. But a person can not exist without the body.

An “**Information Element**” is a piece of information with an **atomic** meaning.

Examples: 1984, grey, Roger, cloud, ...

An information element is NOT a field. A field is a Software concept designed to contain an information element. An information element can be stored in a field.

Remark: For now, the term “information” is used for somewhat broader concepts also like an idea, a concept, a situation, ...

Fields, records, files, documents, messages, data files, databases are all concepts related to logical data storage.

Bits, nibble, bytes, words, blocks and pages are concepts representing units in the coding and the electromagnetic storage of this coded information.

Data - Information - Knowledge – Understanding - Wisdom

30/10/2018

DATA

- Representation of characteristics of objects or events
- Unprocessed information
- Encoded information without considering its meaning

INFORMATION

- Processed data so that it becomes useable
- Data with value
- Provided meaning to data, for example by putting into a context

KNOWLEDGE

Information that is in an individual's memory.

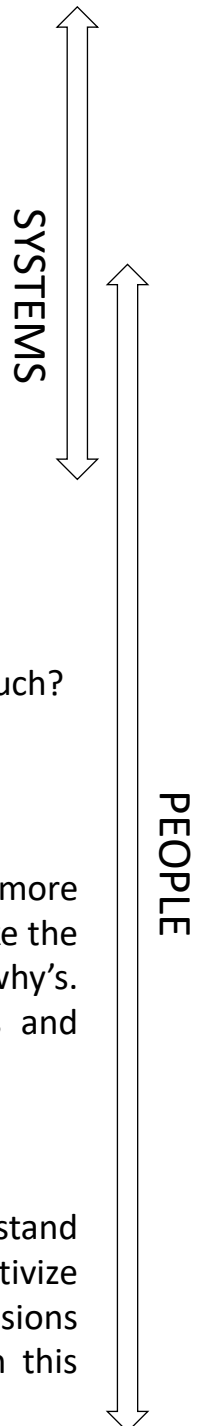
This answers questions like Who? What? When? Where? How much?
How-to? How long? (factual questions)

UNDERSTANDING

Knowledge turns into understanding when the knowledge is more extensive, when much more about the obvious facts is known, like the relations between the known information, the how's and the why's. Understanding is also knowledge about forces and dynamics and what's beyond the facts.

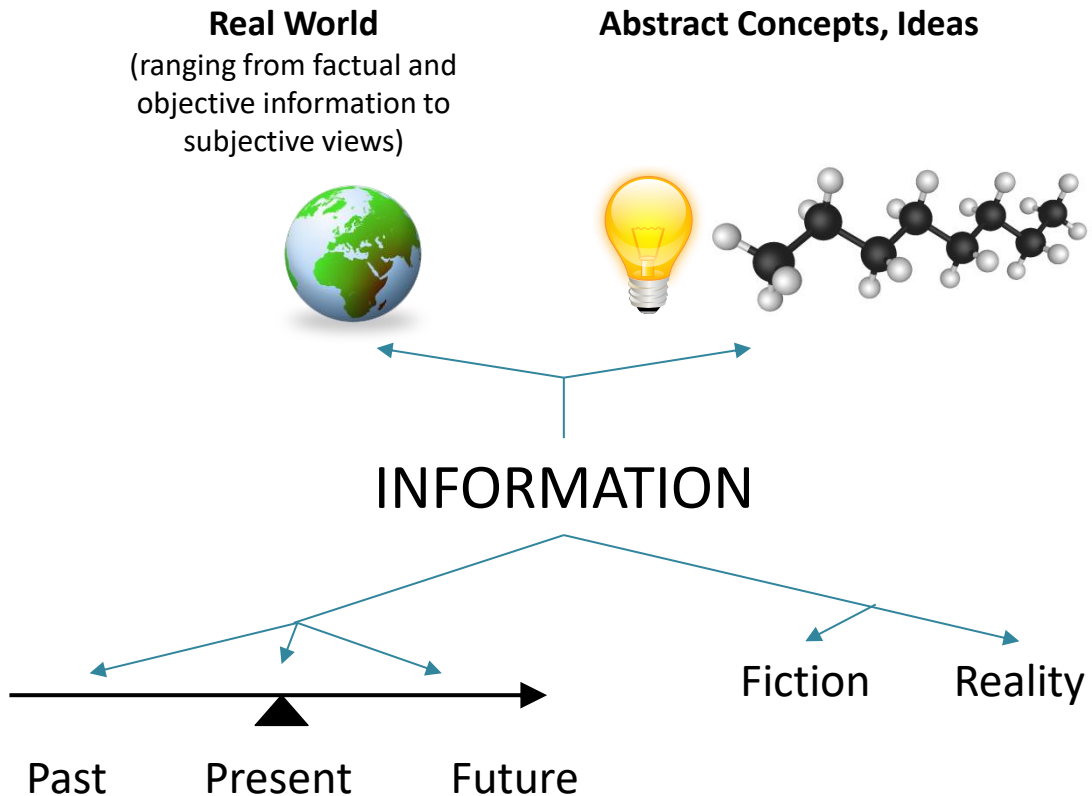
WISDOM

Ability to consider an issue holistically and objectively, to understand the various degrees of importance and the ability to relativize allowing to see the consequences of judgments, (potential) decisions and actions and being able to take sound decisions based on this insight.



Information is:

- the meaning assigned to and conveyed by symbols or sets of symbols.
- anything that provides an answer to a question.



INFORMATION
(study)

INFORMATION
(abstract concepts,
ideas, ...)

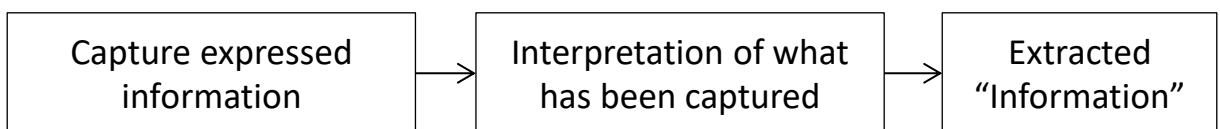


Creation of reality
(Product, Solution,
System, ...)

- ① Brown fox
- ② The colour of the fox is brown.
- ③ La couleur du renard est brun.



5x same “information”, but expressed in different forms



Information is about ‘meaning’. It is immaterial. It has no specific form. Information is an abstract resource describing from the real world to the most abstract and the most intangible, from truth to fiction.

Information is not a tangible, palpable, concrete, physical matter made of atoms. It is not even visible. Only an expression of the information is visible or perceptible.

1. OMNIPRESENT

Information is everywhere in the company.

2. CRITICAL

What would happen if suddenly ALL information of the company disappeared or was inaccessible?

3. INTANGIBLE / AMORPHOUS / INVISIBLE

Information is an abstract concept. Different expressions of information may reveal the same information, a very slightly difference or even the opposite. Information itself is invisible. Only its expression is visible. We can see books and servers or hard disks, but we may not see the letters and the bits and bytes. Even they are nothing more than the expression of information. Even what is NOT said or a silence can be information. Information is similar to knowledge.

4. MULTIMORPHIC

A same information can appear, can be expressed in many different forms and formats

5. MULTIPLE OCCURRENCES/INSTANCES

6. MULTILOCATION

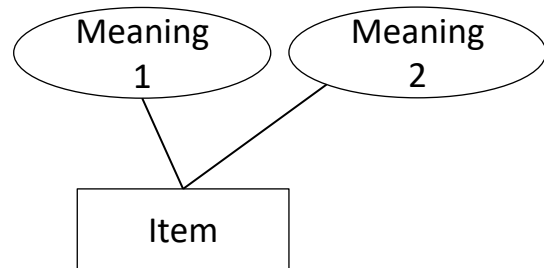
Since multiple occurrences (of the expression) is possible, the information has to be present at different places at the same time.

7. INTERPRETATION

Information has to be interpreted by the brain. People may interpret or turn a same piece of information (meaning) into different 'knowledges'. The value will be different.

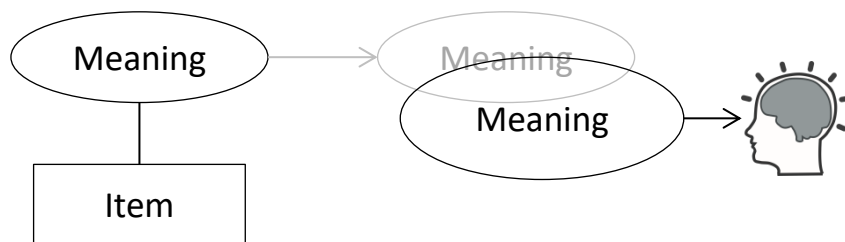
Decreasing information value

Ambiguity

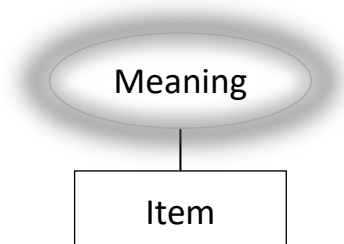


Example: term 'key': 1) piece of metal used to open or close a lock, 2) piece of keyboard 3) Code 4) Crucial

Interpretation

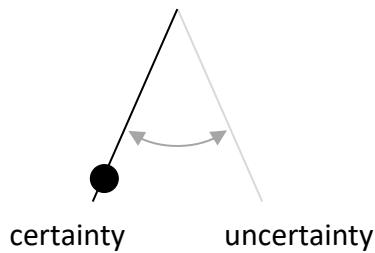


Vagueness

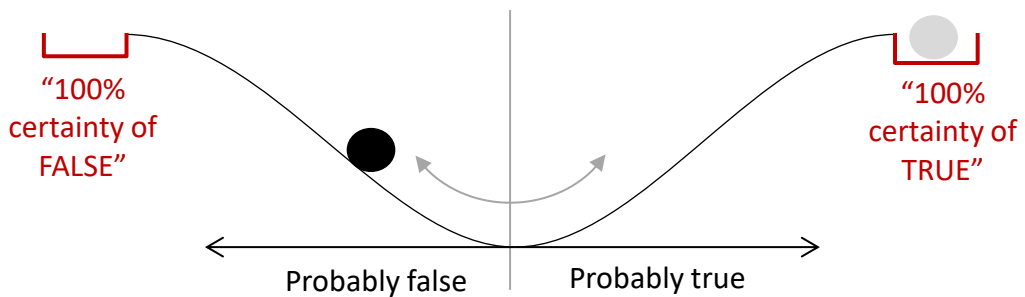


Probability - Uncertainty

30/10/2018

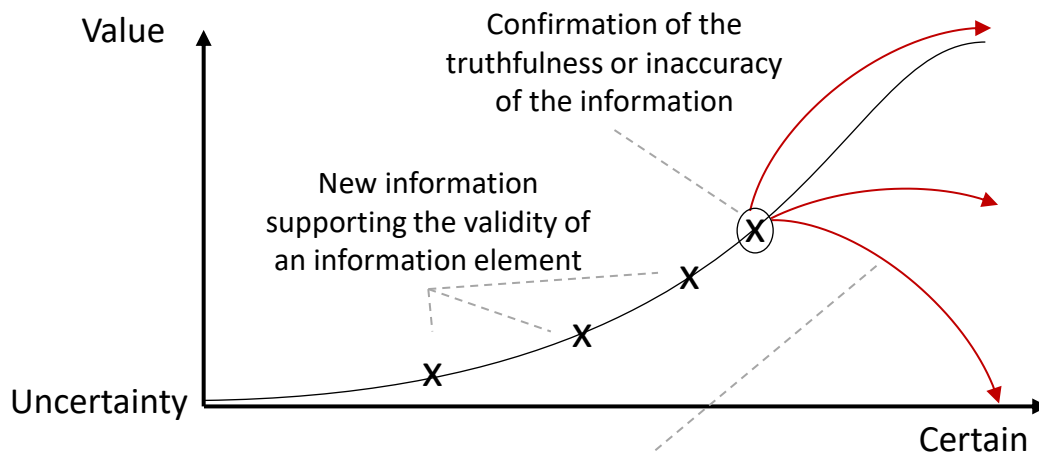


Information balances between uncertainty and certainty. As other information comes in, this movement may gradually move in one direction or it may oscillate until a 100% sure information confirms its truthfulness or falsehood.



Uncertain information is useable, but it is important to take the probability of being either true or false into account and to keep it in mind as long as the information is true.

An information element is (in)validated by other (ex. new) information elements. The "confirmation" itself is an information element.



A confirmation triggers a jump to the final value. Information value is not static. It evolves over time.

Human's Preferences

30/10/2018

0101

5

Yellow



“B4x3C”

“Yellow”

Brown

Brown

Beach of an island in a purple sea. An orange sunset and in the upper right corner 5 birds flying.



Picture

Video

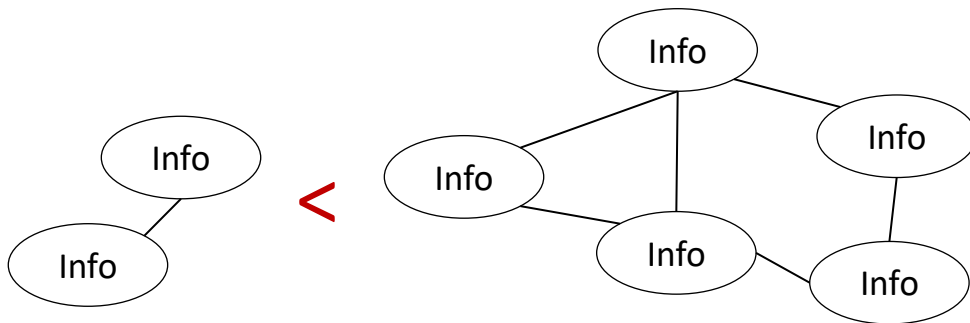


Visual, richer and
easier to process by
our brain

Value

Importance is subjective

- Personal interest
- Usefulness (practical usage)
- Already known or not



- Connecting information increases the value, insofar the information is true and not conflicting.
- Links are information in themselves.
- Linked information provide a richer picture.
- Linking information allows to verify information and to detect conflicting information.



What is context? Context is the set of data linked to considered data element. Context increases the meaning and the value to data.

Information Processing

30/10/2018

5

Number (integer)

5,0

Floating number

"5"

Text string

"Five"

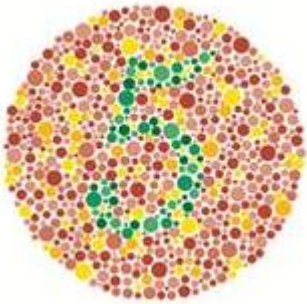
Text



Graphic



Graphics to be interpreted



EASY
to process

HARD
to process

An “**Information Entity**” is a set of naturally related information elements describing a same concept, object, aspect, ...



- PROCESS -

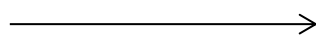
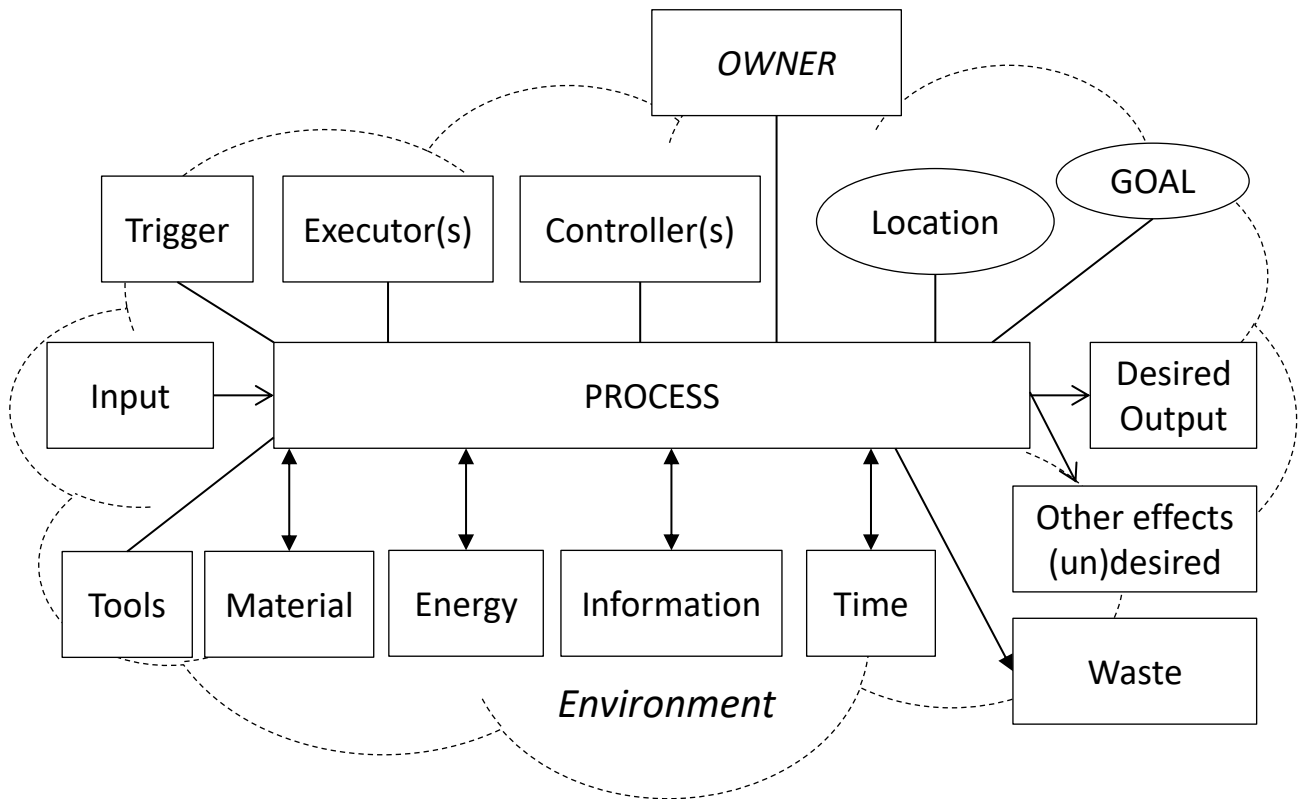


Examples of Processes

- Producing a car
- Selling a car
- Driving a car
- Executing a strategy
- Conducting a meeting
- Applying a Methodology
- Correcting a client's address in a database
- Making coffee
- Learning
- Taking a decision

Artefacts representing processes

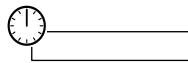
- Plans
- Methodologies
- Procedures
- Process models



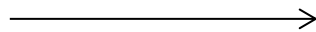
Linear process



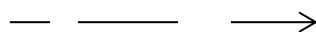
Circular process



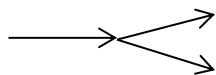
Recurrent processes



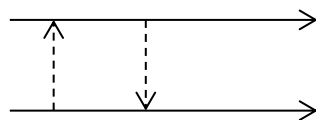
Continuous processes



Intermittent processes



Fork



Signals between processes

Notes

Not all elements are not always present.

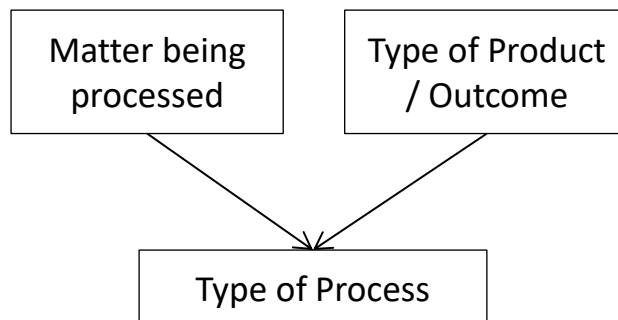
SEE "PROCESS ANALYSIS" for additional key-elements(!) for the Meta--Model

1. Goal
2. Value created
3. Trigger
4. Pre-requisites
5. Input
6. Output / Result
7. Steps
8. Test / Condition
9. Decision
10. Repetition / Loop
11. Discrete, Continuous
12. Linear, Divergent, Convergent
13. Transformation, Assembly
14. Size
15. Complexity
16. Degree of formalisation
17. Degree of structure
18. Phase
19. Gate, stage
20. Used resources
21. Domain of application
22. Required Competencies and Skills
23. Waste
24. Cost
25. Risks
26. Actors
27. Executor, performer
28. Monitors
29. Inspection / Control point
30. Measure point
31. Metrics, Performance, Indicators
32. Capacity
33. Process start
34. Process end
35. Duration of activity
36. Workload per resource
37. Throughput
38. Cycle time
39. Touch time
40. Takt time
41. Work in Progress
42. Work in Queue
43. Critical Path
44. Bottleneck
45. Variability
46. Criticality
47. Supplier
48. Source
49. User(s)

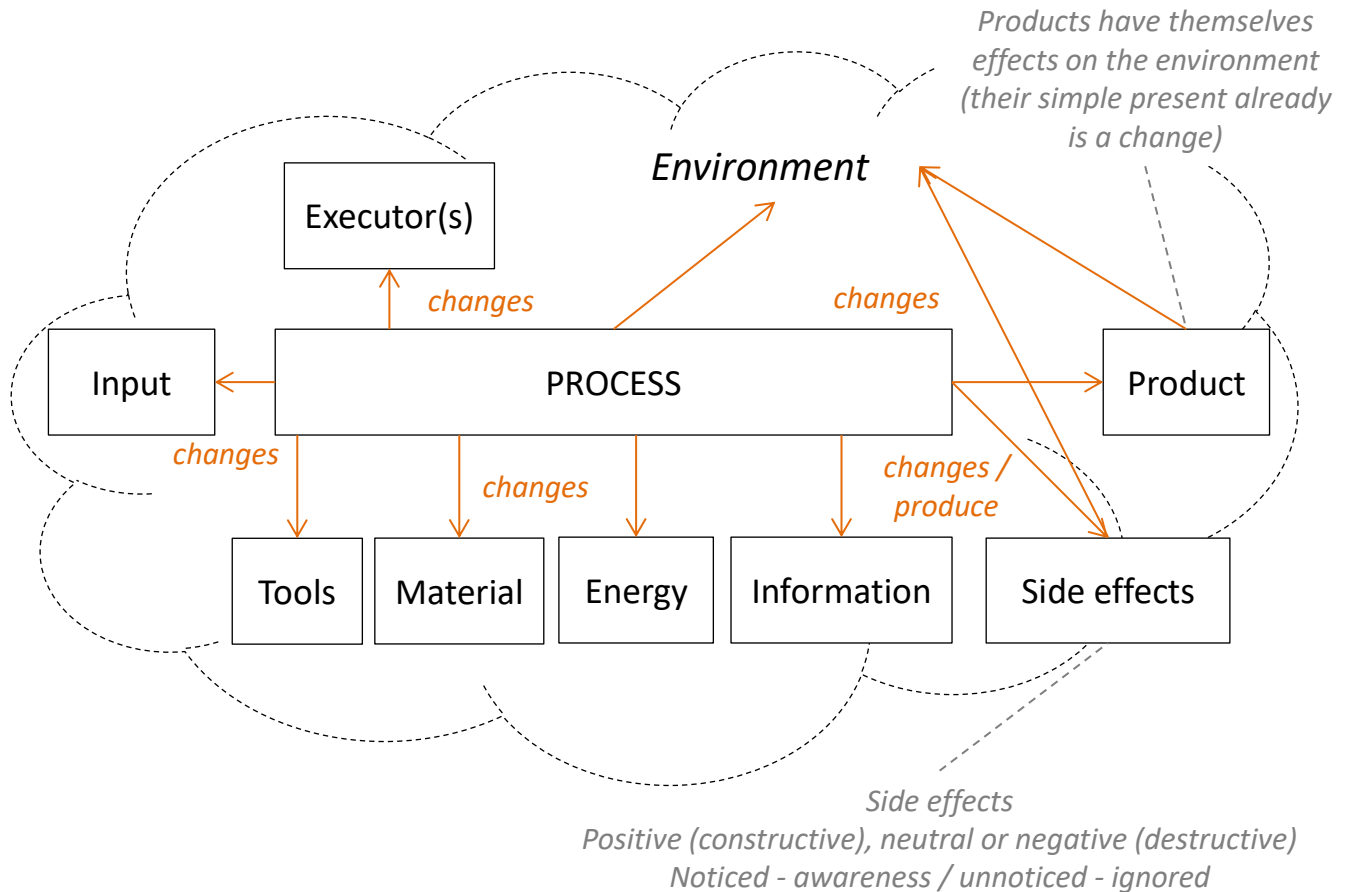
Types of Processes

30/10/2018

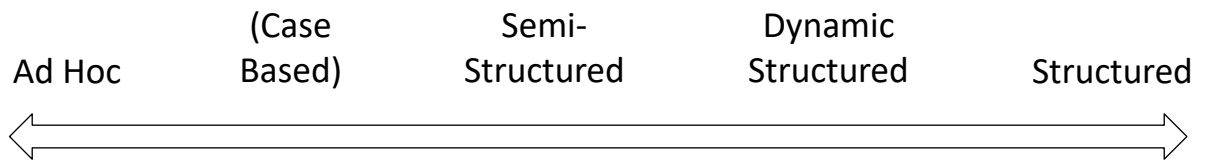
- Industrial process
- Chemical process
- Physical process
- Transport process
- Social process
- Psychological process
- Intellectual process
- Decision making process
- Engineering process
- Management process
- Business process
- Operational process
- Information process
- Financial process
- Communication process
- ...



What does a process change?



10/01/2020



Ad Hoc / Unstructured

- Goal: unknown or may vary a lot
- Phases: None
- Process: No defined process
- Unknowns: A lot of unknowns to be discovered during the process
- Decisions: all decisions are taken during the execution

Case-Based

- See semi-structured
- But case-based can be lesser structured and limited to, for example, an available set of methods.

Semi-Structured

- Goal: Goal or product is known but may vary somewhat
- Phases: Stages and/or phases can be discerned
- Process: Parts (fragments, sections) of the process are defined. These parts can be subject to adaptation. Some steps have to be decided during the execution.
- Unknowns: resolved in order to be able to start next phases
- Decisions: A lot of decisions about the execution and resources are taken during the execution.

Dynamic Structured

- Goal : Goal and product are known, but may still be adapted
- Process: A suitable process based on the knowns can be devised upfront, yet the process is still subject to changes during the execution
- Unknowns: Most unknowns have been resolved. Some are still to be discovered.
- Decisions: Each step is questioned during the execution and appropriate decisions about the course of action are taken.

Structured

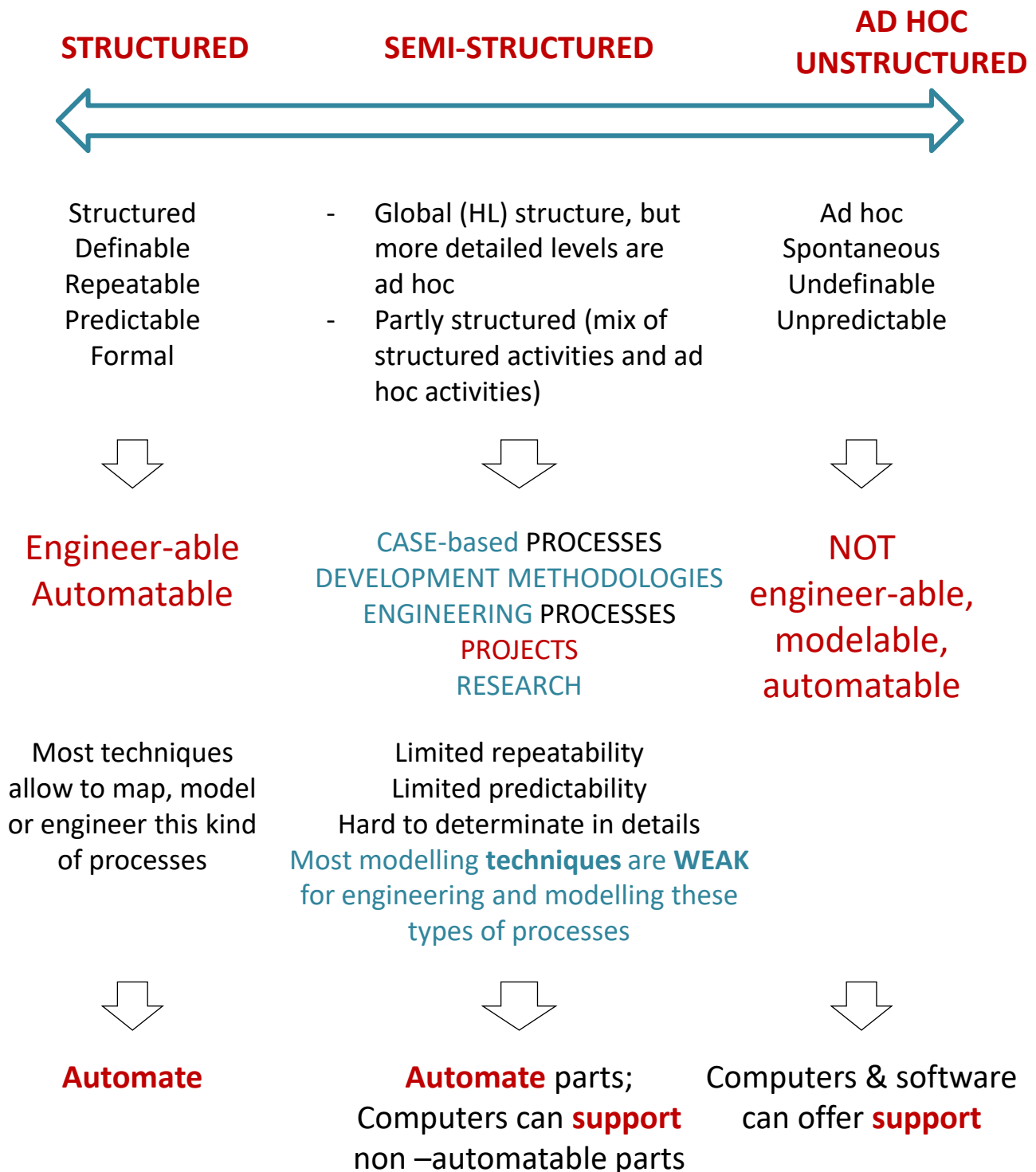
- Goal: goal or product are clear and immutable
- Process: Fully defined. It may include alternative paths, paths to deal with exceptions
- Decisions: no change in the prescribed process
- Unknowns: All unknowns defining the process have been resolved upfront.

Predefined processes can be executed as prescribed, without deeper knowledge (“follow the recipe”).

All other processes **NECESSITATE** a real **MASTERY**.

From Structured to Ad Hoc

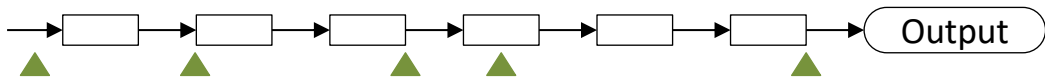
30/10/2018



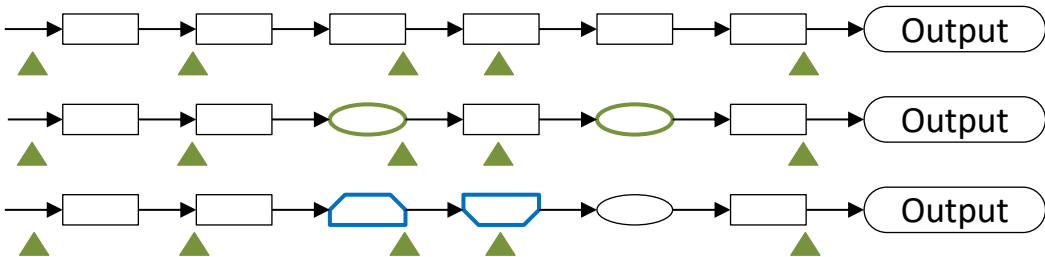
Note:

Not advisable to try to design a case-based process as if it was a structured process. This is ignoring the freedom of decision making of the executor and the freedom to adapt inherent to that type of process.

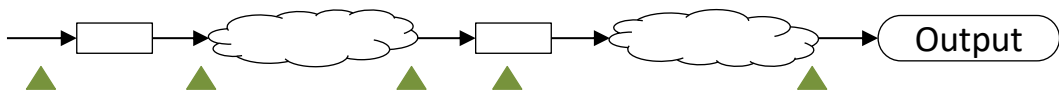
▲ : Control point, measuring point, performance indicator



The process is executed in many different ways



A process is executed in many different ways. It is as if there are different processes (or versions). But we can not compare measurements without knowing the differences between the processes.



If

- the process is undefined,
- ill or loosely defined,
- executed in variable ways
- or frequently changing

then

- it may be unclear what lead to the measurement. The true meaning of the measurement may become vague.
- It may be unclear how this measurement fits into the whole.
- Measurements may not be comparable.

Formalising processes help to study and to understand them, and their measures. Not all processes can be formalised, some only partially or to some degree.

THREE DISTINCT PERSPECTIVES

1. THEORETICAL EXECUTION MODEL

Presents all the steps and decisions and all the possible executions paths.

2. INTERNAL ORGANISATION

Organisation of the process steps and of the logic

- Layers
- Hierarchy
- Libraries
- ...

3. EXECUTION

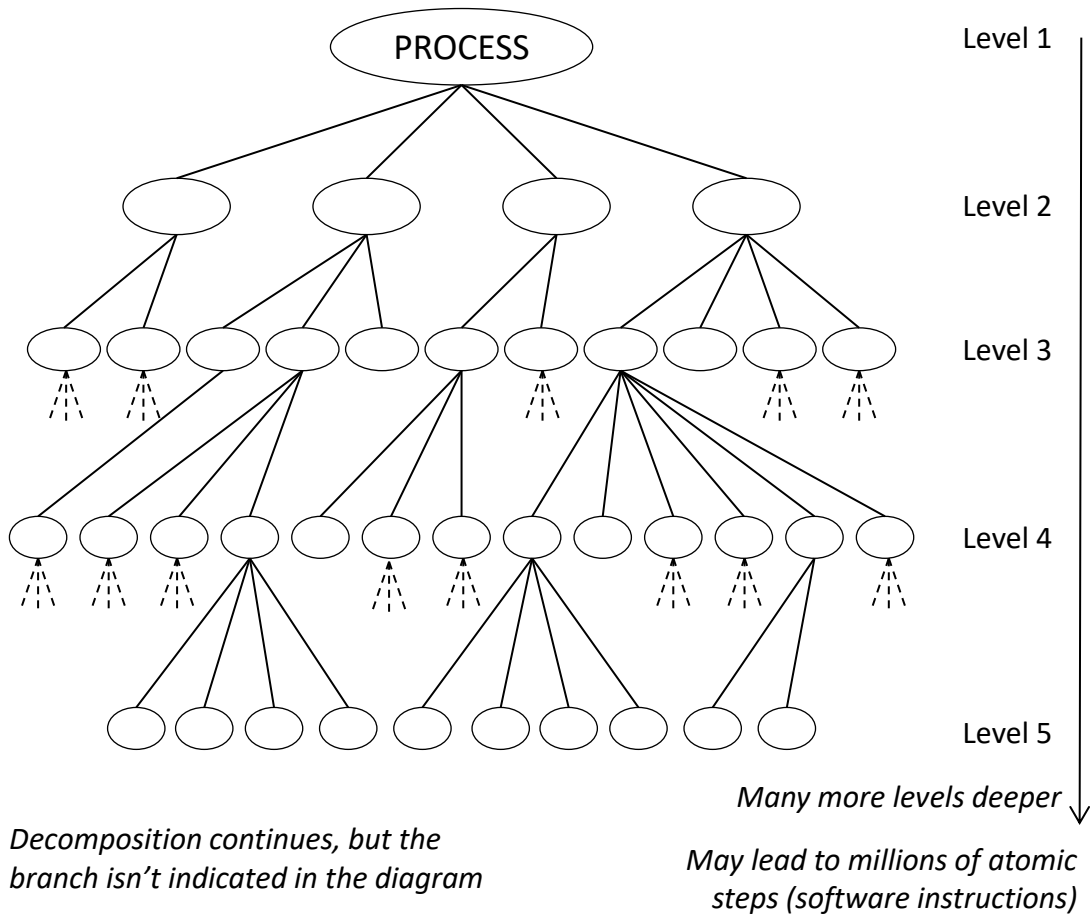
- How the process can, is or has been executed
- Specific instance of the theoretical model

ENGINEERS MUST DETERMINE (MODEL) THESE PERSPECTIVES

VERTICAL VIEW

MAIN IDEA

- *Dividing the process into steps (top-down)*
- *Defining the grouping of the steps*



A sub-process is a process which is being a part of a higher process.

A process can have 0, 1 or more sub-processes.

Notes:

Conditions, repetition can be indicated at the nodes

In software applications, a process can be 5, 10, 20, ... 50 or more layers deep.

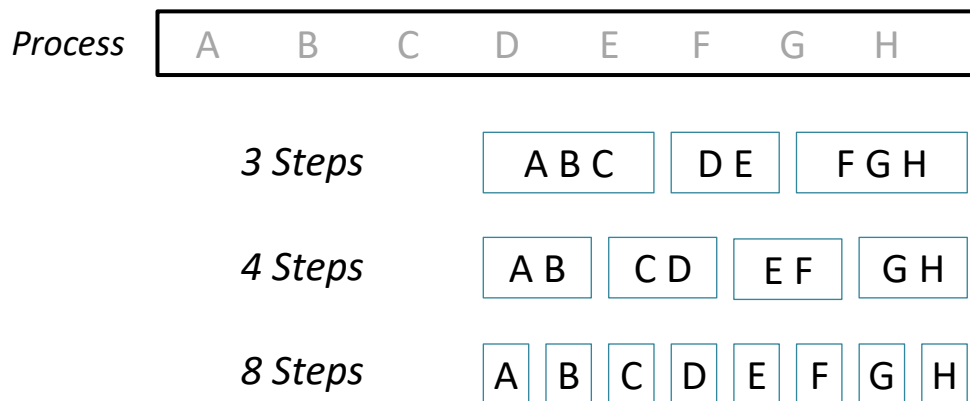
Hierarchical Decomposition

30/10/2018



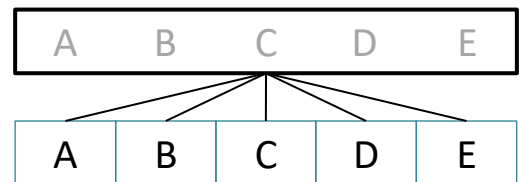
- In how many ways can you divide the rope (by marking the segments, not by cutting it) ?
- Work may differ → different steps. Where exactly stops one step and begins the next one?
- Many ways of grouping steps (defining “phases”)

Different ways to divide a same process into steps

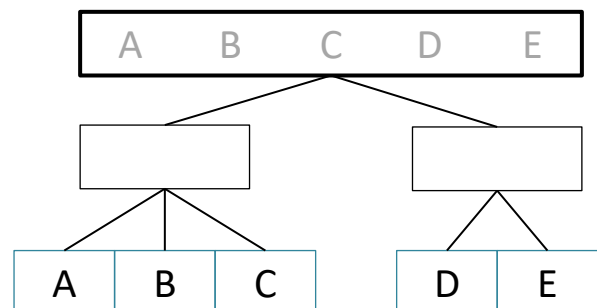


Different organisation of a process in layers

2 Layers



3 Layers

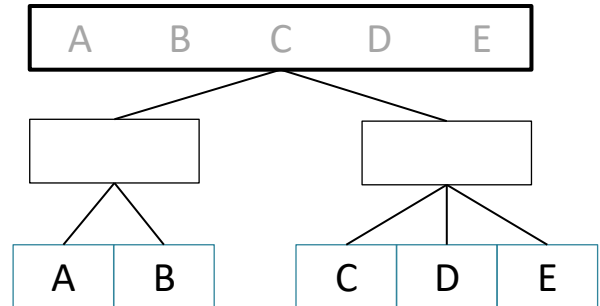


Hierarchical Decomposition

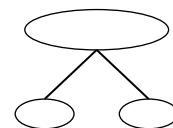
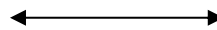
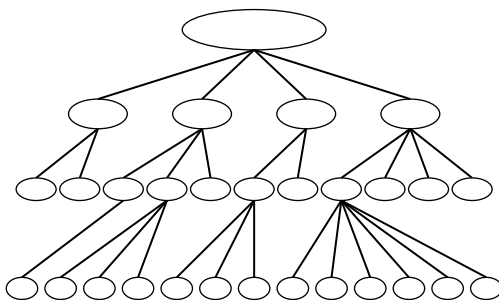
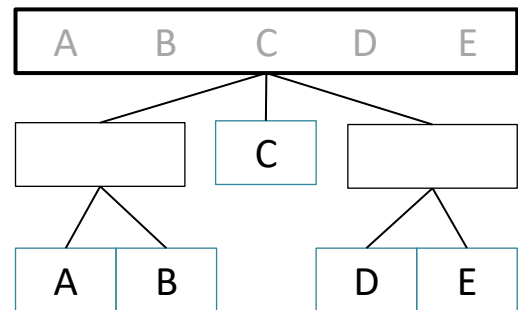
30/10/2018

Many different possibilities are possible

Or tasks are grouped differently in the phases.



Or in three phases.



Can be the exact same process

Notes:

Many ways to represent and organise a same process without changing the execution → affects clarity, maintenance, reuse and flexibility

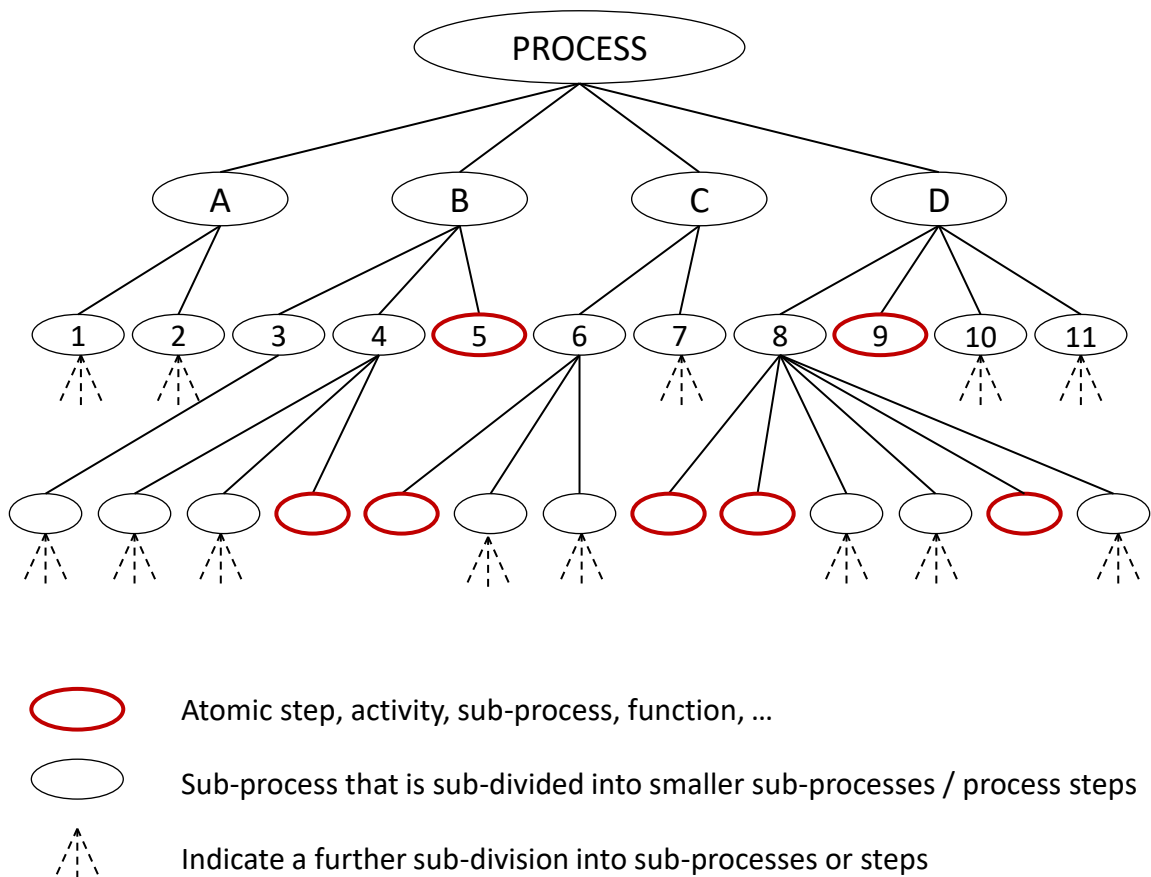
Need to choose the most logical and practical definition of steps, the grouping of steps and number of layers.

Hierarchical Decomposition

30/10/2018

IMPORTANT

This represents a **specific existing or designed** (*) process. All the logic is located in the atomic steps (lowest step of each branch). Atomic steps can be located at different levels (in different layers), and not necessarily all in the lowest layer. The “hierarchy” above is mental construction, normally based on a certain logic defined by a functional grouping combined with a level of detail.



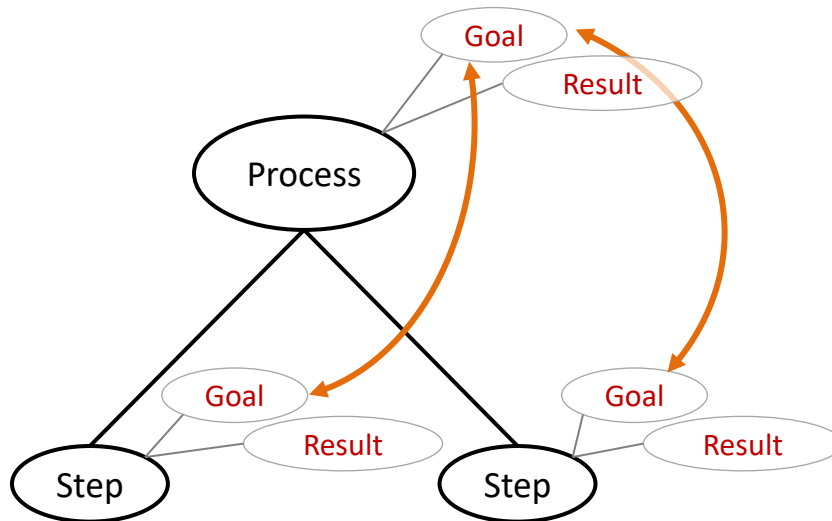
Notes:

(*) A different case the Analysts face is that of process that is still unknown, inexistent and to be engineered.

The “layers” of bubbles tell the degree of decomposition. They don’t represent a level of detail.

PROCESS GOALS

Alignment of the sub-processes and process steps



Goals of sub-process have to contribute, directly or indirectly to the goal (purpose) of the higher process.

The result of the execution of the process should either be useful further in the process or a part of the final result, else it's waste.

Notes

The “goal” is often not just the obvious. It is often not simply the product, problem or business goal. It is actually so much more than what is expressed. Therefore the goal must be identified with the greatest care.

Example:

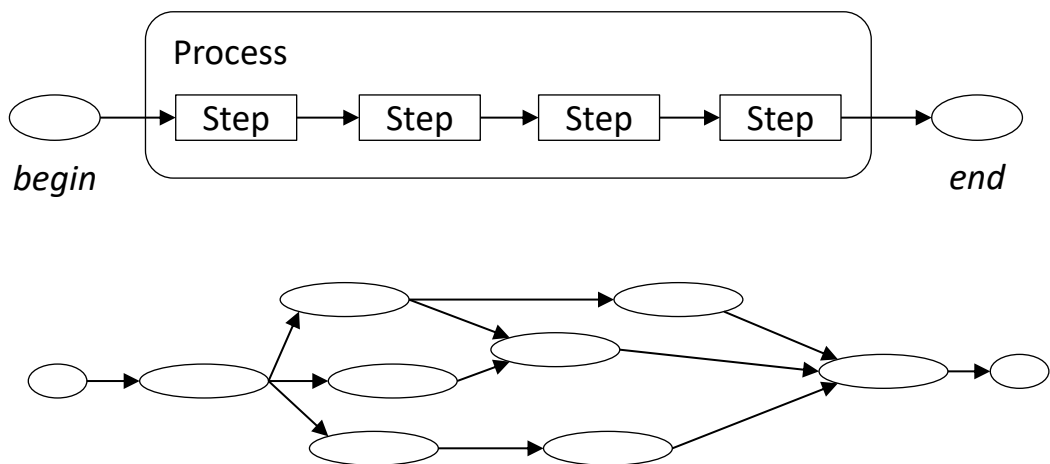
A marathon: Goal: Winning the marathon

Is the goal reached when winning the marathon and dying after the finish line?

HORIZONTAL VIEW – Process Flow – Work Flow

MAIN IDEA

Representing the successive steps of (approximately) same level of detail

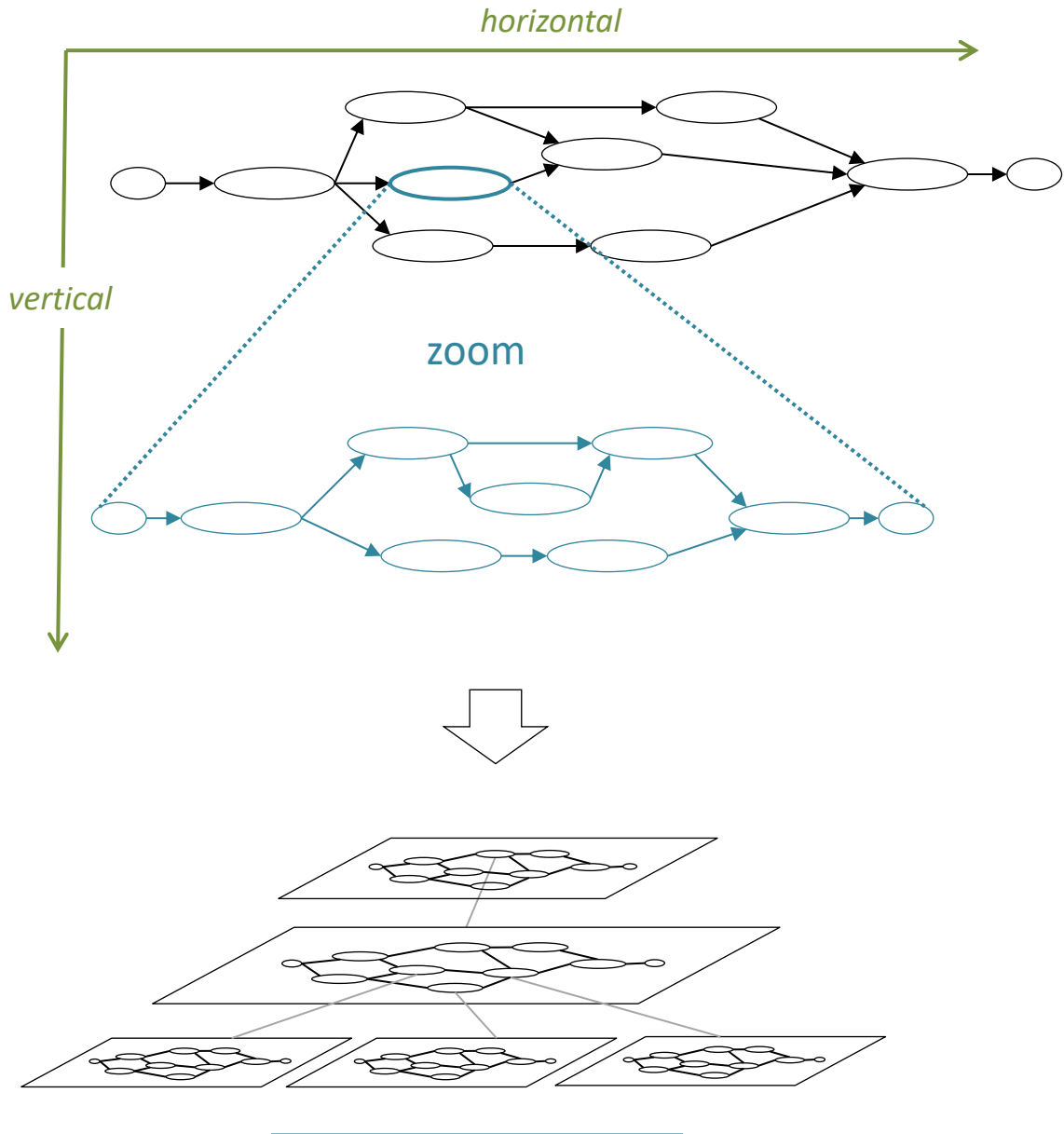


Ideal for representing Business Processes

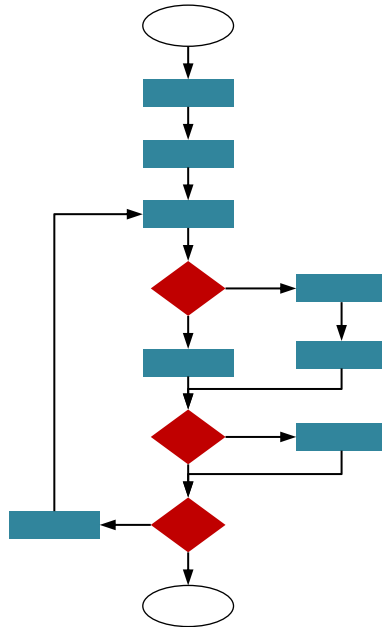
Examples

- Business Model
- SIPOC
- Value Stream Network
- Value Chain
- Data Flows
- Work Flows
- Activity Diagrams
- Swimlanes

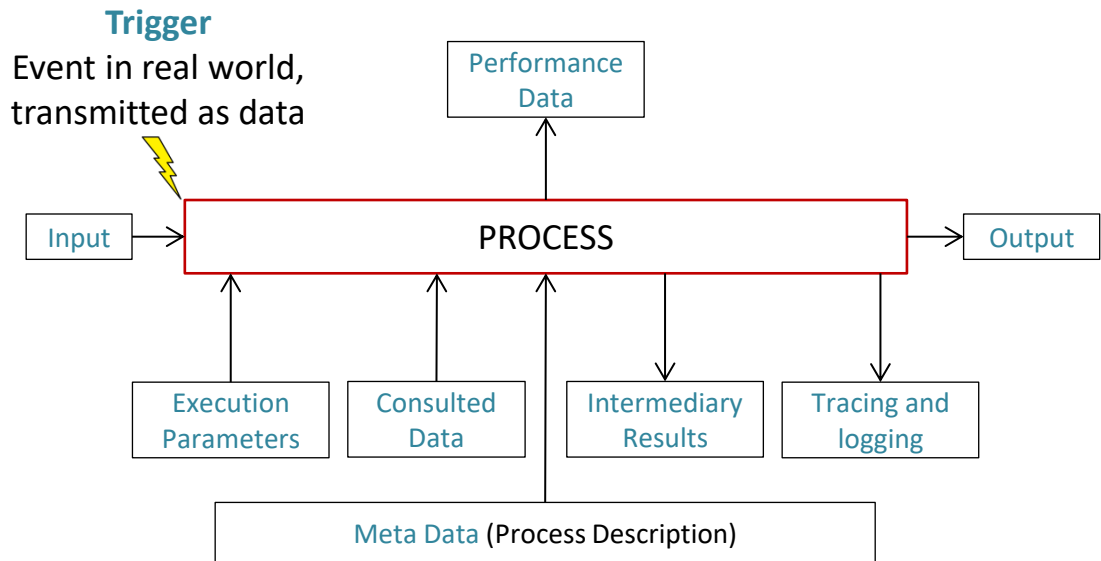
Linking horizontal views with more detailed horizontal views



DETAILED VIEW – Flow Chart

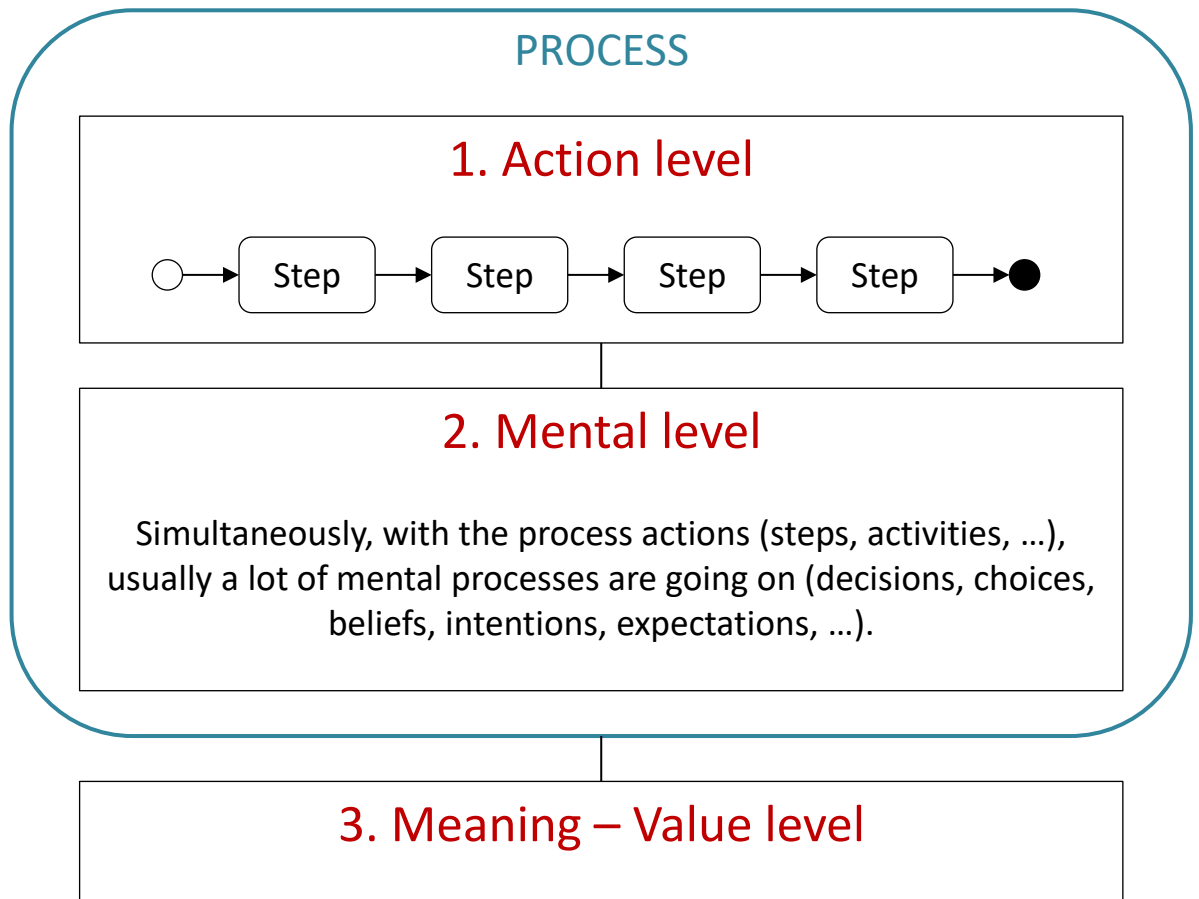


- Shows conditions, decisions, repetitions, branches, ...
- Shows main stream and exceptions
- Very easy to read and to understand
- Can represent any level of a process
- Ideal for more detailed levels
- Can be combined with swim-lanes to illustrate who executes steps.



Identifying existing and required processes

- What **outputs** need to be created?
- To what **events** does the company need to react?
- What may happen in the **real world** ?
- What are the **existing** processes?
- What are the **inputs**, **resources** and **prerequisites** for these processes? Other processes are required to get these inputs, to create the prerequisites and to obtain the resources. What are these processes?
- What processes are needed or possible to keep the information up-to-date?
- State transitions in information entities also points to events and processes to trigger and perform these transitions.



EXAMPLE

WAKING-UP process

Action-Level: alarm clock rings, waking up, brush teeth, put clothes on, prepare breakfast, have breakfast, ...

Mental level: Why did you or didn't you push the snooze button? Did you take the decision to brush your teeth? How did you choose your clothes? To make the choice, did you think on the activities of today? Why? Did you check the weather outside or the weather forecast? What else were you thinking? How did you choose your breakfast? Did you decide that morning about the order of actions? And so on.

Meaning / Value level: It is important for you to feel good, to look good, to impress and not to be hungry later in the day. And so on.

Goal - Result	
Concrete Process Dynamics	<p>or activity, loops, the inputs, outputs, timing, messages, ... (level of UML, BPMN, Flow charts, ...)</p>
Meaning & role of outcome	<ul style="list-style-type: none"> • What is the result of the step or of the whole process? What is it worth? • Role of this result / outcome? • Value of this result / outcome?
Meaning of the step	<ul style="list-style-type: none"> • Meaning, sense of the step or of the whole process? • Why is the step, action, ..., process executed? Why executed/designed like this?
Intellectual perspective of the process step	<ul style="list-style-type: none"> • What decisions are made defining the process? • What is the thinking driving the process? • What is the intention during each step (or whole process)? • What are the expectations of the steps (or whole process)?
Beliefs	<ul style="list-style-type: none"> • What perspective(s) is/are at play ? • What is the belief system? • What assumptions are made?

Depth

Fully understanding of the process

Notes:

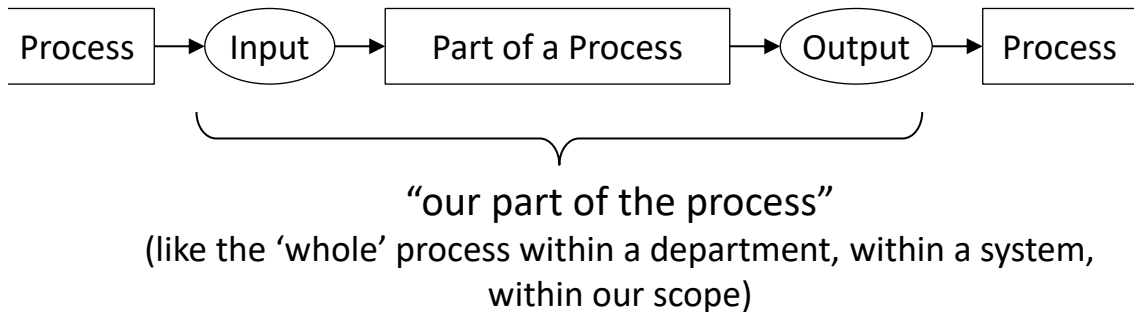
'Step' can be a phase, an action, an activity, a process, a function, a feature, ... a group or a mix of these.

Models of processes can be drawn by looking only to the actions, the inputs and outputs. But it won't get further than the existing processes or "order taking". A process can't be designed or properly evaluated without a consideration of the deeper levels. The deeper levels bring true and essential and even critical understanding of the deeper sense of the process. A **methodology** is, in essence, also a process which without deeper understanding can't be applied correctly, let alone adapted to suit the project and to create excellent results.

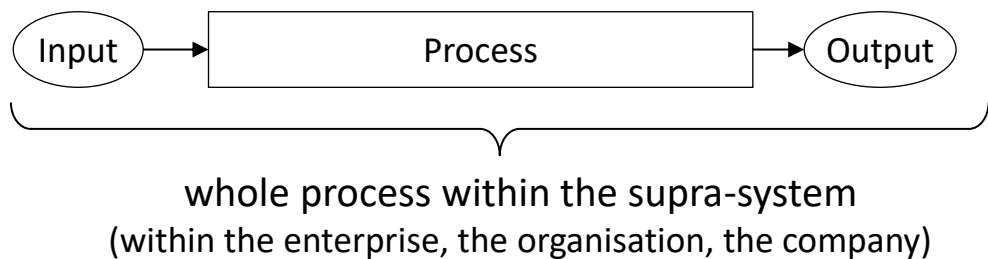
Scope of Analysed Process

30/10/2018

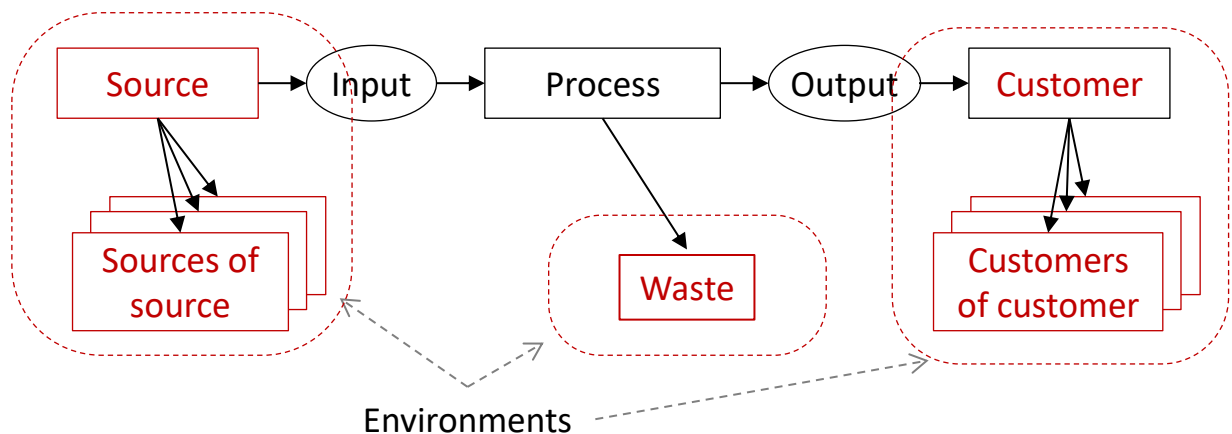
Limited Scope



Whole Scope



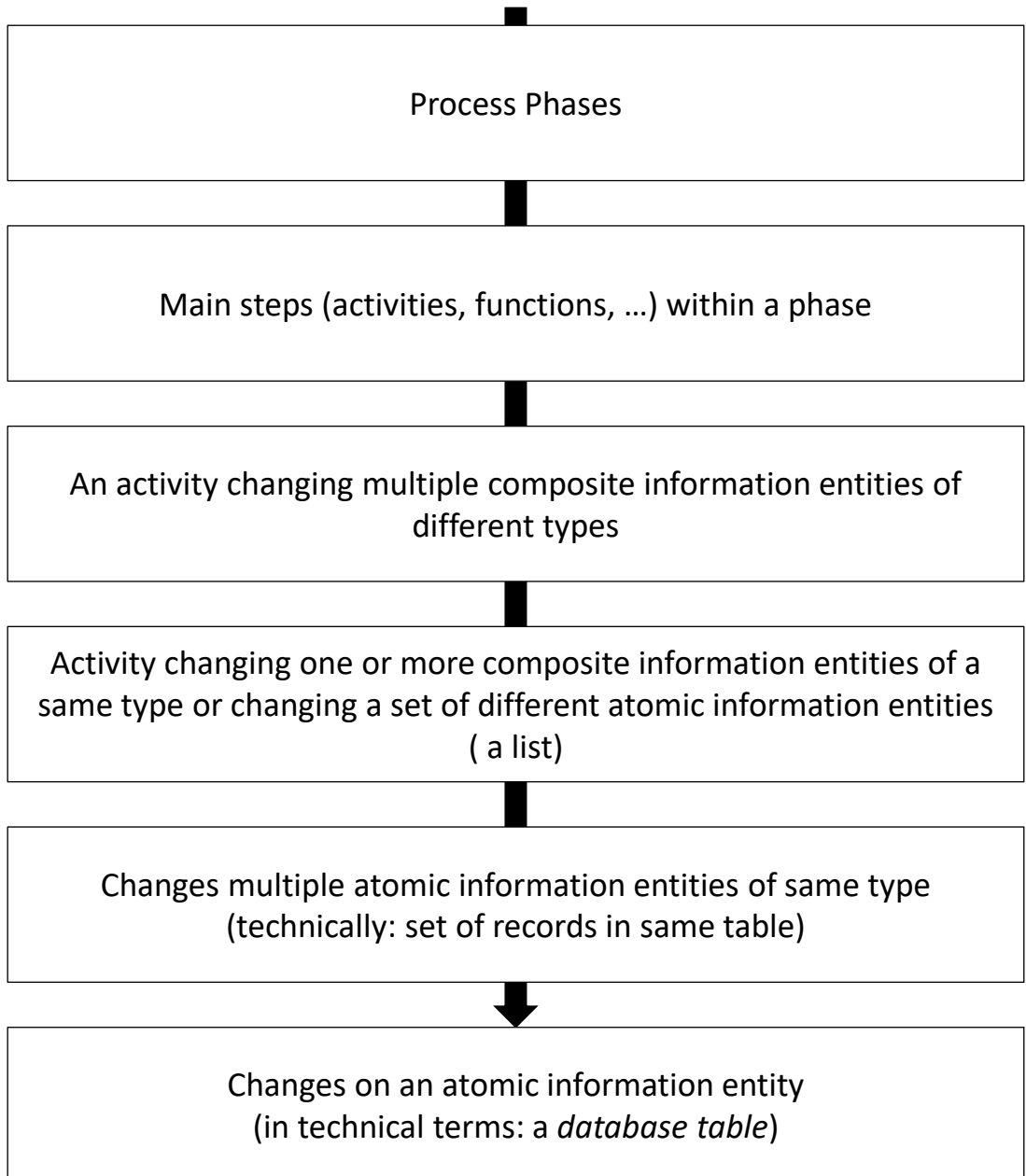
Extended Scope



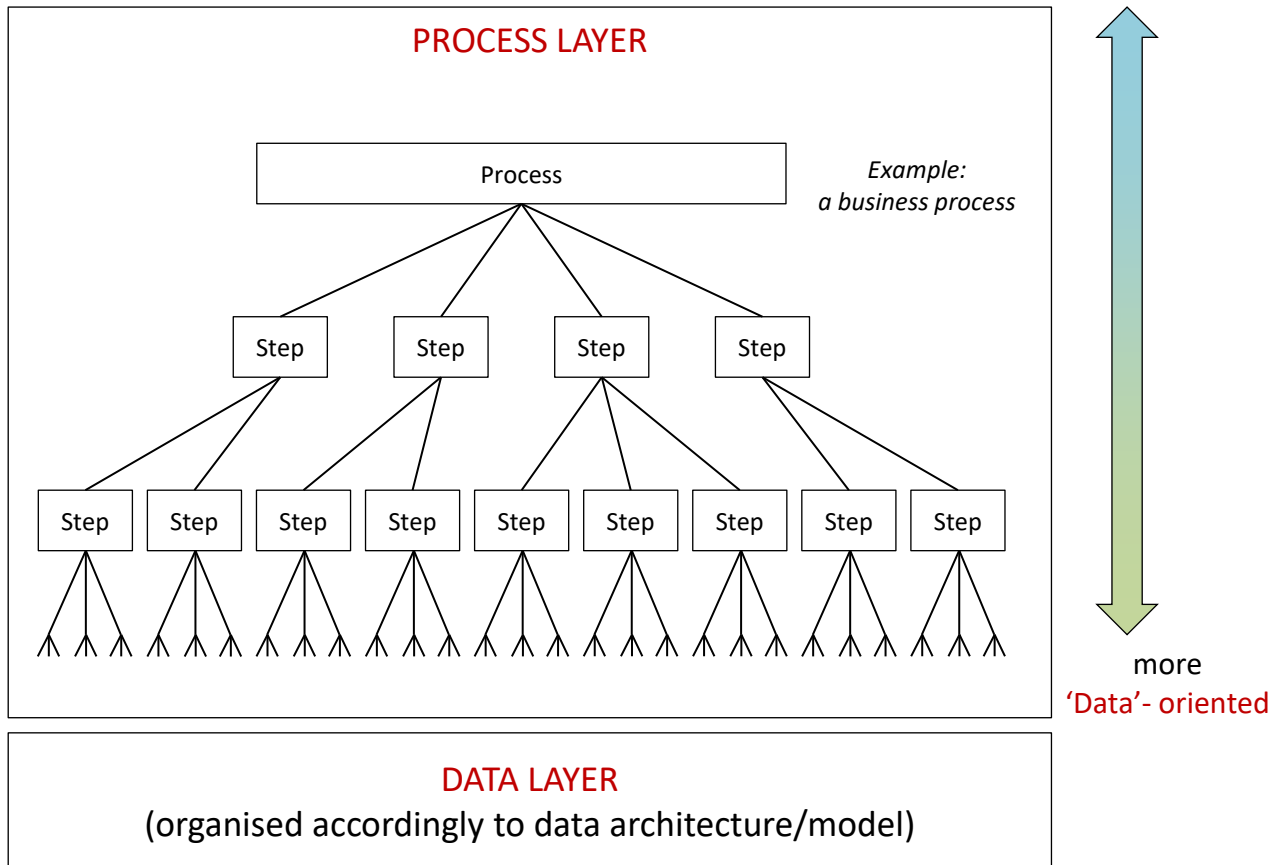
- The sources, the sources of the sources and their sources
- The customers, their customers and their customers
- the waste
- triggers, (hidden) consequences, side effects, ...
- and the behaviours, cycles, relations and environments of all these elements

can be analysed as well

Example: SIPOC-model in management



Structure of processes is more 'Process'-oriented



Notes:

High-level horizontal cross-border processes shows the process in its entirety, often/mainly regardless of the underlying data architecture.

At a deeper layer, we may find process steps dealing with closely related data entities. Even deeper we may find process steps dealing with lists of same data entities and finally steps dealing only with a single data entity.

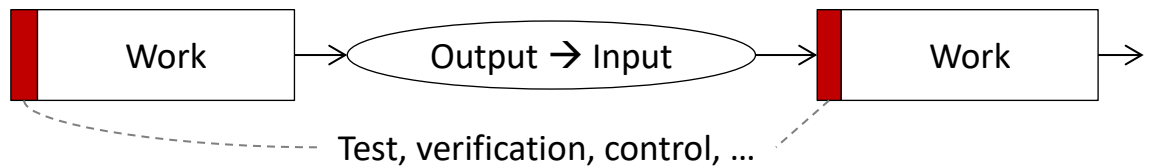
The lower levels of process steps can be better organised accordingly to the data architecture.

Representation: It presents one view: the decomposition of a process in its sub-processes, steps, sub-steps, These steps can be implemented as services, performed by agents or organised in libraries. They can be executed iteratively or form a network structure. These are just other possible views of a process, its execution or its organisation.

GIGO is not just a phenomenon that happens or caused by bad input. It is a **process design issue**.

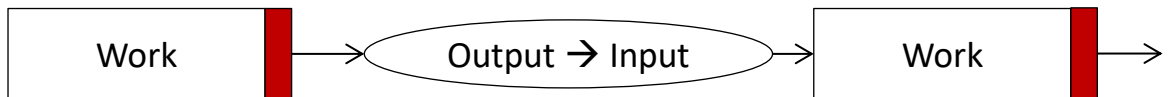
It is caused by processes that don't check their inputs and outputs and are not able to deal with wrong input.

Input Control



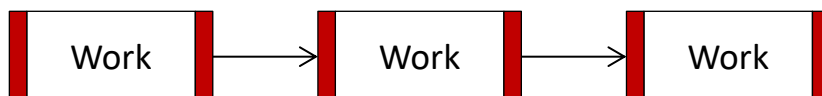
Question: Is the input fine?

Output Control



Question: Is the output, the work performed by the process, fine?

I/O Control



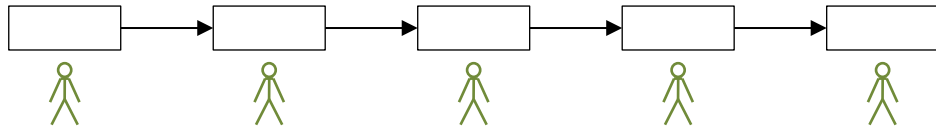
Question: Are the input and output of the process fine?

Control of output of previous step \approx / $=$ control of input \rightarrow control is duplicated and 2 same or similar control performed.

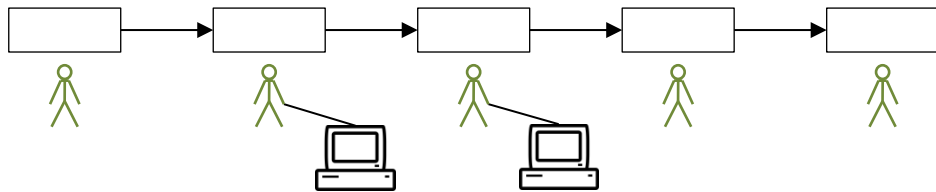
Continuous integrated testing



Process executed by people (manual process)

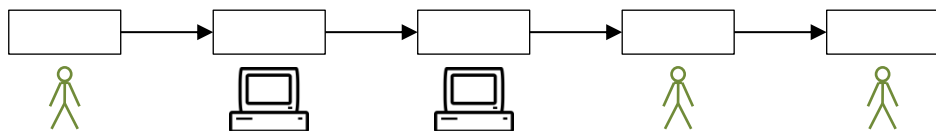


Process executed by people with support of software systems

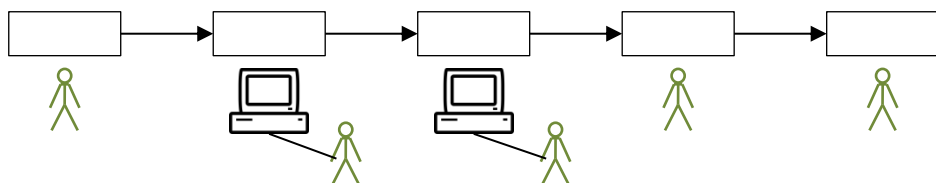


Computers do not execute the process. They are used to increase effectivity and efficiency, like for gathering information to take better decisions.

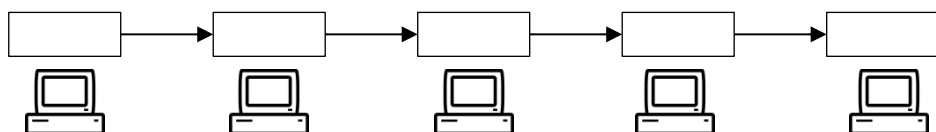
Partly automated process



Partly automated process with support of people on automated parts



Fully automated process



- Assigning an owner
 - Processes without owner tend to deteriorate.
 - Owner must understand process engineering, monitor the application of the process and keep the process efficient and effective.
 - Not an honorific title
- Organise processes in a logical and practical manner
- Define processes for reuse (modularity)
- Allow processes to be triggered, put on hold, interrupted, ...
- Allow processes to be traced and monitored
- Configurability of processes
- Record process metrics (key performance indicators, and other performance indicators)

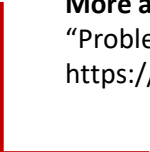


- PROBLEM SOLVING -

More about problem solving:

“Problem Solving – Concepts and Approach for Systems and Strategies”

<https://goo.gl/IgxPw0>



A **PROBLEM** is a difficulty to be resolved, an intricate question to be answered, a difficulty to understand or accept. It indicates the existence of a cause, an obstacle or a difficulty.

Why solving problems?

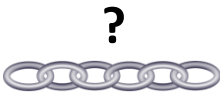
Some problems may:

- Go away by themselves
- Be solved by someone else
- Remain the same
- Amplify
- Change
- Create one or more negative consequences

Solving problems is a way to progress.

Advanced Problem Solving skills help to avoid problems, solve them early and more easily. This limits their negative impact.

**The Analyst is first of all
a “Problem Solver”**

Problem  (In-)Ability

No Problem	Anything that is OK or that can be solved easily
A Problem	Anything we are confident we can solve. But it will require some dedicated effort.
Big Problem	Anything we don't know yet how to solve. But with a big effort, we believe, we might still be able to solve it.
Very Big Problem	We are unable to solve it.

Each problem tests and trains our ABILITIES

“Problem” = personal & subjective concept

A lesser competent person is more likely to create more problems and to have more problems.

A smart and competent person will have more challenging problems, which may require more effort and time or a higher intelligence to solve them.

Notes:

- A *problem* is a subjective perception linked to our own ability to understand, conceive, build, adapt, change, improve, perform and action, get a result or reach a goal.
- It is also linked to our (im)possibilities, to various barriers and limitations existing in the situation and to the importance and criticality of the problem

A **SOLUTION** is only a solution if it solves the problem (the cause, the gap, the difficulty) without creating (too many) new (significant) problems. Anything else is a **figment**.

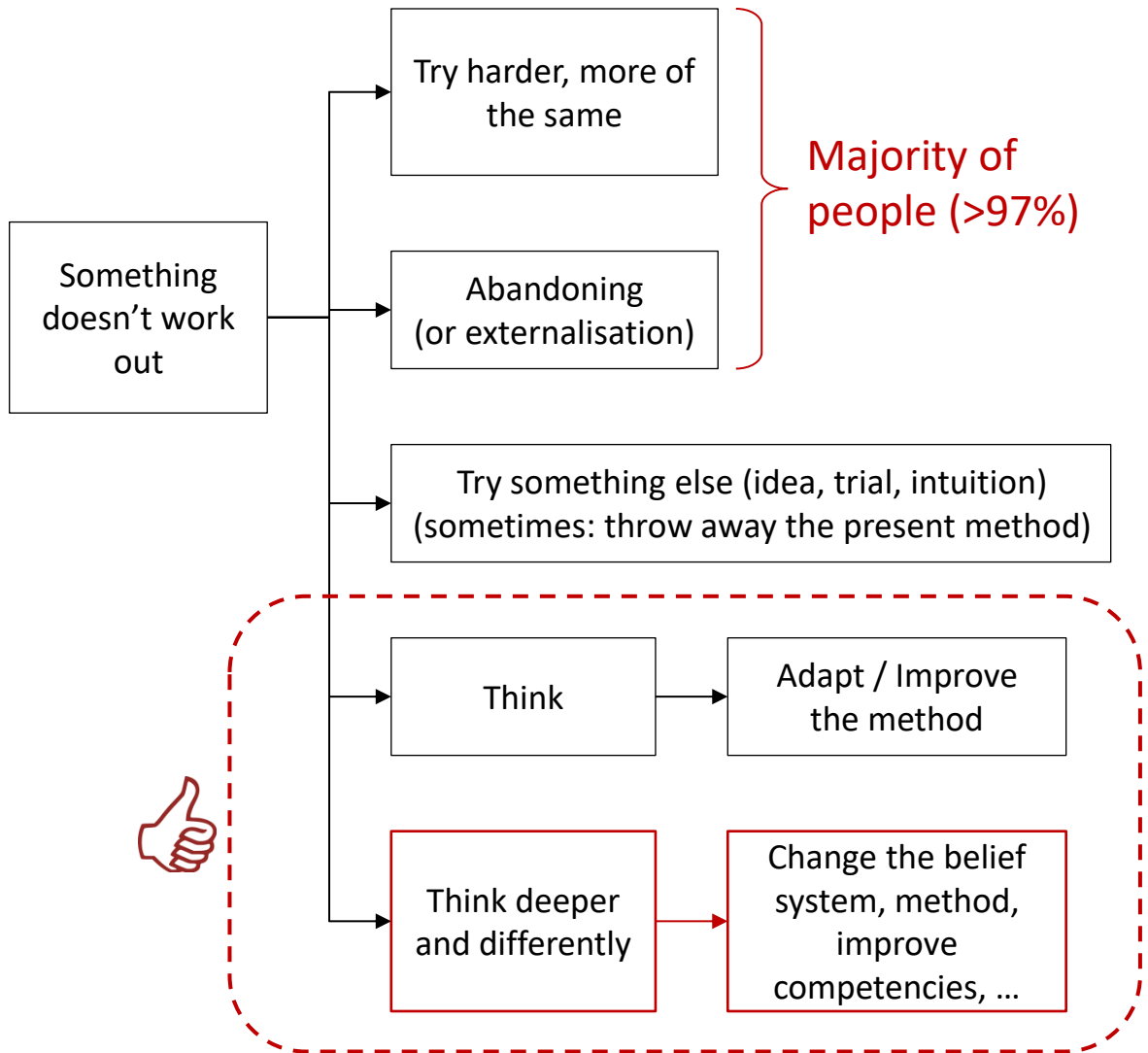
- A solution can only exist if a problem exists and has been identified.
- A solution to an unidentified problem is not a solution.
- A 'solution' is not a solution if
 - it doesn't solve the right problem
 - it doesn't solve a problem in a beneficial way.
- "SOLUTION" is NOT synonym for "PRODUCT"
 - As a product to be bought
 - As the delivered system of a project

Peter Senge's 1st Law:

"Today's problems come from yesterday's solutions."

Problem Solving Attitude

30/10/2018



Notes:

Examples:

- Delegating, subcontracting, divestiture, closing down a company branch, privatisation
- Quitting instead of improving
 - Plans don't work. Let's stop planning or plan lesser.
 - Requirements are frozen but aren't good. Let's allow them to change freely.
 - Documentation isn't used. Let's document lesser.
 - The 'Waterfall' doesn't work. Let's abandon the Waterfall model.
 - Systems Analysis doesn't lead to good solutions. Let's try hope emergence will bring good solutions.

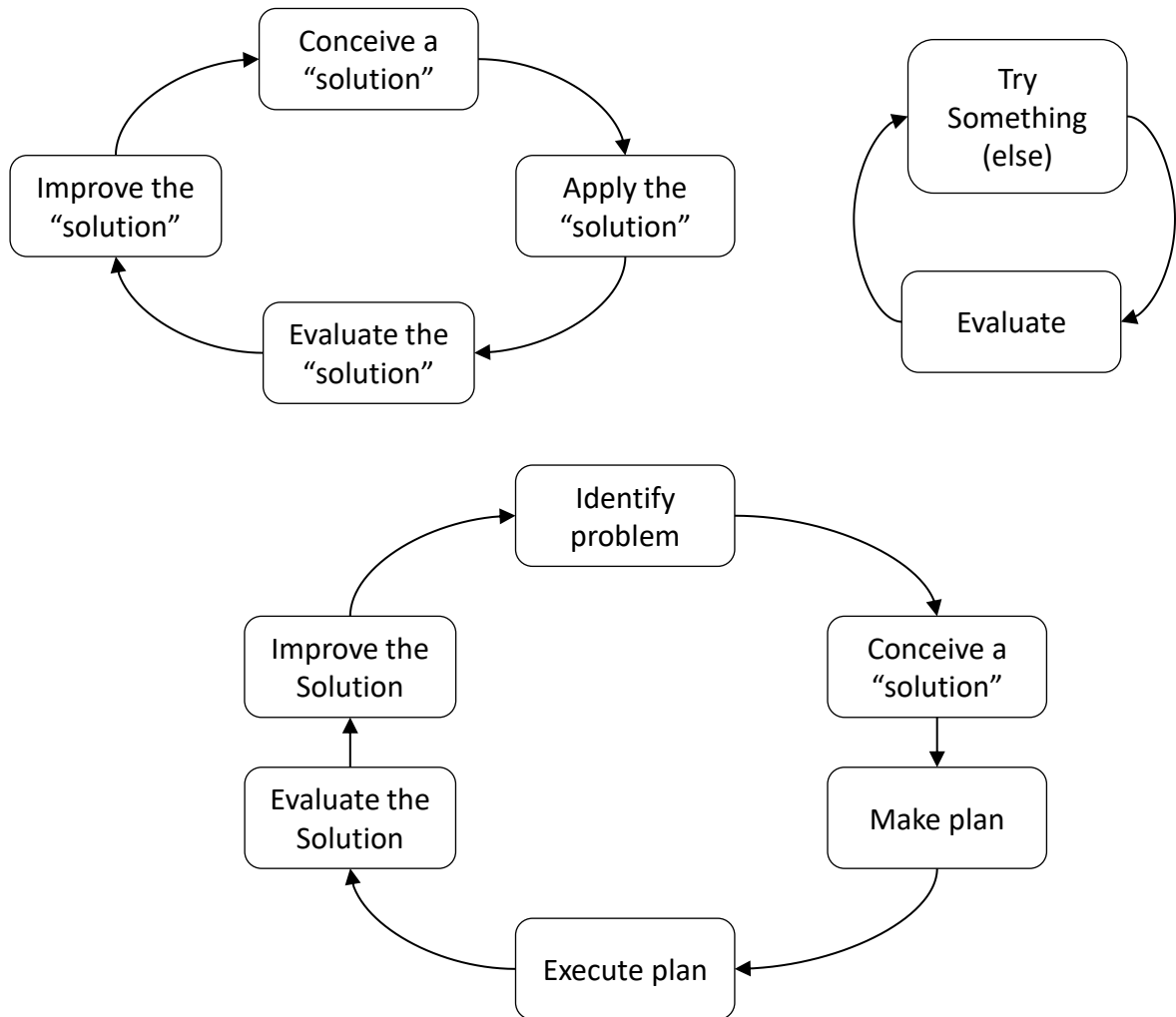
Reality: It's not the tool. It's an incorrect application or lack of skill.

Improve and Learn

Traditional Problem Solving Models

30/10/2018

TRY UNTIL IT'S OK - MODELS

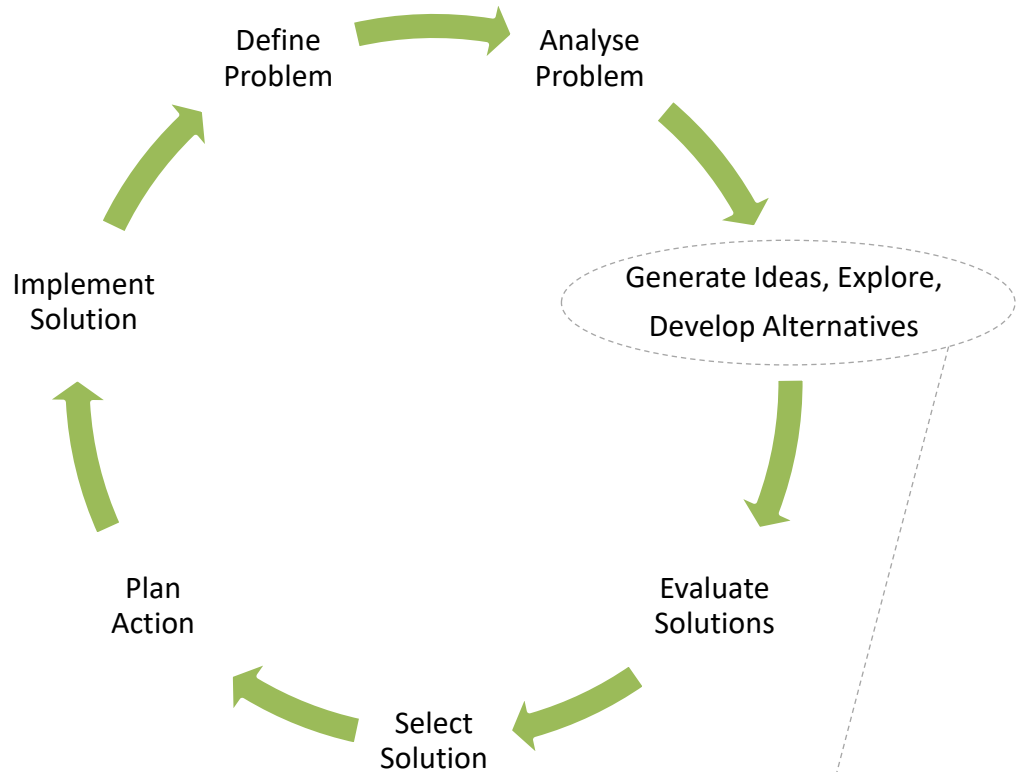


Each iteration:

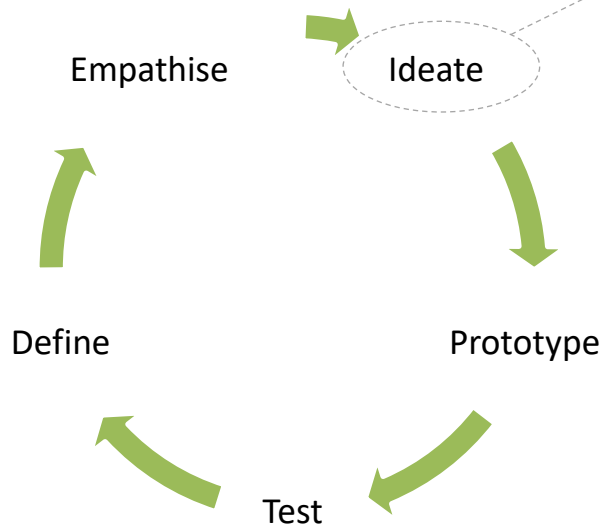
- Adds time
- Adds resources usage
- Adds rework (more waste)

Unfortunately, these models **don't help** in problem solving. The whole difficulty of problem solving is concentrated in the step "conceive a solution". The models don't explain how to solve a problem.

There are better approaches than cycling until a solution is obtained.



Design Thinking



Problem: How to augment a restaurants profits?

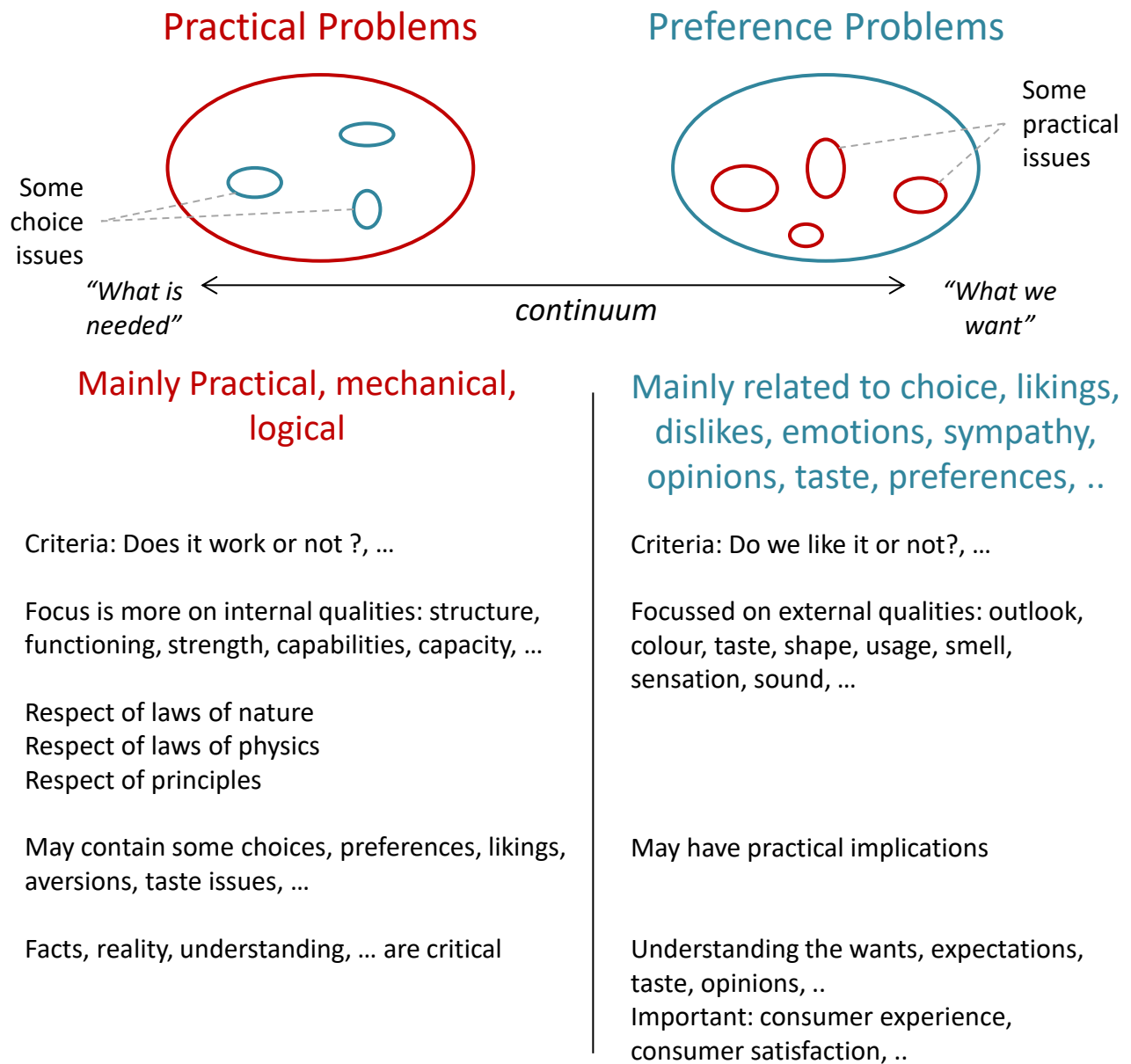
1) Rational, Analytical, Structured Approach

1. Identify the problem
 - Study of the location, visibility, accessibility of the restaurant.
 - Study of the clients (who, do they come back, complaints, ...)
 - Analysis of the welcoming process, the restaurant service, waiting times, ...)
 - Analysis of the processes in the kitchen
 - Analysis of the restaurant and kitchen collaboration
 - Analysis of the offered plates
 - Analysis of the cooking processes
2. Solve the identified issues
 - Identify the causes
 - What can be changed and what can't
 - ...
3. Implement the solutions
4. Evaluate the results

2) Creative Approach

1. Generate ideas by posing the question: "What can be done to make the restaurant profitable?"
 - Introduce a new theme every month (Mexican food, Italian food, sea, hunting, ...)
 - Get some more healthy child dishes
 - Offer a little gift to children
 - Install a playground for children
 - Organise now and then an activity or performance for children
 - Offer rare dishes to customers
 - Disguise the waiters and waitresses and decorate the restaurant
 - Install a bar
 - Get more adapted music
 - Get waiters and waitresses better trained
 - ...
2. Think of the feasibility, investment, cost, implementability, risks, consequences, ...
3. Select and plan the order of the ideas
4. Implement the ideas

Practical versus Preference Problems



**A practical issue has to be solved essentially in a practical way.
Choice issues should essentially be solved in a different way.**

Notes:

Does it make sense to discuss the colour of internal parts of a pump in an industrial plant? Or discussing the technical characteristics and production process of a romantic dinner you are about to enjoy in a nice restaurant?

Do information systems in corporate environments solve mainly practical problems or rather choice problems?

How do you see the balance (practical vs choice) in consumer IT products compared to the balance of information systems? What is the impact on the development approach?

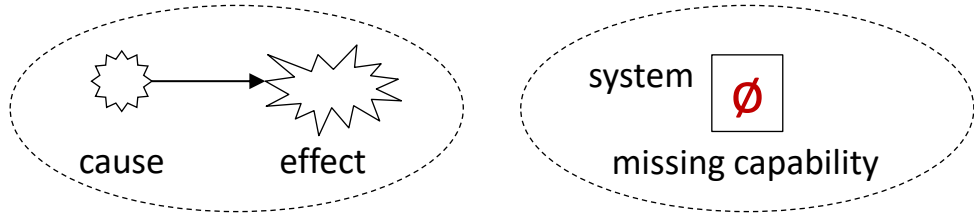
What happens if we try to solve a practical problem with an approach that suits much more choice problems?

Two Types of Problems

30/10/2018

1

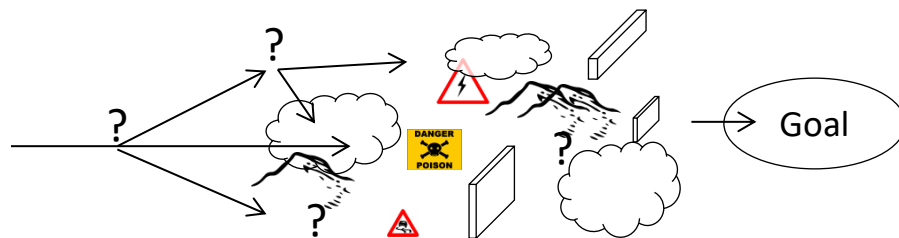
Obtaining a tangible result? → Solution: **SYSTEM**



- Production of products or services
- A dysfunctional system, a system that doesn't function as expected or which doesn't produce the expected results
- A system missing abilities or capabilities

2

Obtaining intangible result ?
(Reaching a GOAL) → Solution: **STRATEGY**



Goal seems to be unreachable - Path to a goal is unknown.

Obstacles, uncertainties, unknowns, risks, dangers, traps, dead-ends and uncharted territories...

Examples: Winning a battle; winning a medal in a sport; increase the market share, learning strategy, ...

Notes:

A system executes processes and a system may be part of a solution to reach a goal.

Some problems may require both, a system and a strategy (process) to be solved.

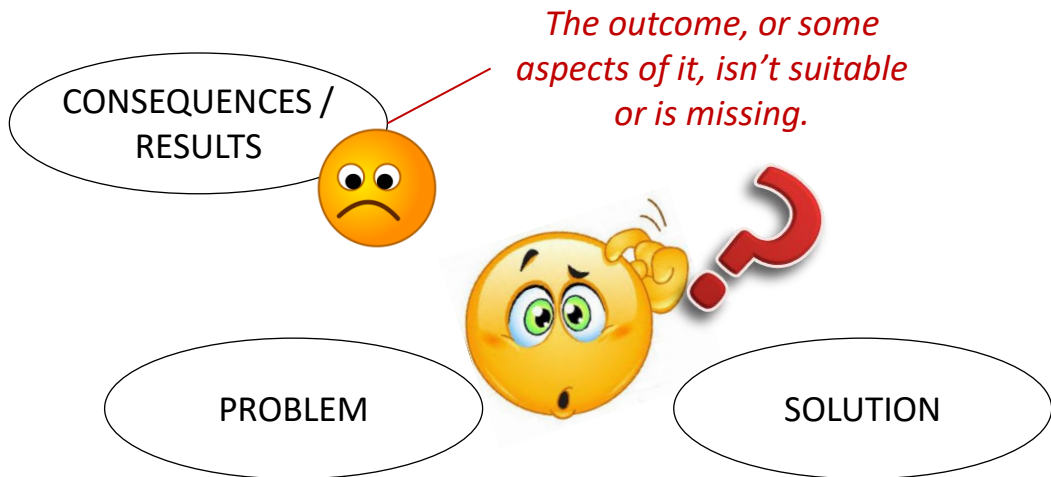
The conception process of systems and strategies differ from each other.

We need a process (strategy, approach, methodology) to build the right system rightly. An inappropriate strategy will hinder the conception.



A Core Aspect in Problem Solving

30/10/2018



The main problem in problem solving is (often & mainly)



THE UNKNOWN



- Not knowing what is happening
- Why it is happening
- What the causes are
- What (all) the consequences are
- And not having (not knowing) the solution
- ...



The problem solving initiative is (often/mainly) about

searching and solving the unknowns.

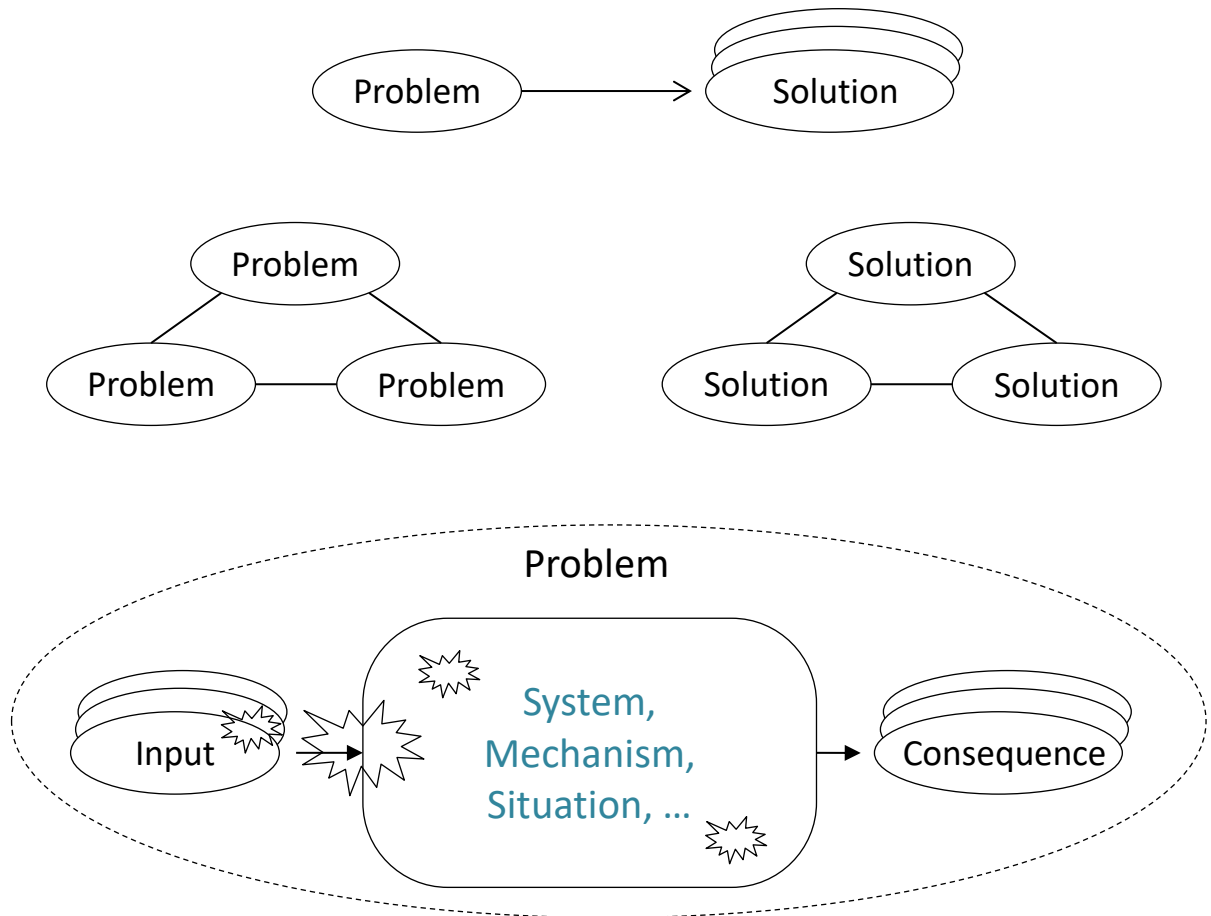
It is about acquiring understanding.

*Understanding the anatomy of problems, systems, processes, ... helps
(and the dislike of effort that will be required)*

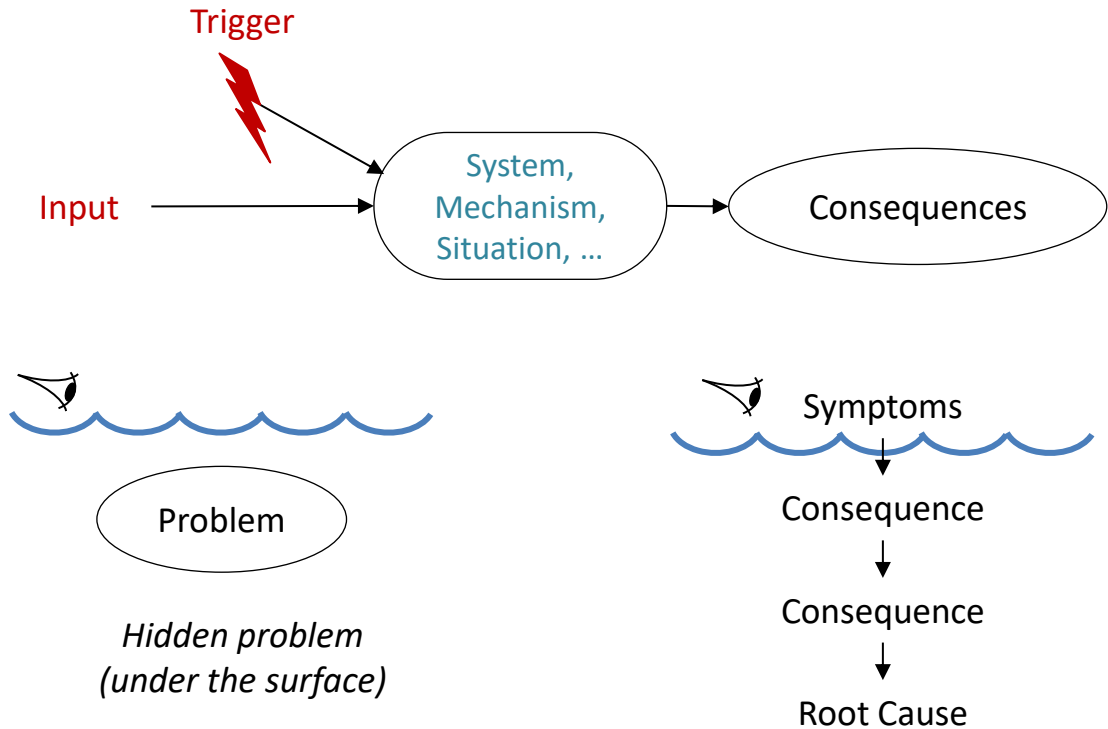
Anatomy of Problem

30/10/2018

Focus is on 'System'-Problem (not on 'Strategy'-problem)



- A problem is solved by a solution. (problem – solution relation)
- One problem may have several solutions.
- Several problems may have a single solution.
- Not all candidate solutions are equal.
- Different problems may be linked. (common deeper cause? Solving together?)
- Different solutions may be linked.
- Several causes (☼) may exist, or several conditions may form a cause.
- A cause can be inside and/or outside the system.
- A problem creates one or more consequences (missing or bad results).



Consequences can be

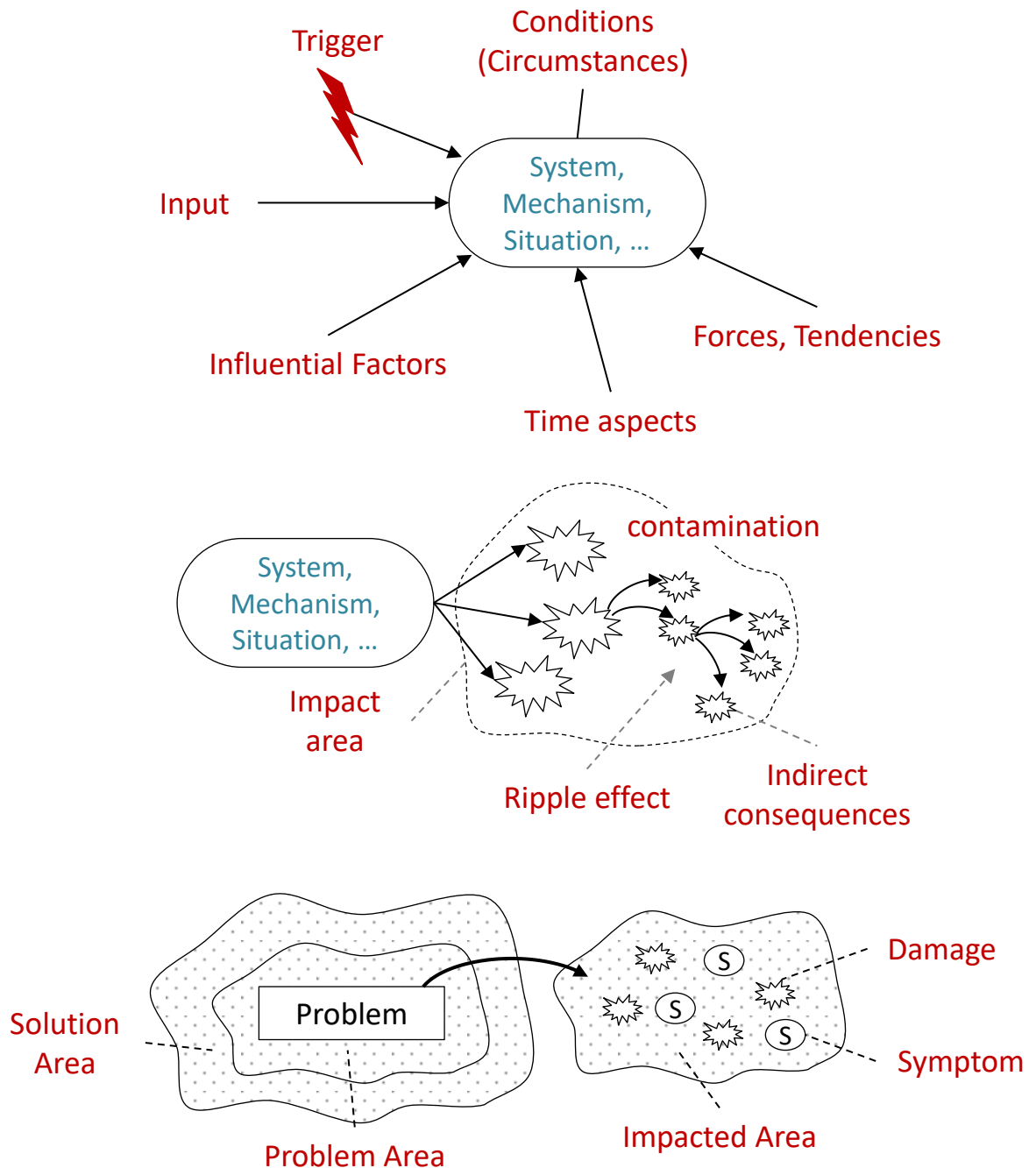
- Missing
- Inappropriate
- Visible / invisible; detected / hidden; known / unknown
- Positive/ negative
- Spread or concentrated
- Specific, generalised
- Irregular, regular or continuous
- Annoying, embarrassing, painful, causing discomfort, irritating, ...
- Temporary, permanent
- Reversible, irreversible
- ...

A **symptom** is a noticeable and/or experienced consequence or a side-effect; a sign of an underlying problem

Anatomy of Problem

30/10/2018

Behaviour of system can be influenced, hindered, obstructed, ...



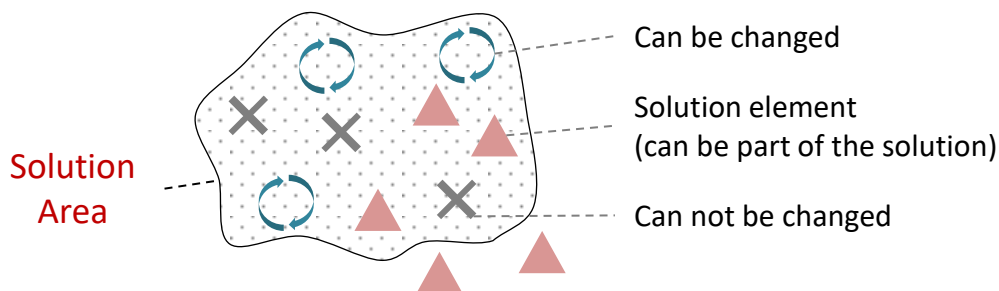
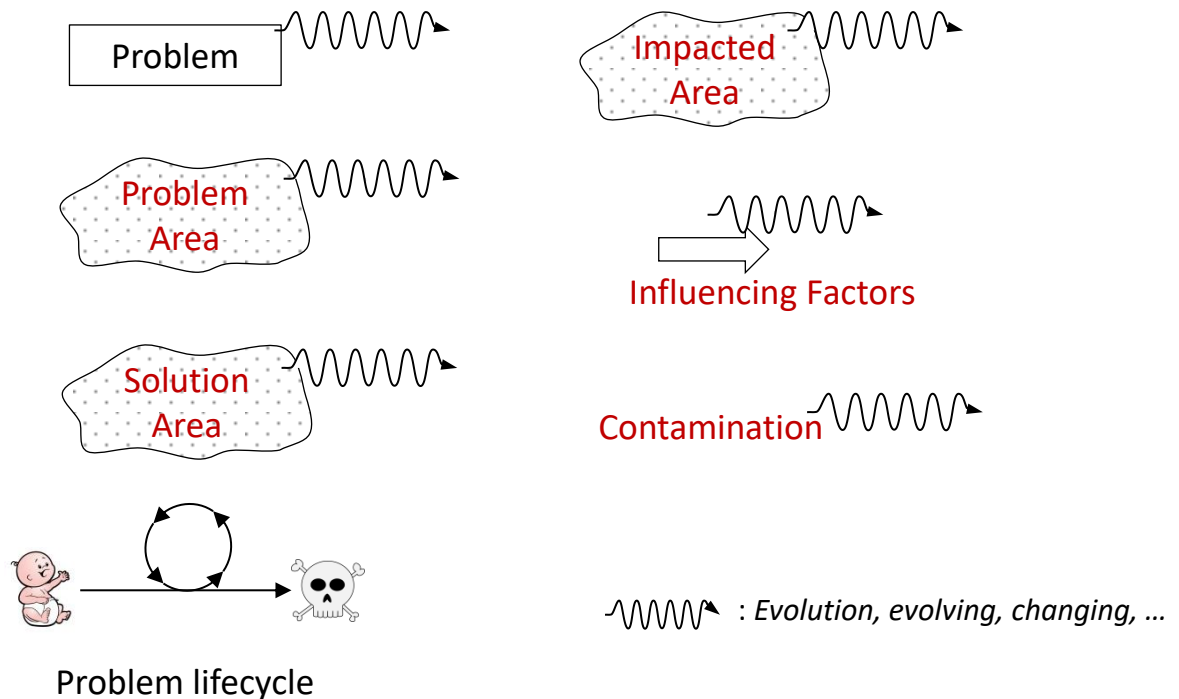
Notes:

The identification of indirect consequences, contamination, impact area, etc. help to correctly diagnose problems, understanding problems, solving them and cleaning up their consequences.

Anatomy of Problem

30/10/2018

Problems are (often) dynamic or exist in a dynamic environment



Solution Element

Anything that can be part of a future (candidate) solution

Candidate solution

Any potential solution; the alternative solutions

Understanding the following concepts helps
in analysing and solving the problem

- | | |
|---|--|
| 1. Complaint | 12. Trigger, Event |
| 2. Symptom | 13. Input |
| 3. Impact | 14. Condition, Circumstance |
| 4. Consequence | 15. Cause-effect-Relation |
| 5. Risk | 16. Dynamic: System,
Mechanism, Process |
| 6. Damage, Loss | 17. Problem's Evolution /
Life |
| 7. Problem | 18. Forces, Tendencies |
| 8. Nature of the problem | 19. Problem Area |
| 9. Perceived (subjective)
problem | 20. Impact Area(s) |
| 10. Problem Domain | 21. Root-Cause Analysis |
| 11. Indirect Effects / Ripple
Effect | |

Note: Concepts related to the solving-part of the process
are presented later.

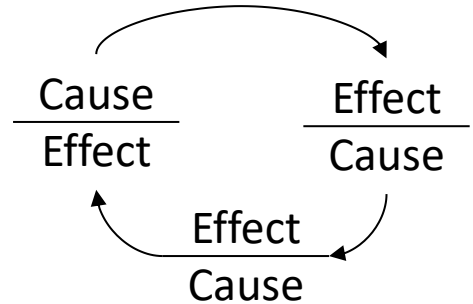
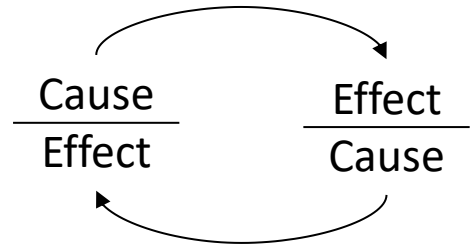
Problem Solving – Cause - Effect Relation

30/10/2018

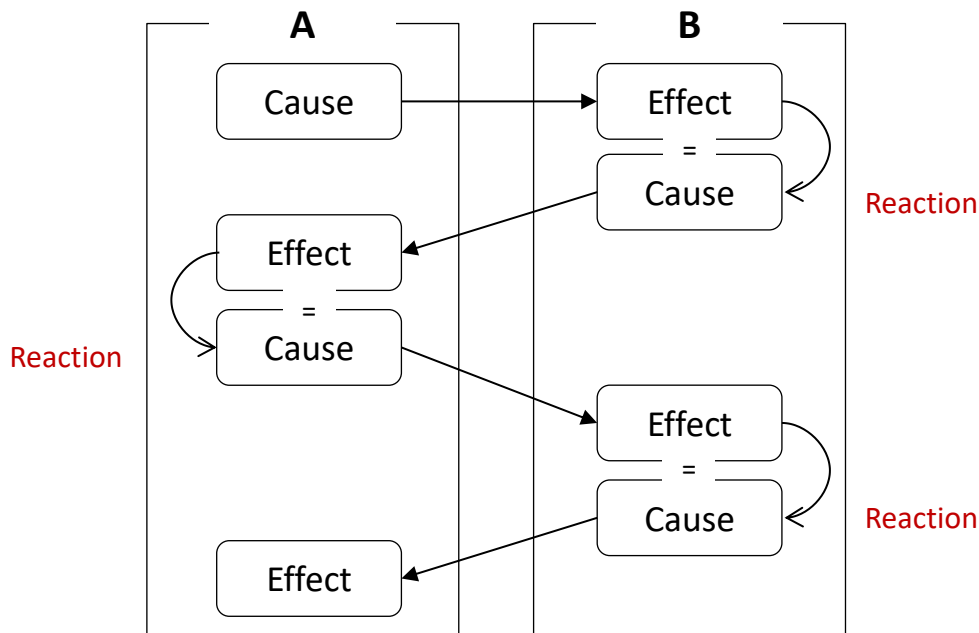
Cause \longrightarrow Effect

Cause $\begin{cases} \nearrow \text{Effect} \\ \searrow \text{Effect} \end{cases}$

$\begin{matrix} \text{Cause} \\ \text{Cause} \end{matrix} \begin{matrix} \nearrow \\ \searrow \end{matrix} \text{Effect}$



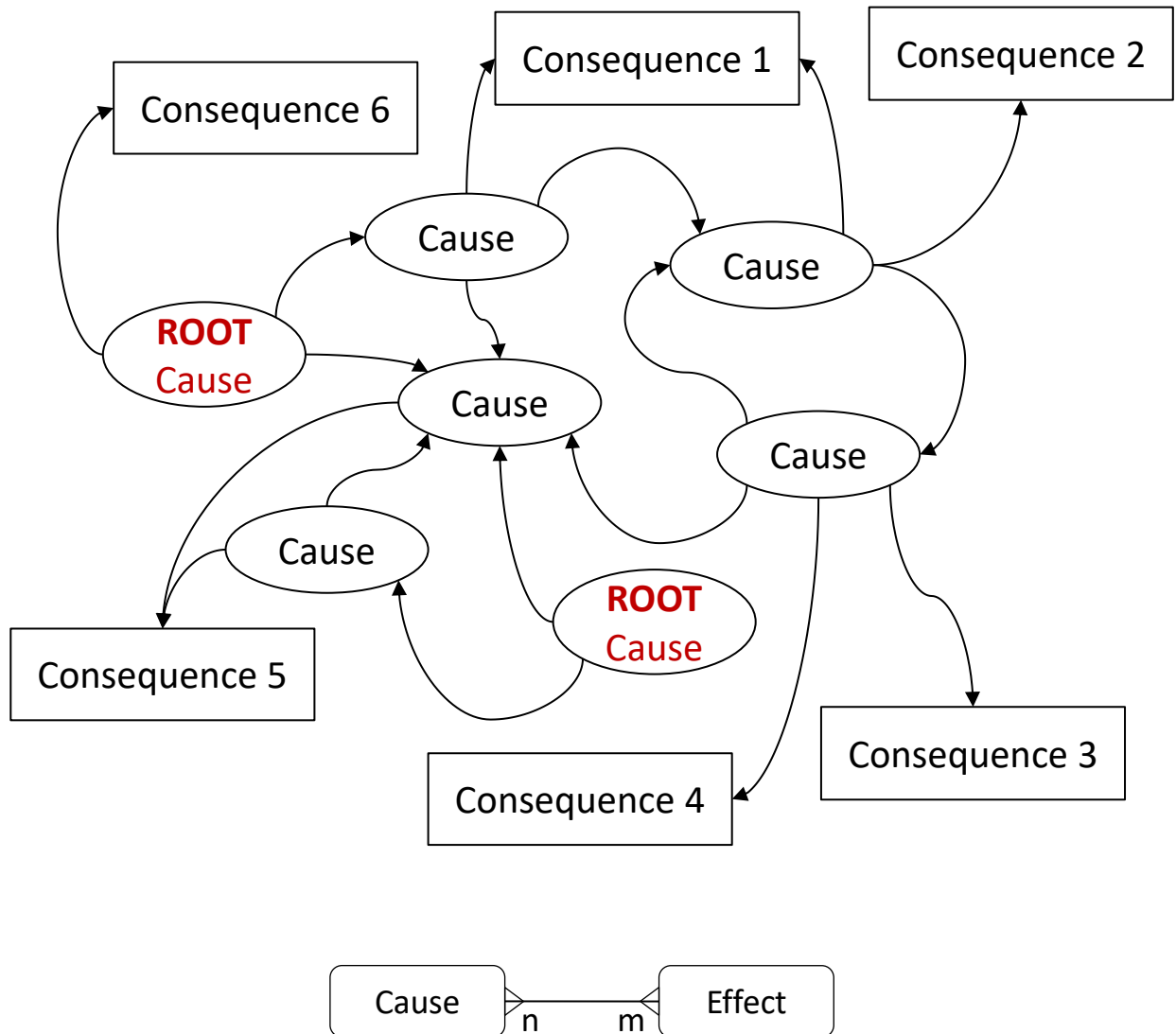
$\frac{\text{Effect}}{\text{Cause}}$: Effect becomes itself a cause of another effect



- Implies presence of a mechanism
- Not always just a single cause-effect relation. It can be much more complex.

Problem Solving – Network of Causes

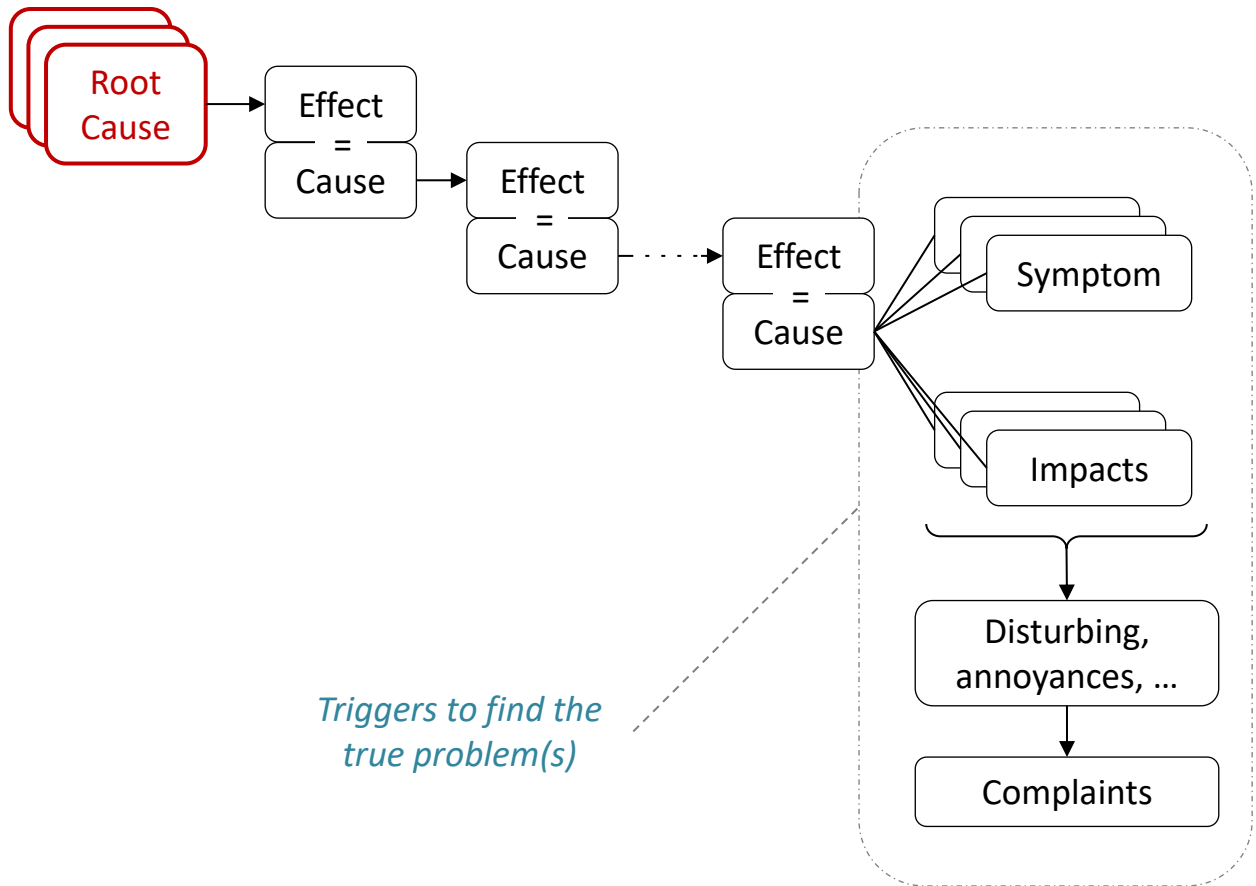
30/10/2018



Sometimes, no root cause can be identified. It might be better to start solving one or a few causes. This may activate a dynamism that will solve many problems.

Note that a cause alone may not materialise on its own. Circumstances also play a role. Are these circumstances normal? What caused these circumstances?

Example: finding a job may solve many other of someone's problems.



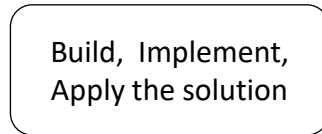
Treat anything labelled as “problem” or as “cause” as an “effect”!

- This is the problem to be solved.
- Really ? What caused the problem
- This is the cause.
- Really ? What caused the cause?
- This is the cause of the cause. (or “These are the causes...”)
- And what caused the cause of the cause of the problem?
- Etcetera.

Basic Problem Solving Approach (1)

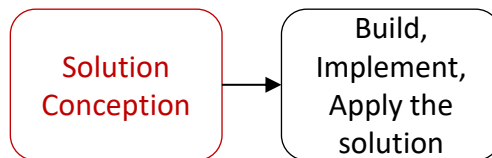
30/10/2018

Clear and easy solution



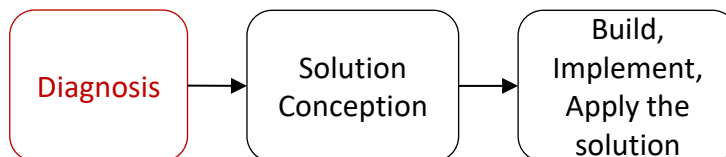
Assumption: The solution is simple and known.

Solution is unknown or complex



Assumption: The problem is known.

Problem is not clearly identified and understood



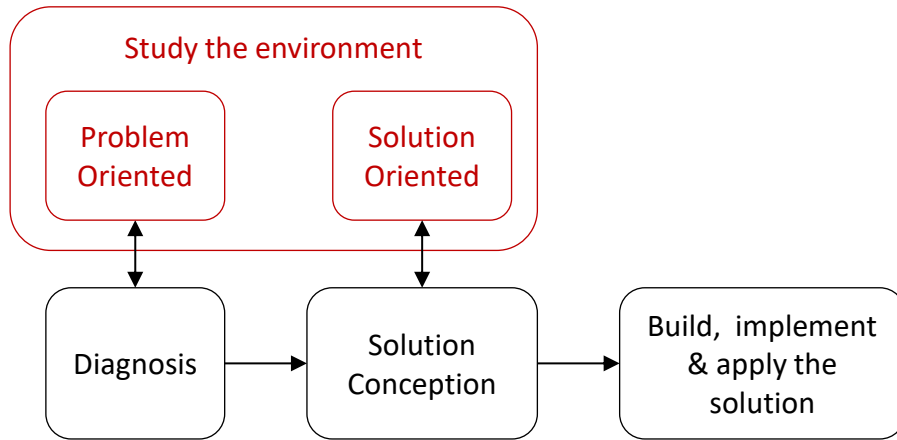
Assumption: The environment is rather simple and very well known.

Communicating the diagnosis is important. The diagnostic has to be established and shared. Only then individuals can work together on a same solution.

Basic Problem Solving Approach (2)

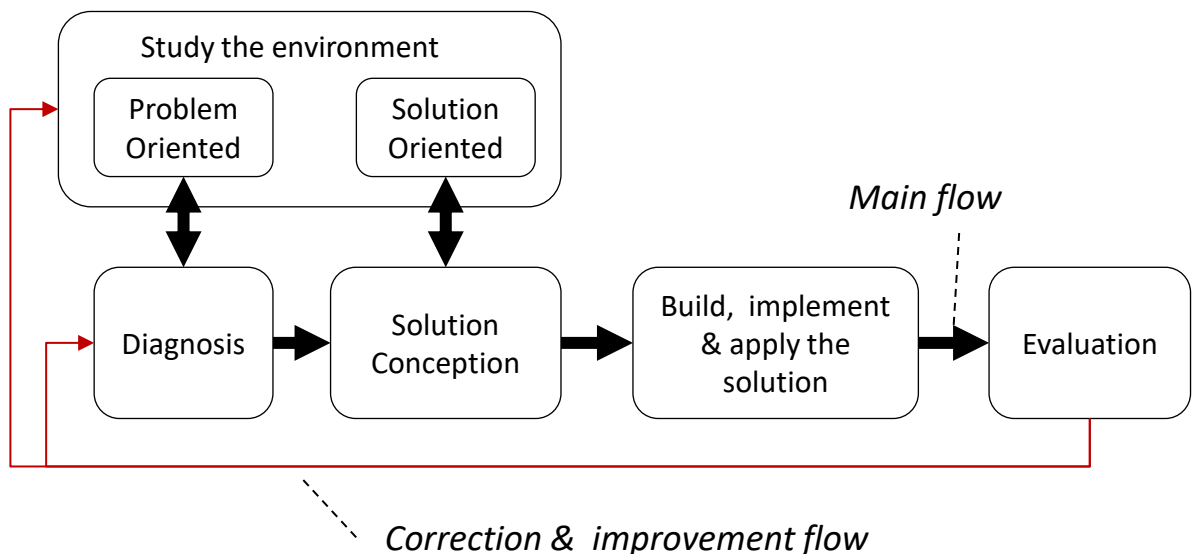
30/10/2018

Complex environment, problem is not clear - solution is complex



Assumption: Assumes the solution is built flawlessly from the first time.

Second built solving teeth problems & continuous improvement



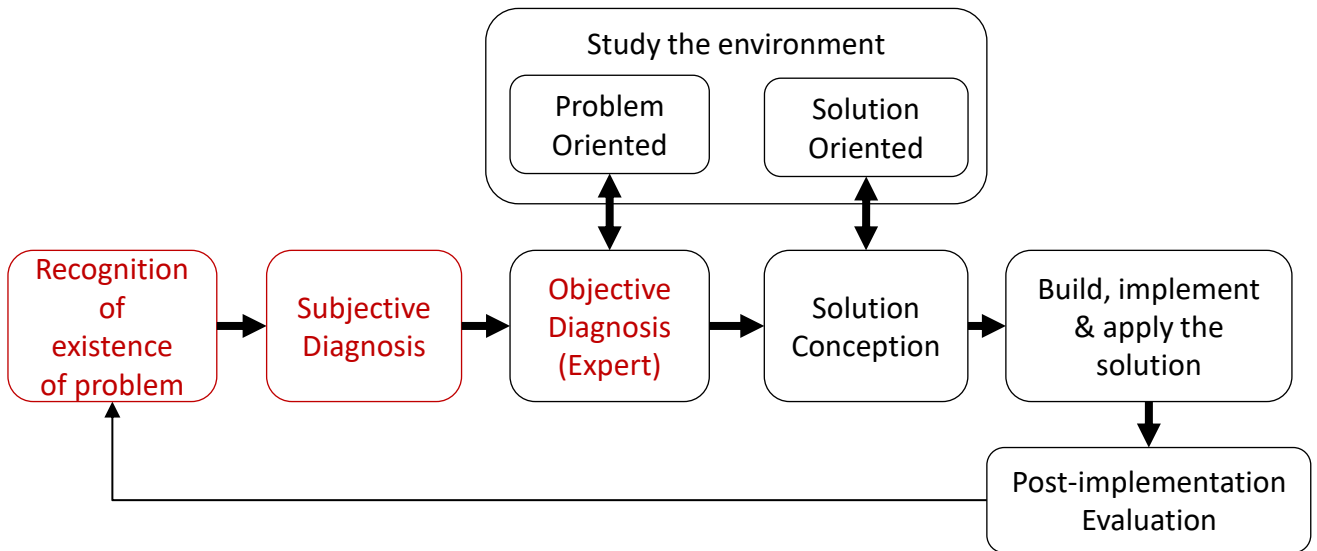
In general, the main part of the solution can often be built right from the first time. The backward loops are necessary for rectifications, adjustments, improvements and subsequent additions.

Basic Problem Solving Approach (3)

30/10/2018

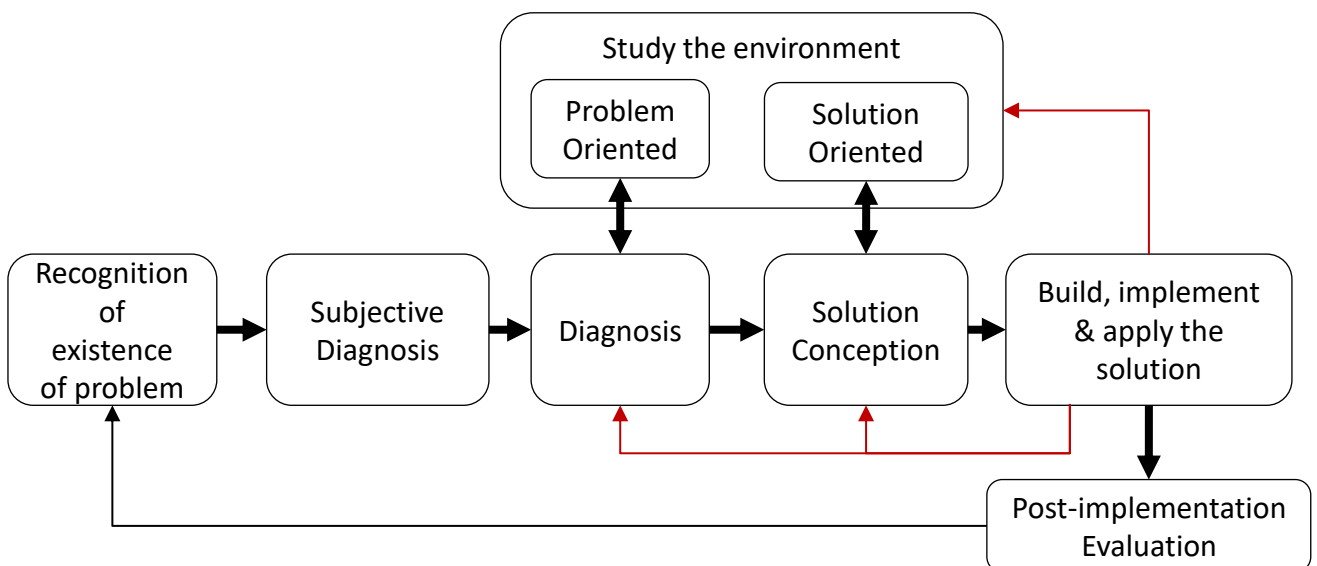
Diagnosis has to be performed by an expert

The person detecting the problem, isn't an expert. An expert is needed for rightly diagnosing the issue.



Each task (phase) isn't necessarily performed flawless

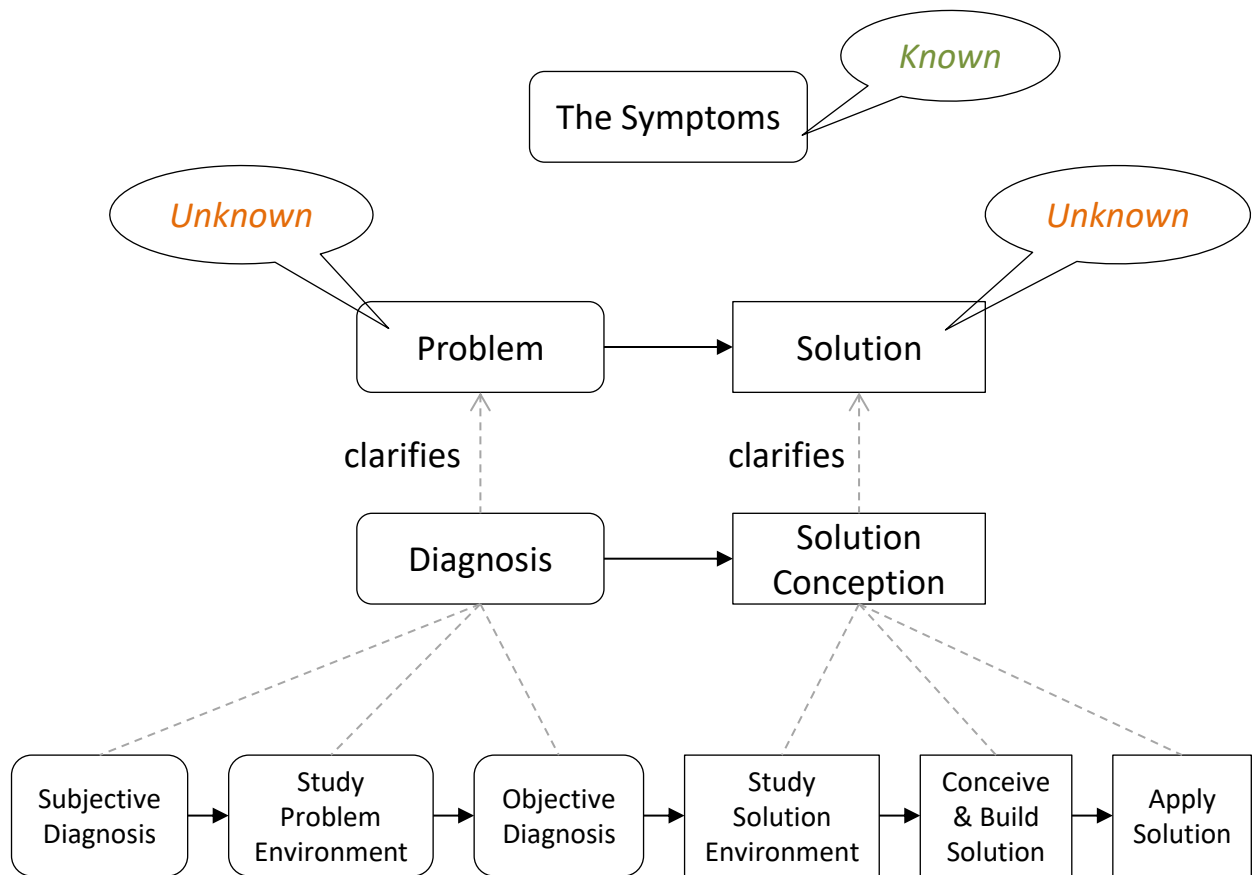
Although it is not intended, work of earlier activities may have to be clarified, checked, corrected, refined or expanded.

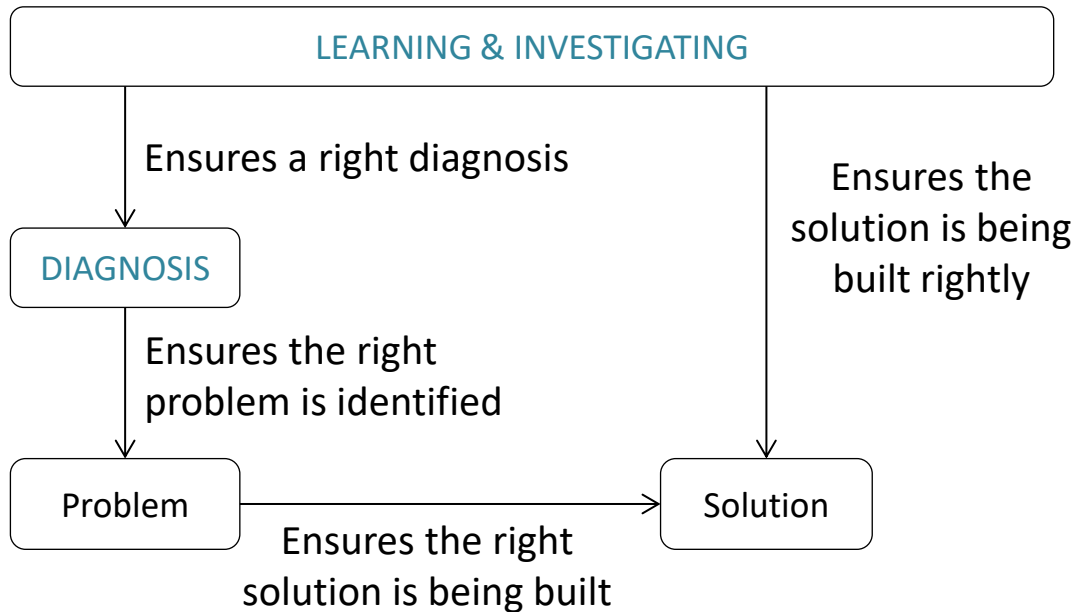


Subjective diagnosis vs Diagnosis: subjective diagnosis is diagnosis of non-specialist. Diagnosis is posed by expert.

Solving the Unknowns

30/10/2018





A problem can't be solved unless the diagnosis is right.

A solution can't be devised before the problem and its environment are properly understood.

Engineering

= Methodical Approach

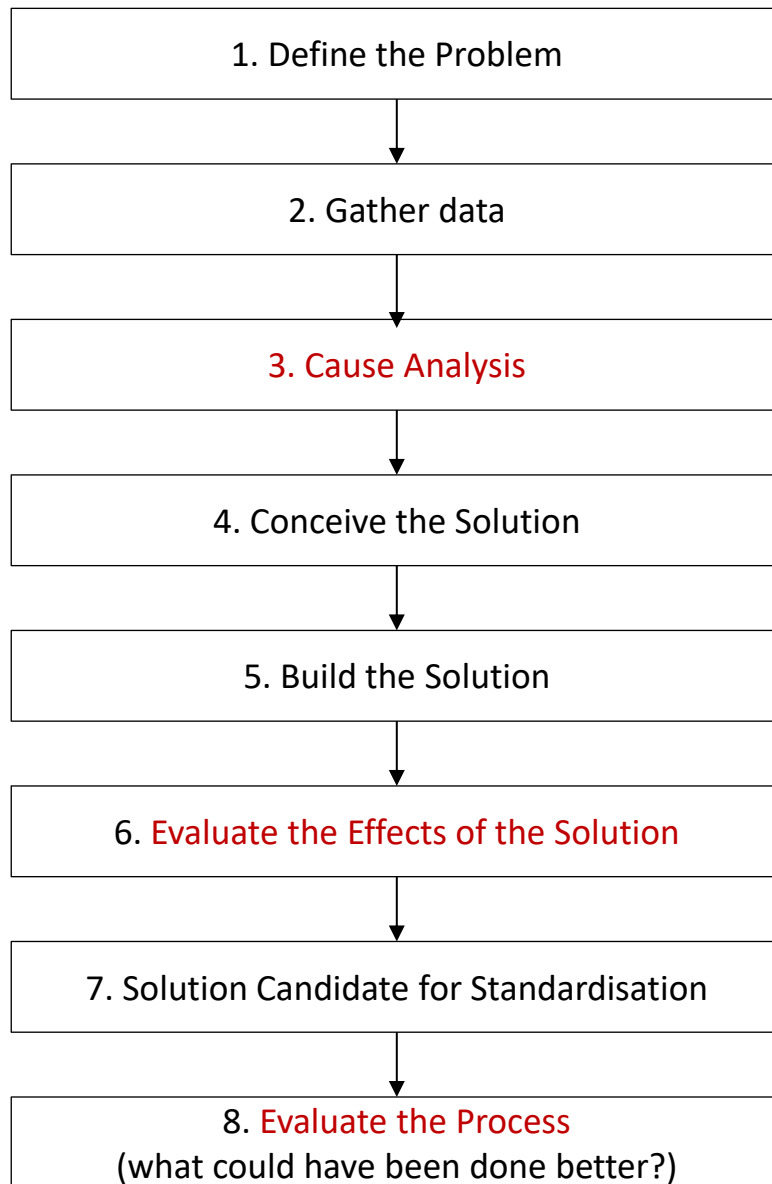
= Based on Problem Solving Approaches

Doesn't fit the true spirit of engineering :

- Taking actions with little understanding
- guessing
- repetitively trying until success
- counting on luck and accept increased risks.

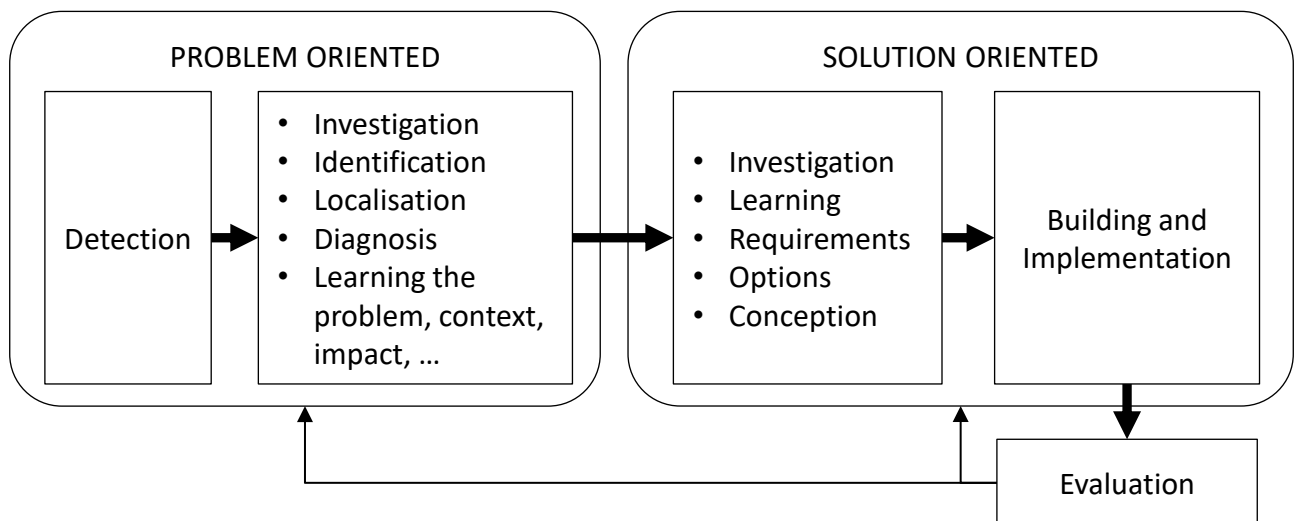
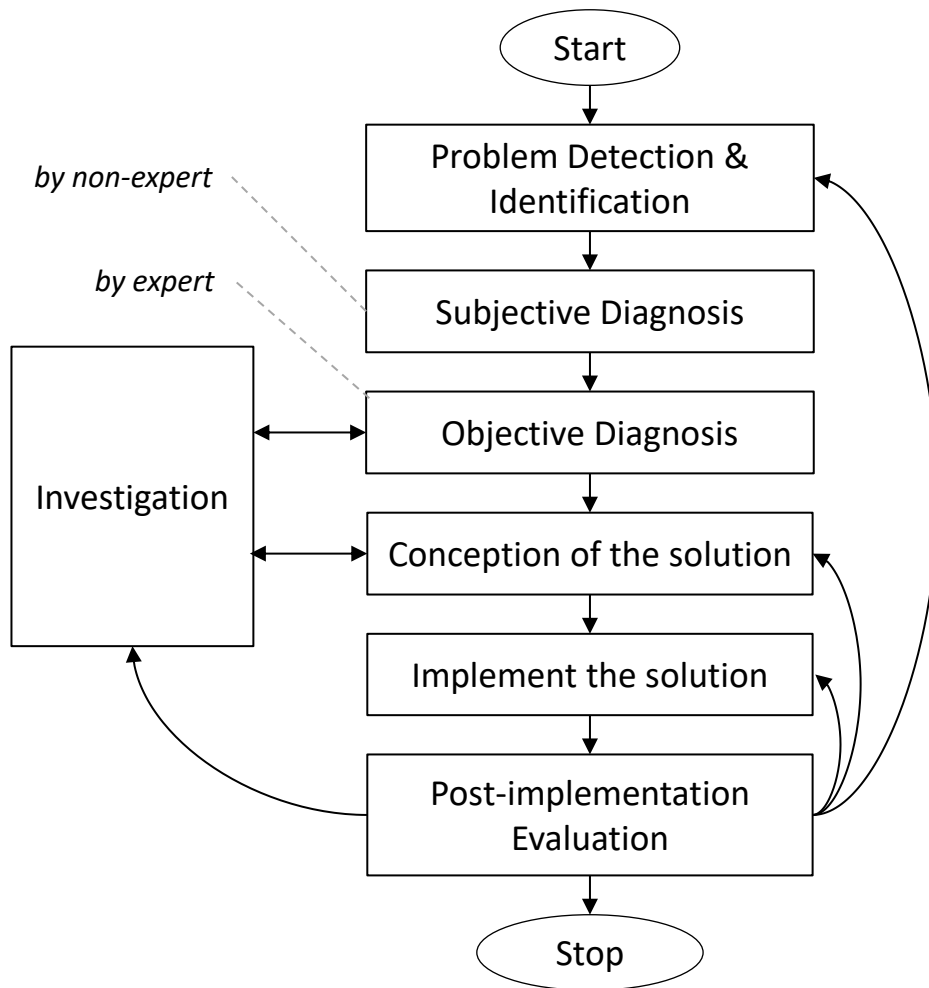
A Simple Problem Solving Process

30/10/2018



Problem Solving Approach

30/10/2018



1. **Awareness** of the existence of a Problem

We are aware that “something” is wrong but we don’t know what. The problem itself can not be pinpointed. This happen often through symptoms, consequences and results.

2. **Identification** of a Problem

The problem is recognised. It is vaguely located and vaguely known to “get a name”.

Trying to solve the problem at this state is likely to result in symptomatic solution.

3. **Localisation** of a Problem

The problem has been precisely localised. The circumstances and cause(s) are known.

4. **Understanding** of the Problem

The true root cause(s) is/are identified. The whole mechanism leading to the consequences and symptoms are understood. The consequences throughout the company are identified. The behaviour, evolution and importance of the problem are precisely understood.

5. **Deeper Understanding** of the Problem

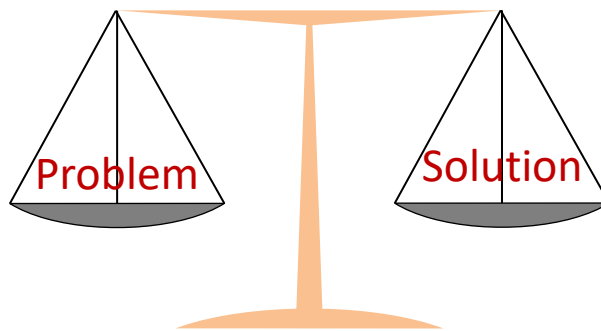
These understanding is about the underlying principles, beliefs, the assumptions, the designer’s reasonings and reasons, the WHY’s, ...

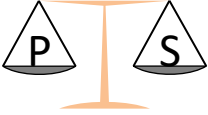
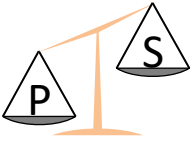
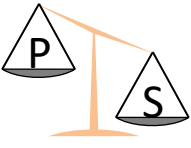
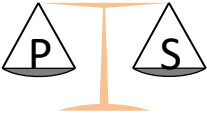
Complexity of Problem and Solution

30/10/2018

Do we need to spend more attention on the problem or on the solution?

Note that if the diagnostic is wrong, it will be hard to solve the problem (guess, luck, 'intuition', many trials, ...).

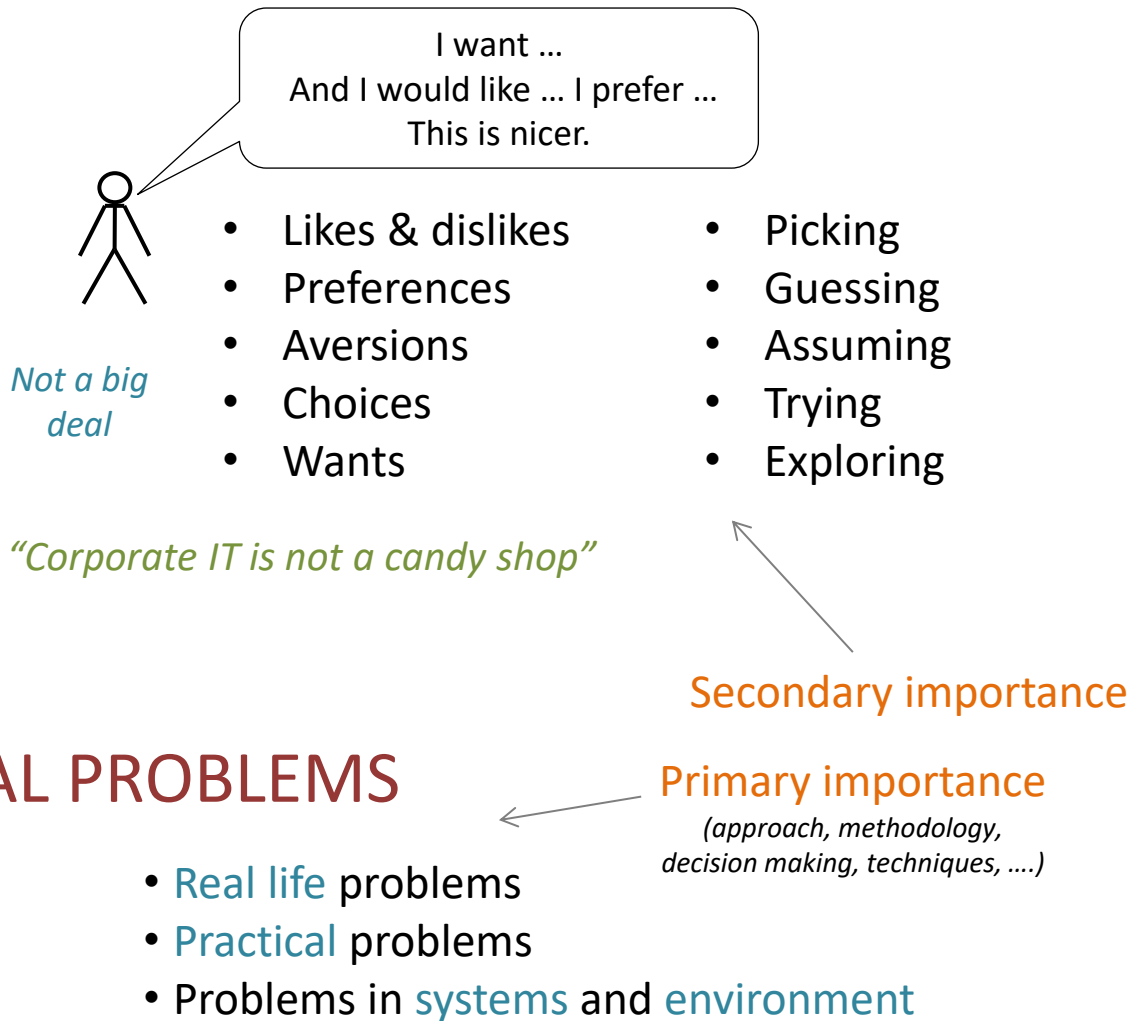


		Problem	
		Simple	Complex
Solution	Simple		
	Complex		

No relation between complexity (or size) of problem and complexity (or size) of solution.

A complex problem may be solved with a simple, small and/or cheap solution. A simple problem may require a complex, huge and/or expensive solution.

Art, Leisure, Clothing & Candy Shops



Real problems require real solutions

- Understanding of the situation, real world, system
- Right diagnostic, real solutions
- Decisions, reasoning, arguments
- Tastes, preferences and the like are of secondary importance or optional.

- These model depict a general, ordered and ideal process. They show a sequence of aspects to be considered.
- They don't describe a strict sequence.
- Overlaps are possible.
- If something is wrong, unclear, ambiguous, unverified, ... perform the activity that solves it and go on.
- If necessary or useful, the problem solver can jump backwards or forwards in the process.
- A step can be as short as the mind considering an aspect and may happen in a blink of an eye. But it may take hours or days as well.
- The problem/solution environment might already be known. However, often a problem occurred because we didn't understand something of this environment. A deeper understanding might be required. Problems teach us something.
- The study of the problem environment may provide the required insight and therefore may make the study of the solution environment useless.
- Such a process model is a basic template that has to be adapted and enriched to the specific problem. For example:
 - maybe different smaller solutions may together solve a single problem
 - maybe some preparatory work needs to be done
 - maybe a solution has to be applied in phases

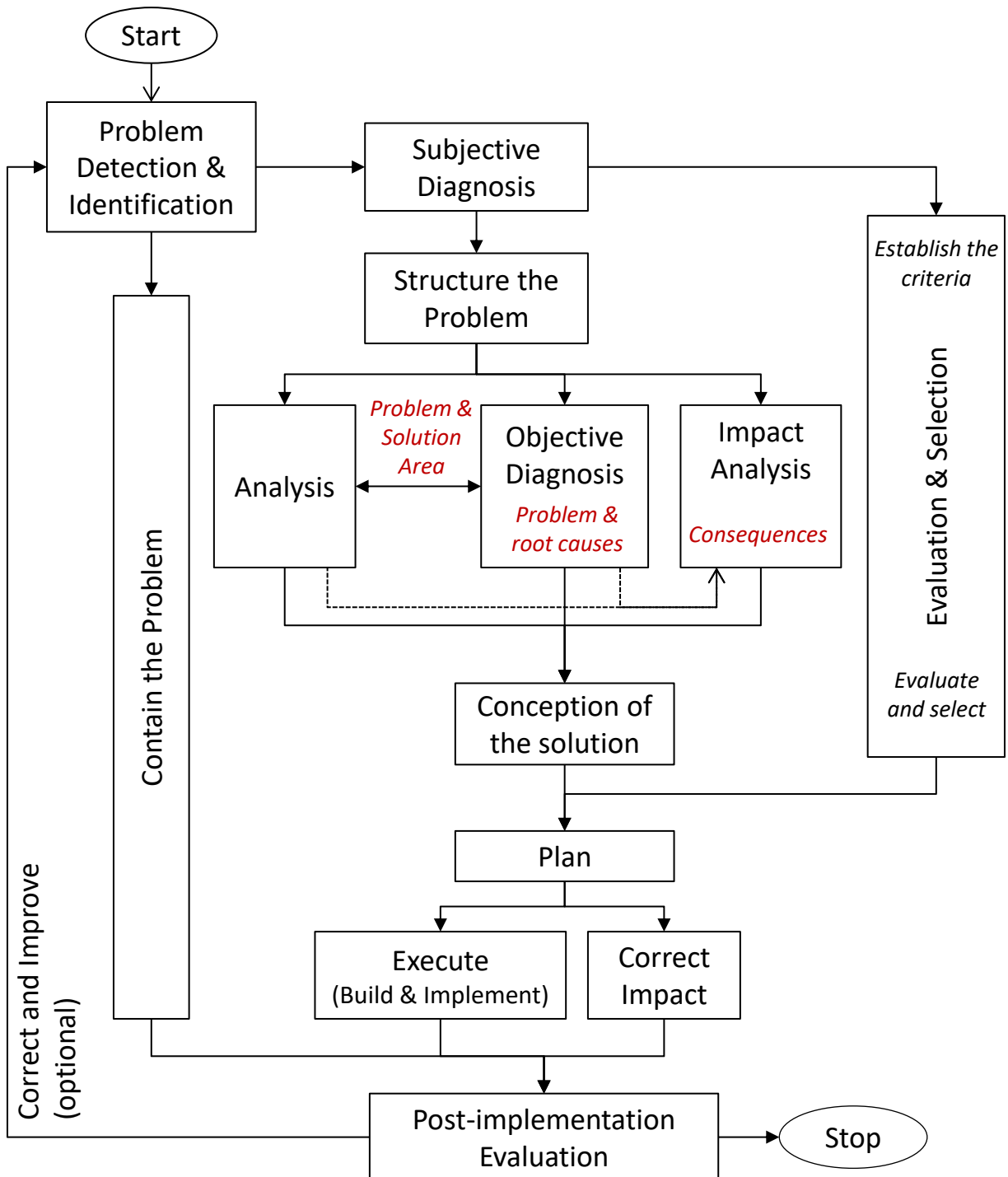
More Problem Solving Activities

30/10/2018

1. Describe the problem
2. Determine nature of problem
3. Categorise the problem
4. Describe the objective
5. Information gathering
6. Symptoms analysis
7. Stakeholders analysis
8. Do background research
9. Analyse the context and situation
10. Understand the problem
11. Identify assumptions
12. Determine the causes
13. Determine the factors contributing to the problem
14. Determine the root cause(s)
15. Determine the impacted areas
16. Determine and evaluate the impact (damages)
17. Conceive a solution to clean up the damages and apply it
18. Determine solutions to contain the problem
19. Determine solutions to solve the impacts
20. Establish what is fixed, unchangeable, beyond your span of control and what can be adapted, changed, replaced, ...
21. Establish requirements
22. Verify requirements
23. Find elements of solutions
24. Generate ideas (brainstorming, finding analogies, ...)
25. Think of alternative solutions
26. Identify feasibility, implications, obstacles, risks, limits, cost, resource requirements per alternative solution
27. Elaborate a proof of concept, mock-up, prototype
28. Investigate collateral effects
29. Establish criteria to evaluate the alternative solutions
30. Evaluate the alternatives
31. Select the best alternative solution
32. Implement solutions to resolve the impacts
33. Implement the solution
34. Evaluate the solution
35. Improve the solution
36. ...

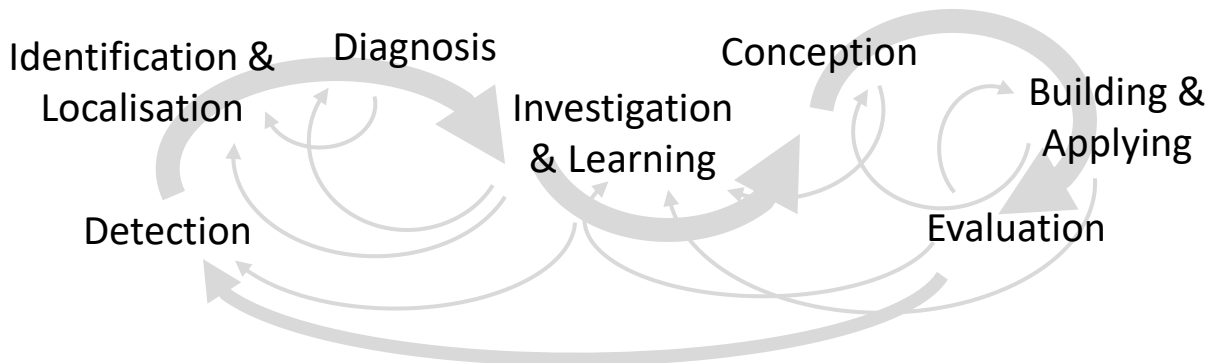
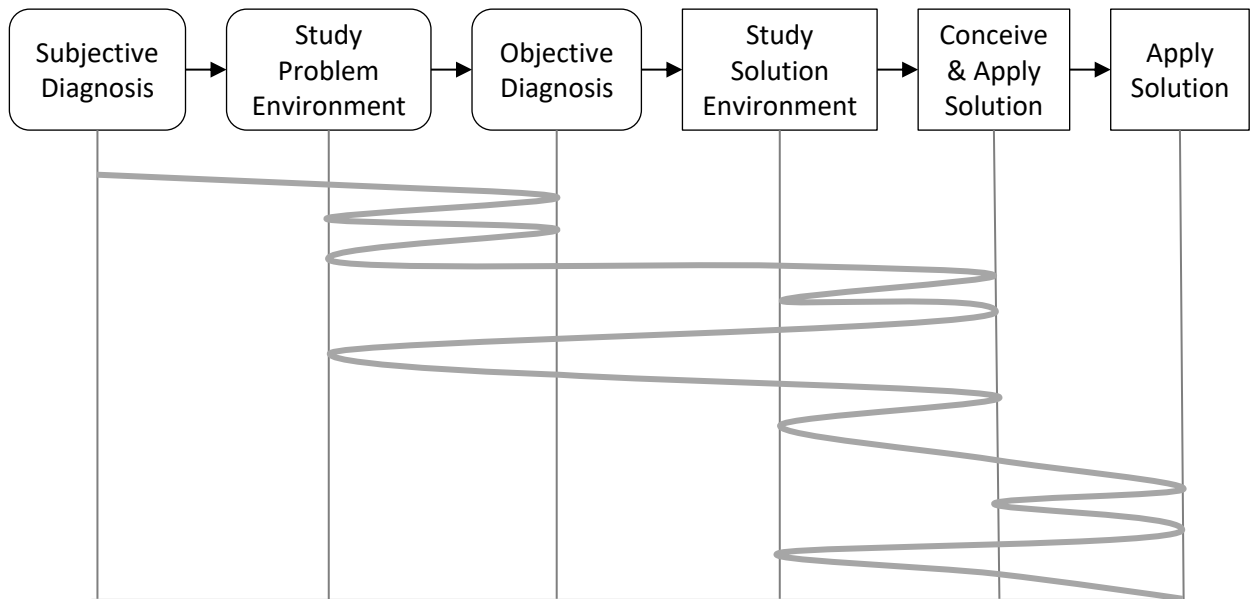
A More Elaborated Approach

30/10/2018



The Mind in Problem Solving

30/10/2018



The mind doesn't think or learn perfectly from the first time. The first analysis or answer may be imperfect. It doesn't work in a strict linear or sequential way. It jumps back to a previous step to correct or to complete a question already dealt with, and seconds later it may leap forward. There is a clear convergence towards a definitive insight and solution.

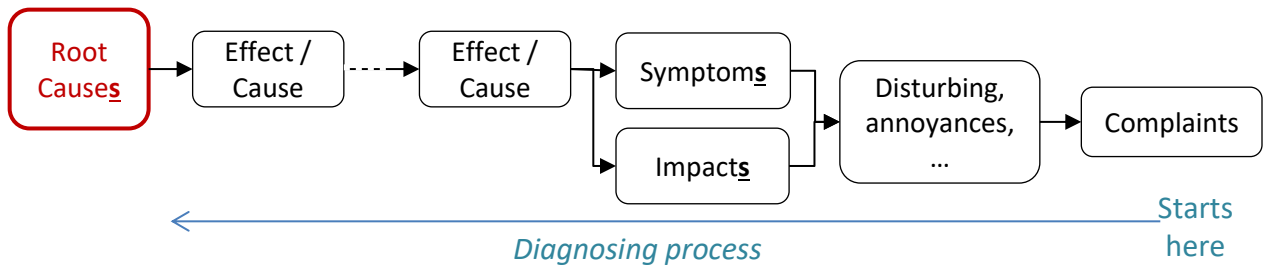
The global process won't be perfectly or strictly sequential. But a pattern of different phases can be distinguished.

The mind can somewhat be guided. By establishing what has to be resolved in what order, our attention, our focus can be better controlled. This reduces the erratic behaviour and increases the tendency towards a structured approach.

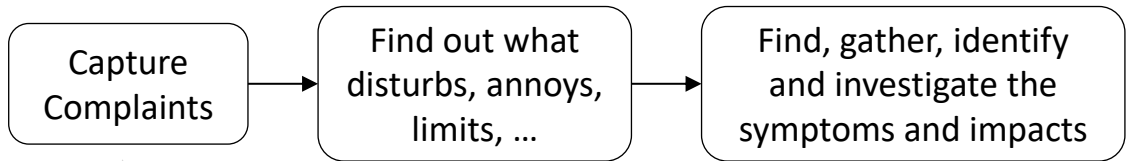
A superficial, improper and incomplete performed task produces a bad result. Learning happens then through result. More corrections are needed. More iterations will be required or each iteration will concern a larger change. Sloppy thinking and sloppy work lead to more iterations and more rework.

Diagnosis Process

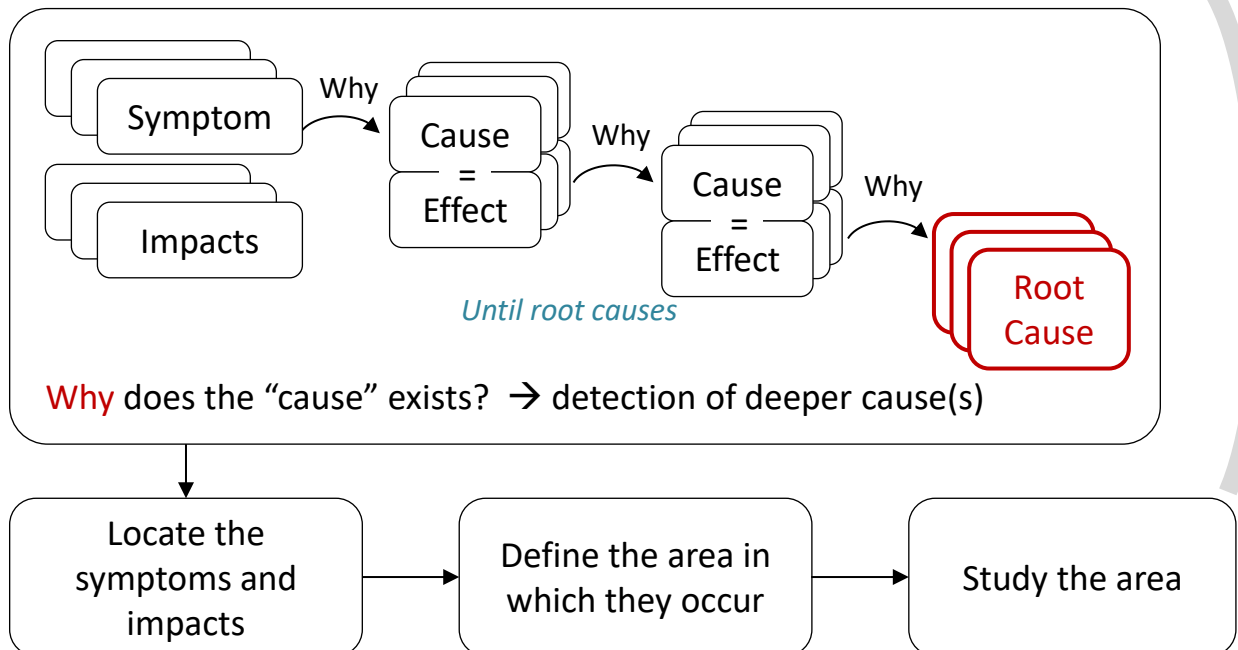
30/10/2018



Starting Point



Finding Root Causes



Notes:

If the symptoms or other disturbing effects (impacts) are labelled as 'problem', no effort will be made to question their occurrence and no investigation to find the true problems, the root cause will be initiated. Result: symptoms will be solved and true causes will remain.

Rules of Tacks (Medical)

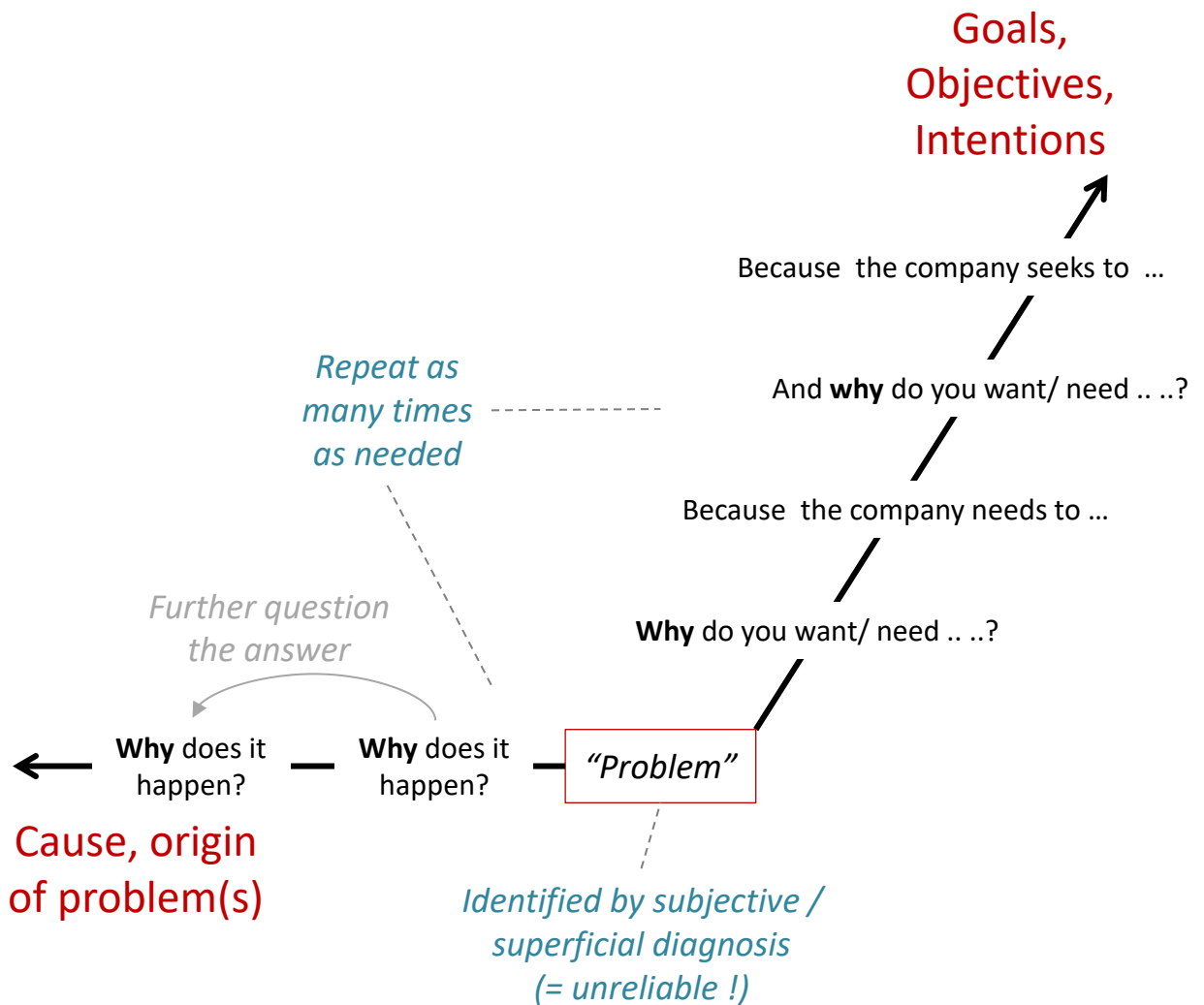
1. If you are sitting on a tack, it takes a lot of aspirin to make the pain go away.
2. If you are sitting on 2 tacks, removing one does not lead to a 50% improvement.

Syd Baker, M.D.

Accurate diagnosis is important !!

- Do not rush to build solutions solving the symptoms
- Symptoms (complains, obstacles, flawed design, late delivery, ...) are messages about underlying problems. Don't ignore these messages.
- Remove tacks where possible. Treat underlying causes.

Notes:



"We fail more often because we solve the wrong problem than because we get the wrong solution to the right problem."

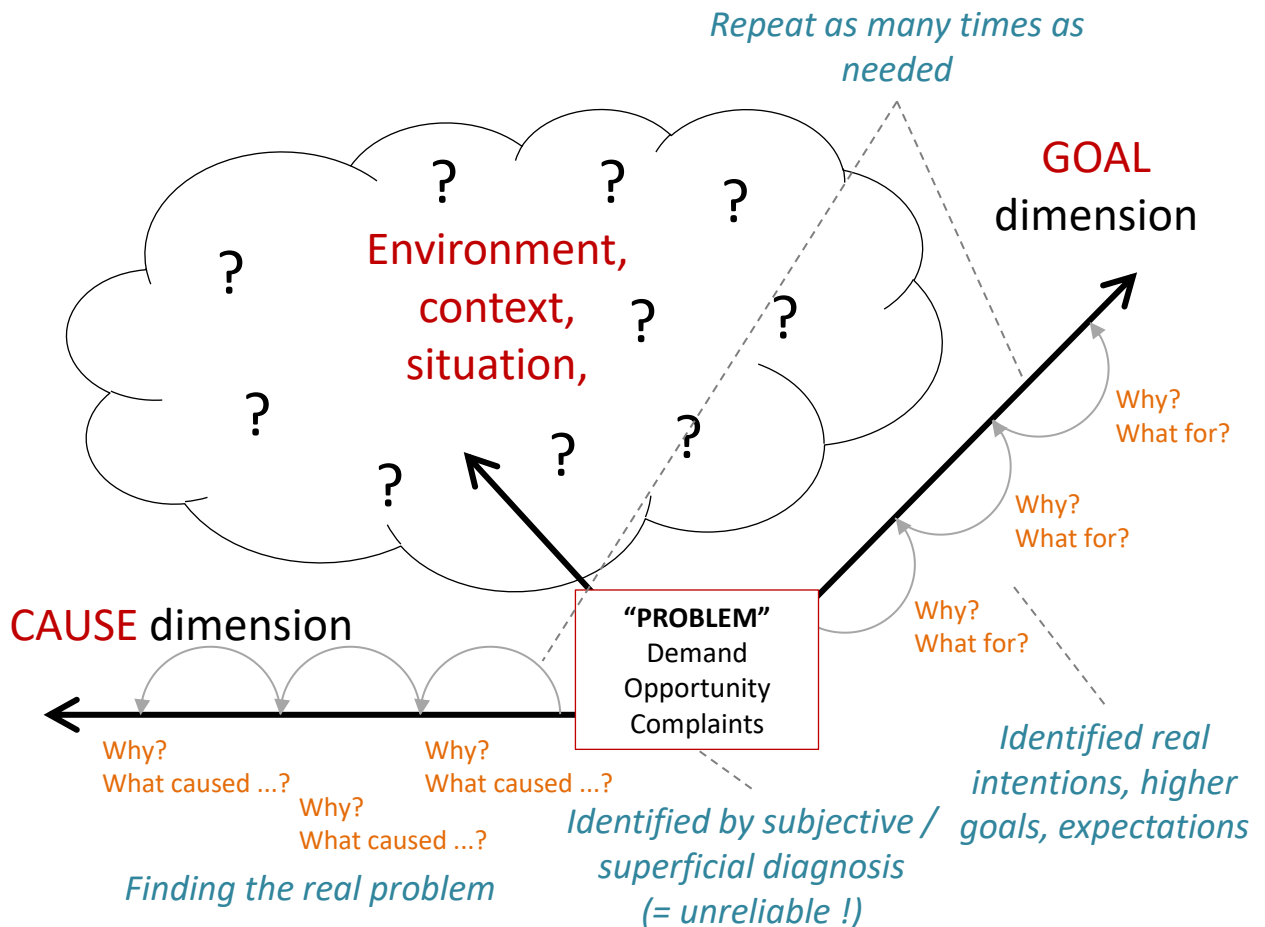
(Russell L. Ackoff)

Note:

The subjective diagnosis can be right. We don't know if it is right. Cause and Goal dimension allows to get a confirmation and a valuable broader insight, even if the subjective diagnosis is right.

Diagnosing Process

ASKING QUESTIONS



Two directions to pose questions

- Objectives
- Causes

**WHAT
and
WHY**

Never enough questions can be posed about the demand, the assertions, assumptions, context, situation, evolutions, ...

**HOW can it
be solved
rightly**

Efficient Problem Solving

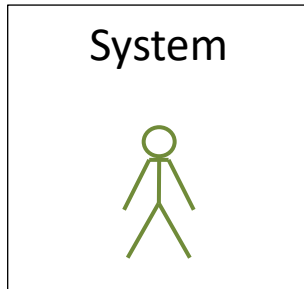
30/10/2018

During problem solving, the mind runs continuously and iteratively (without specific order) through a myriad of questions like:

- What is the essence?
- What is at stake?
- What are the boundaries?
- What do we know?
- What is the root-problem? Diagnose it!
- What is the main objective? Identify it!
- What is important? What do really matters?
- What doesn't matter?
- What is impacted?
- Where can we get information?
- What do we assume?
- How reliable is an information? Can it change?
- What can we deduce?
- What don't we know? Maximise the knowledge and insight?
- Is what we "know" true? (not an evidence at all)
- What can we change?
- What can't we change?
- What will mandatory have to change?
- What are the mandatory steps?
- What is likely to be changed?
- What are the solution elements which may be part of the solution ?
- What are the options?
- Are their different ways to the goal ?
- If this or that is changed, what is the impact? What are the consequences? What does it imply?
- Can something be more detailed?
- Can something be clarified ? Remove ambiguity.
- What is still unclear? What do we assume? What did we forget?
- ...

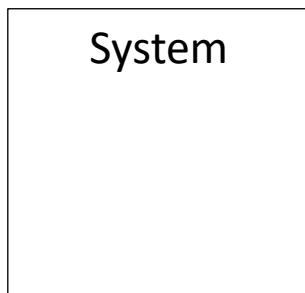
Notes:

Increasing and repeating the number of questions the mind is running through improves the problem solving skills.



A person can be a part of the system.

It is harder for a person who belongs to the system, who is inside the system to contemplate that system from a higher perspective with a higher level of thinking. Similarly, it is also harder for the system builder to contemplate the system (s)he builds to consider it with a different way of thinking used to build and or to adapt it.



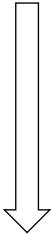
It is easier for a person outside the system and for a person who didn't build or adapt the system to consider that system with a higher level of thinking and a fresh view. Issues and opportunities may more quickly be noticed.

But knowing the whole system may take some more time to know the entire system/situation.



City Planning

Focus: Roads, industrial area, agricultural area, parks, energy supply, emergency services, religion, administrations, education, cultural aspects, shops, parkings, public transportation, sports, nature, communication network, waste water system, ...



Building of individual items of the city, such as roads, buildings, specific infrastructure, ...



Building: Purpose, size, number of rooms, foundation, shape, bearing walls, size and purpose of the rooms, doors, electricity network, water supply, ...



Design of room: floor, wall paper, furniture, decoration, illumination, ...

A building is much more than a set of rooms. A city is much more than an amount of buildings. Each level adds new specific elements and aspects.

1. Military Strategy



Plan to invade a region or country

2. Military Tactics



Plans for individual battles can only be devised when the strategy is known.

2. Military Logistics



Logistics can only be designed when the strategy is known.

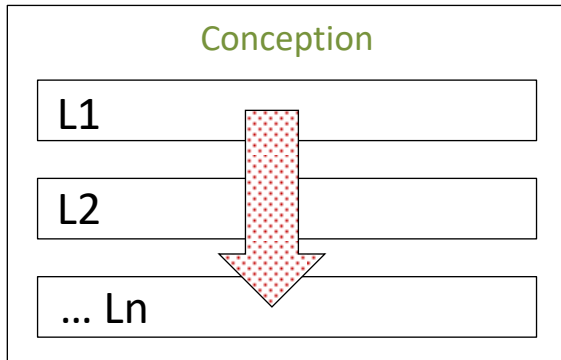
Of course, movements of enemy troops, new information, outcomes of battles, possibilities and impossibilities of logistics, and so on, can influence a strategy. I will then have to be adapted. A strategy is a dynamic plan.

Important Principles:

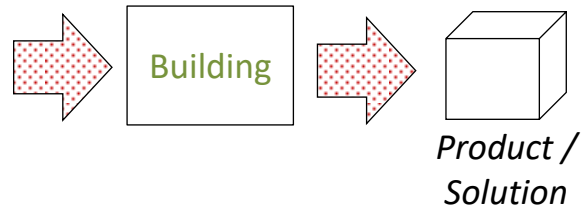
- Each layer resolves the problem at a certain level only. It solves specific questions.
- It has its own purpose, intentions, goals, perspectives, matter and elements, obstacles etc..
- The nature of each problem is very different of the problems solved at other levels.
- The problems of the higher level needs to be solved before the question of the lower level can be solved.
- Each level requires specific knowledge, competencies and requires a specific way of thinking. The knowledge, ways of thinking and competencies of one layer are inappropriate to solve questions of the other layers.

Layered Problem

14/02/2020



Layered problem \neq Top-Down !!



A **Layered Problem** is a problem that has to be solved by answering questions, by taking decisions, by dealing with issues and by conceiving a part of the solution within in subsequent layers.

One layer deals with matter of a specific nature and solves certain types of problems proper to that layer. Generally speaking, these problems can't be resolved in another layer. Each layer has its own intentions, goals, priorities, elements, aspects, principles, knowledge domain, expertise, way of thinking, ...

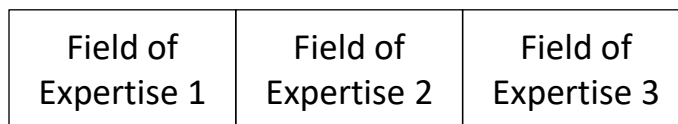
Only when the solution is conceived within one layer, it can further be conceived in the next layer.

It may happen that possibilities or impossibilities in lower layers influence the conception in a higher layer.

If a problem is layered in nature, this has to be respected in the solving approach. It is crucial to recognise this type of problems. An indication is when different fields of expertise are required.

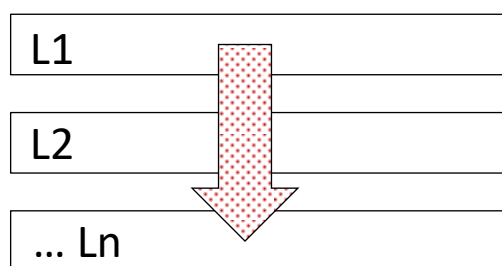
Layered Problem

Fields of expertise are at the same level

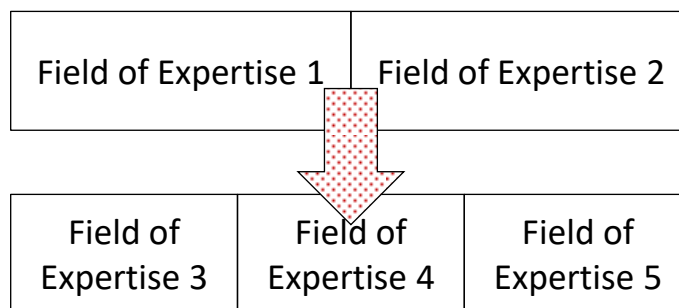


Different experts collaborate together.

Layered Problem



Layered Problem requiring different Fields of Expertise



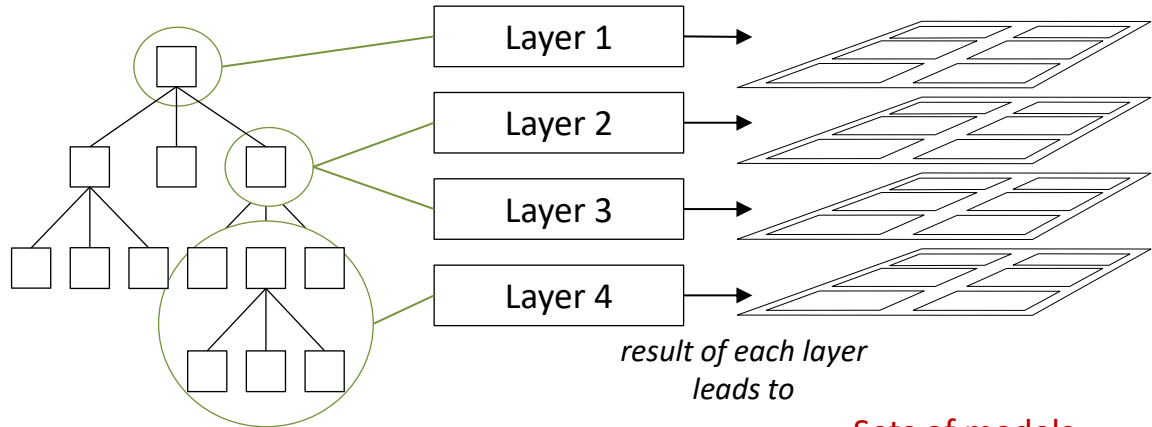
Each layer may require different fields of expertise to collaborate.

If a problem is layered in nature, then the approach and the collaboration must respect and reflect this.

Else, it's like, for the conception of a airplane, putting all the engineers and the designers of all the different necessary disciplines together in a single room in order to conceive that plane.

Layered Problem

14/02/2020

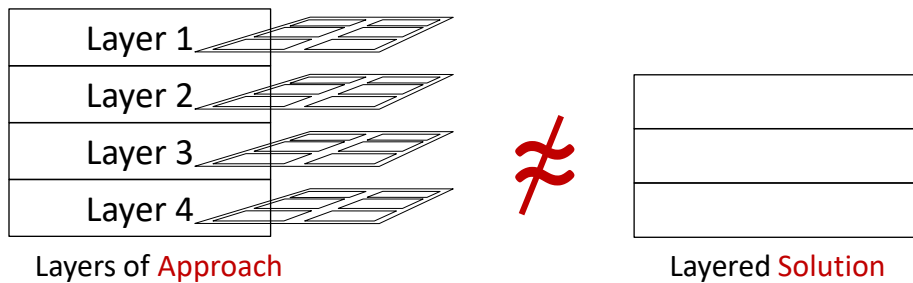


Hierarchy of **Goals**

Layered **Approach**

Sets of models
representing the
solution in that layer.

BEWARE



- A layered problem is often linked to a hierarchy of goals.
- The solution to a layered problem can be represented to layers of conceptual solutions.

For each layer, one or more models can be drawn representing the solution at that level (/ in that layer).

- A Layered Problem does not necessarily imply that the physical solution has to be organised in layers, or that these layers matches the layers of the problem.

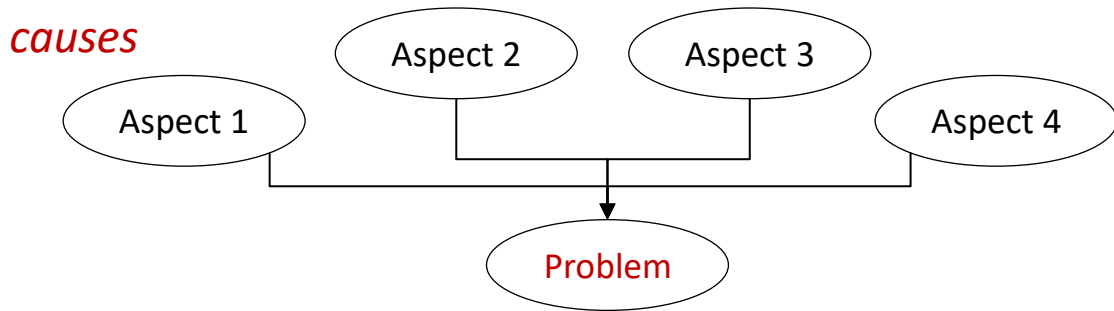
- A Layered Approach may lead to the definition of lower level goals.
- In a Top-Down approach, the matter, intention, goals, expertise, etc. may remain the same.

For example, a process that is being decomposed and further detailed.

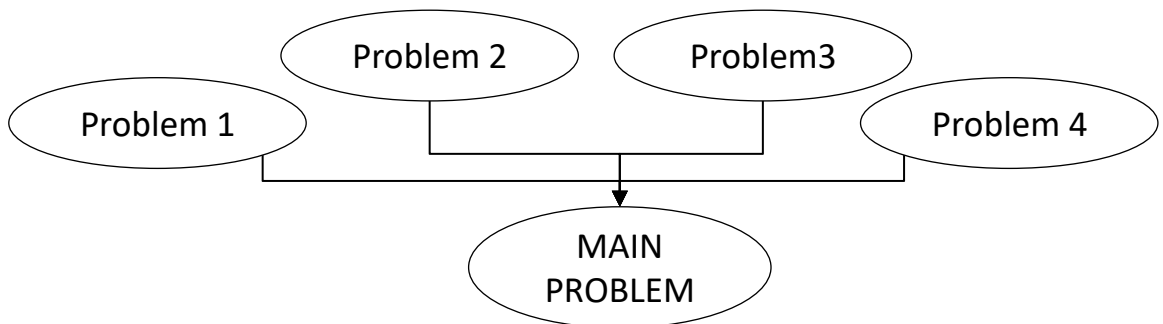
- A Top-Down Approach and a Layered Approach provide a good basis for architecture.

Multiple-Cause Problem

14/02/2020



- Different causes, often of different nature causing together a problem.
- Often, this can be seen as different elements, aspects and/or circumstances aren't right and together they cause a problem.
- They have all to be tackled in order to solve the problem.
- Its a kind of multi-problem problem.
- A **composite problem** is a problem consisting of a set of interacting sub-problems. (**composite problem analysis**)



The problem can't be solved by solving only one cause.

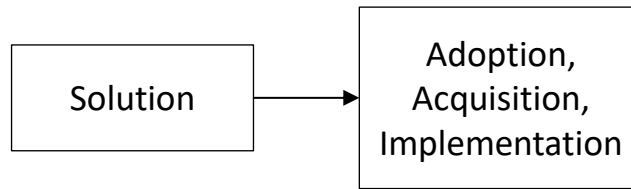
By solving problems 1 to 4, which are individual issues and may be different in nature, in time, in place, in importance, in size, ... automatically will solve the main problem.

For example:

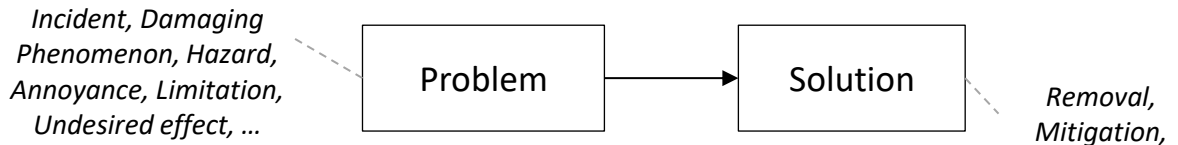
- A company isn't doing well. Marketing strategy has to be adapted. The products and services have to be improved. And the motivation of employees have to be increased and their skills adjusted and sharpened. These problems are unrelated. They are of different nature. Together, they may make the company profitable again.
- A person feels a bit depressed because of various circumstances. To feel better, (s)he can eat healthier, sleep better, go in nature, do sport, meet friends, create something, help people, be kind. (S)H may feel much more energised again.

Using Simplistic Problem Solving's Models

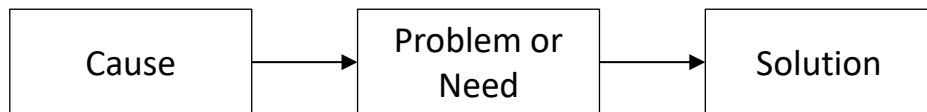
10/01/2020



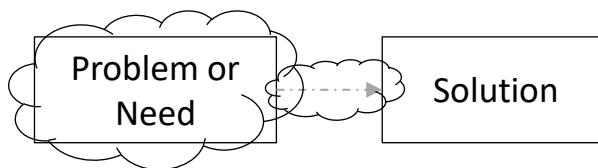
No problem or need?



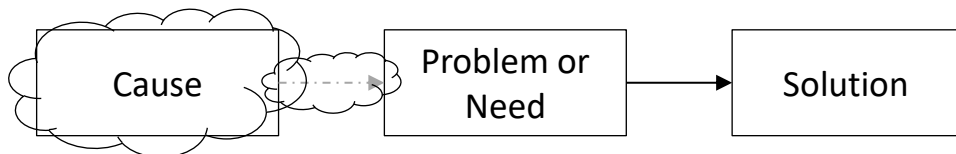
No Causes?



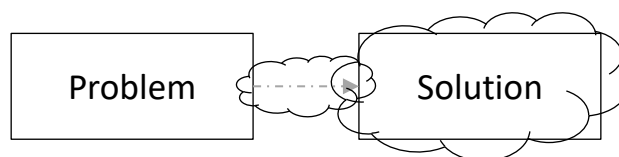
Only one cause? What caused the cause? What caused the cause of the cause ?



Vague idea of the relation between problem and solution (assumptions) and/or superficial understanding of and a lot of uncertainty about the problem or need.

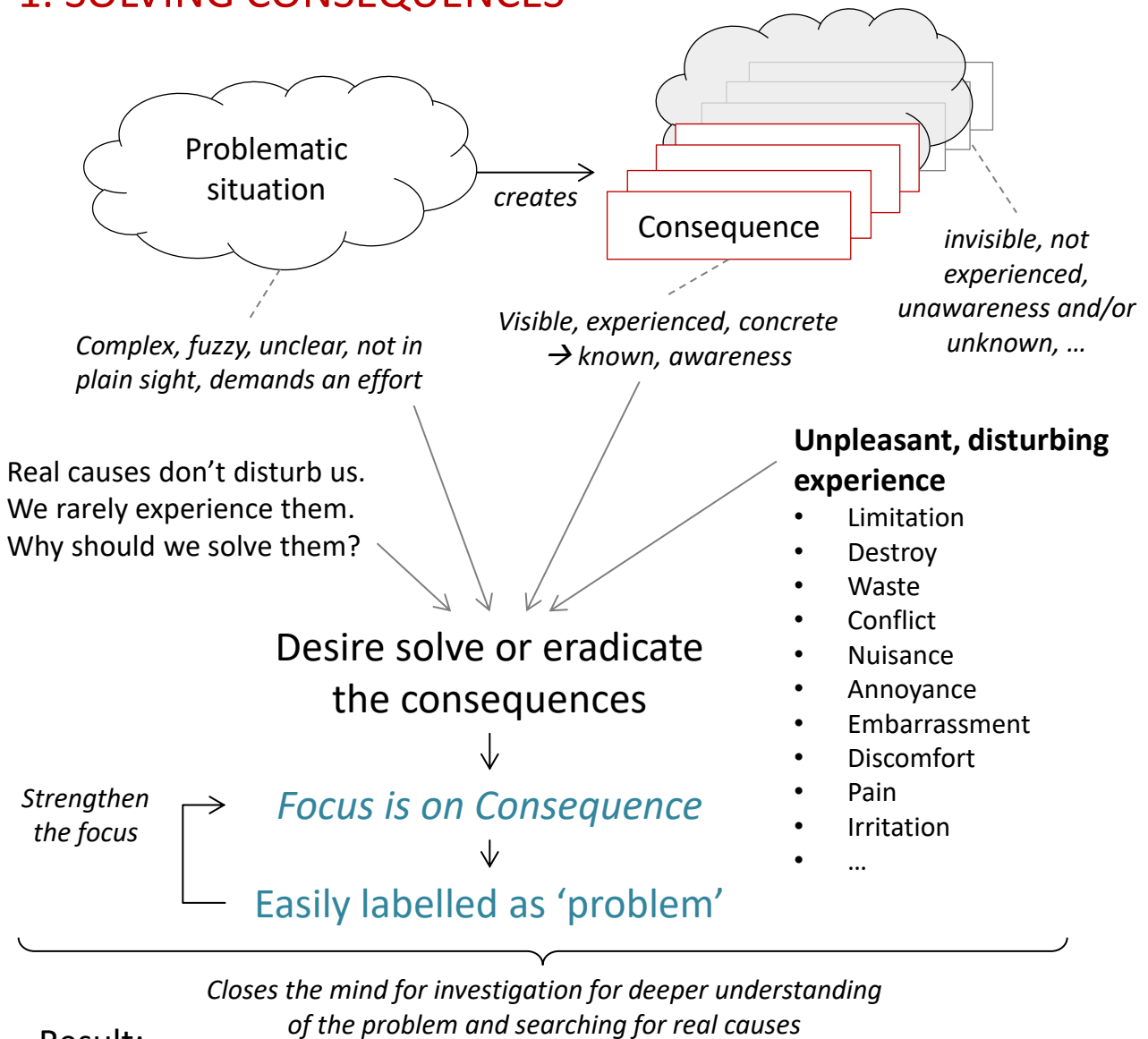


Vague idea of the relation between problem and its cause(s) (assumptions) and/or superficial understanding of and a lot of uncertainty about the cause(s).



A vague understanding and many assumptions, beliefs and hopes about how the solution will solve the problem and/or about what the solution will be.

1. SOLVING CONSEQUENCES



Result:

1) Bad solutions

2) Problematic situation (real causes) and invisible consequences remain unresolved.

3) Symptom → "solution" → Chaos

Systematically solving symptoms creates chaos



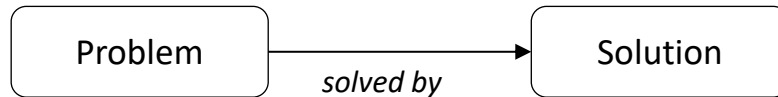
Any 'problem' should be considered by default as a consequence, until no problem can be found that created that consequence. 381

2. PROBLEM = THE MISSING SOLUTION

“The problem is that I have no umbrella”

The actual problem is:

Since it is raining, you will be wet when going outside

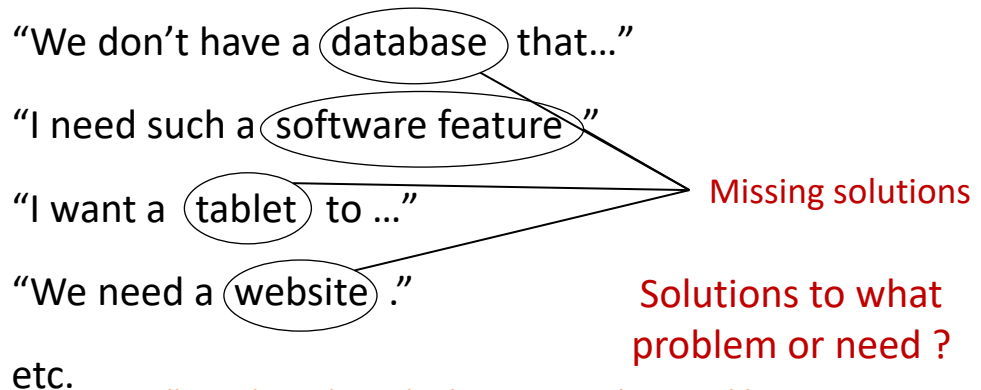


Once we have the solution, the problem disappear.

But, the problem is that the solution is missing.

Solutions: use a raincoat, find/buy an umbrella, go by car or by train, find a shelter, leave when rain stops, ...

In practice:

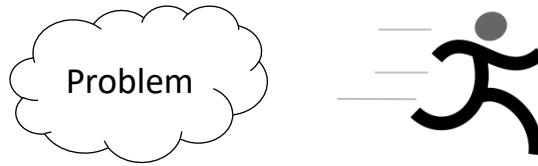


*Talking about desired solutions, not about problems.
They can be built and delivered, but will they solve the real problem?
If not, what is the created value? Obtained satisfaction?*

The business community needs information. It has a information problems and information needs.

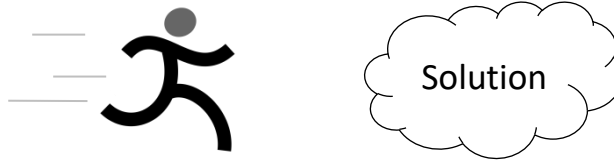
Information is needed for business activities, for decisions, ...
to achieve business objectives.

3. RUNNING AWAY FROM A PROBLEM



- Minimising the issue
- Postponing taking measures
 - “Not now because...”
 - “First we have to...”
 - “Too busy now...”
- Take some initiatives, doing some attempts
 - Not with the genuine intention to solve it (you never know), but with the idea that we did something. This brings a peace of mind. But it doesn't solve the problem.
- Avoidance:
 - Wait and do nothing. Look the other way. Ignore it.
 - Pass the problem to someone else:
 - Informing someone about the problem.
 - Making sure someone else notices the problem
 - Letting the problem grow until it becomes too big for you to handle.
 - Externalisation: privatisation, outsourcing, consulting, disinvestment, ... (getting it solved by someone else)
 - Quitting. Stop the plan, the practice, the activity.

4. RUSHING TOWARDS A SOLUTION



- without knowing or assuming knowing the problem
- without knowing or assuming to know the problem's context
- Attraction to solution: Solutions ease our fears.
- Once the solution is known, the building can start. This makes progress more tangible and visible. Many prefer 'doing' over 'thinking'.
- Common mistake: wanting a particular "solution" instead of wanting to have a problem solved. Very different in approach, talk, attitude and result !! We may end-up with a "solution" but the problem is badly or not solved.
- Assuming too quickly the understanding is sufficient
 - No proper diagnosis
 - No Proper understanding of the context, situation

"YESS! We have a Solution", "We know how to solve it", "We know what to do"

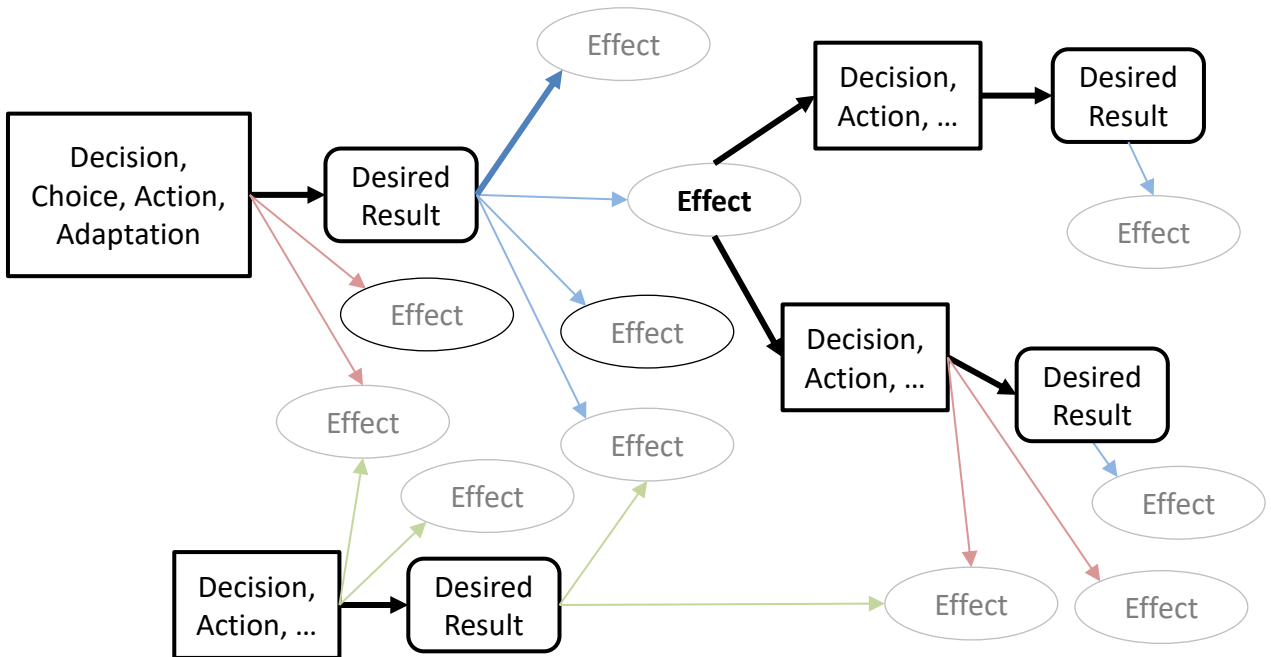
Overly focus and to early focussing on the Solution

Early engaging in searching for a solution, in solution thinking, in conception of the solution

Result: Non-solutions and more problems

Unawareness of Consequences

30/10/2018

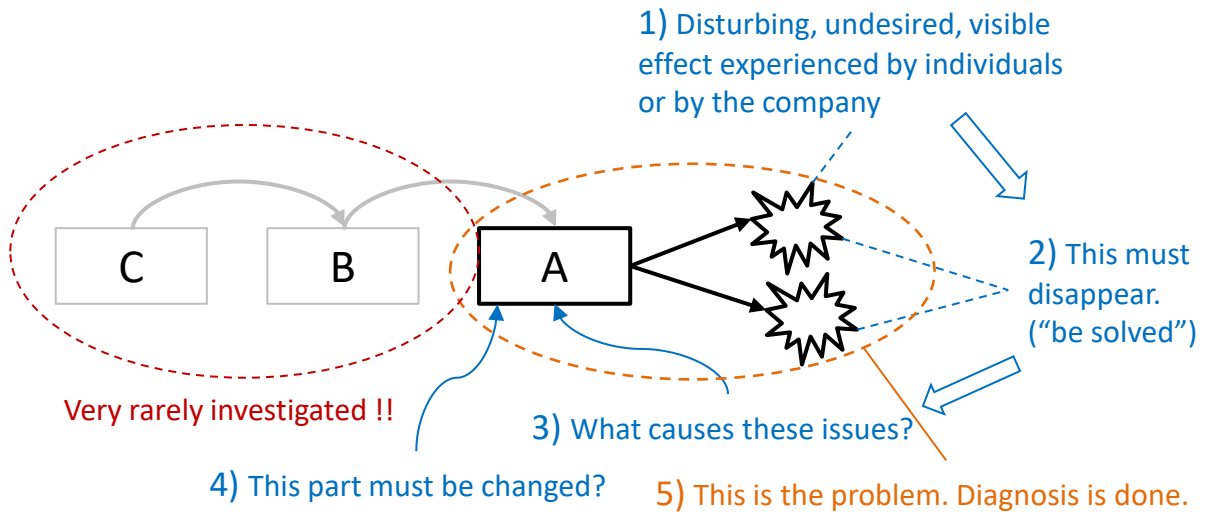


1. Decisions and actions are (sometimes) motivated by the desire to get a specific result (**bold black** = what is known and desired).
2. Attention goes to the decision, action and outcome (the intended and the real outcome).
3. Decisions and actions have effects on the system, environment, resources. These are consequences. Something that changed. (red arrows)
4. Even the simple presence of a new system or object in an environment changes this environment.
5. The obtained outcome (result) also has effects on the system, environment and resources. (blue arrows)
6. These effects (changes, consequences) may alter the capacity, the capabilities, the behaviour or other characteristics of the system. The effects may also have no further (significant, noticeable) impact on it.
7. These effects may create circumstances for an adverse event, from small to catastrophic, to happen.
8. Or, they may increase or decrease the impact of future changes and events.
9. Other effects, particularly the indirect effects are not observed. We may remain unaware of the effect. Or, we may not be aware of the link between a decision or action and some of its effect (light grey ovals).

Examples:

- Motivation or demotivation of people is the result of other factors.
- An awkward design may make maintenance or future changes on the system very costly. The effect of this additional cost is unlikely to be recognised.
- A change of colour of a car won't affect the car's characteristics. But in some circumstances the colour may, as a camouflage, make the car lesser visible *'causing'* an accident.

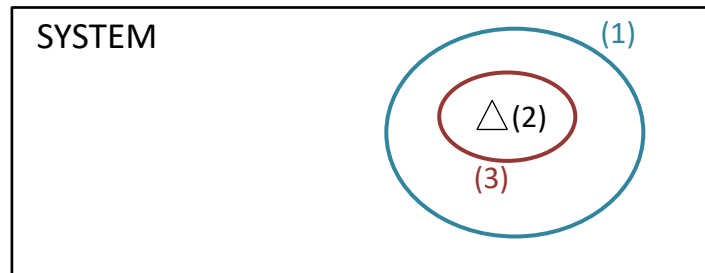
1. Limited DIAGNOSIS – Solving Consequences



'A' is seen as the cause. It produced the unwanted effects. It is not considered as a consequence. B, and a fortiori C, remain unresolved.

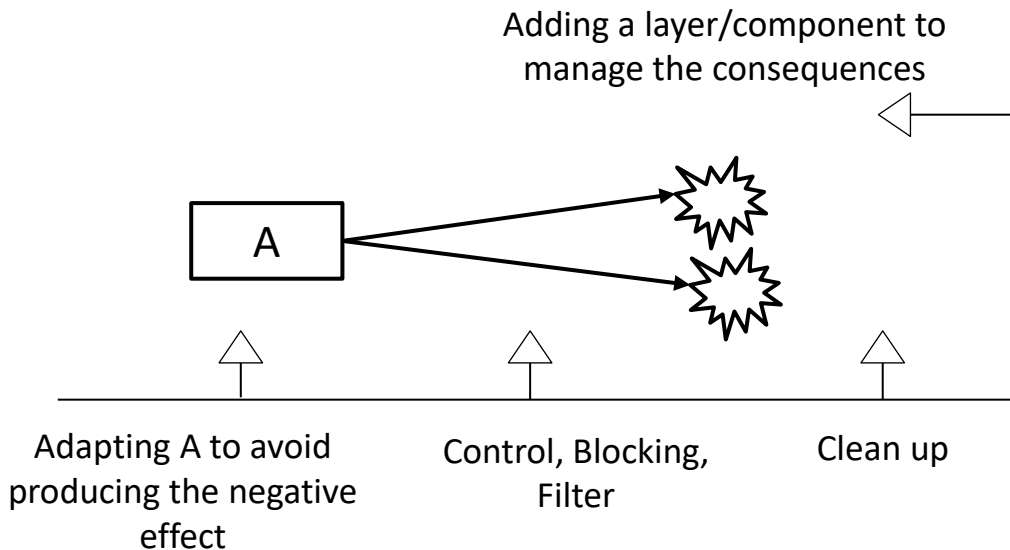
→ Not much fundamental matter is learned.

2. LOCAL SOLVING

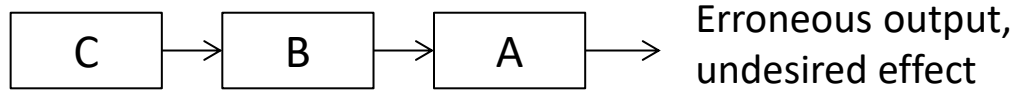


- 1) Locate the area in the system producing 'the problem'
- 2) Locate 'the problem' (from broader area to a more precise area)
- 3) Analyse that part of the system
- 4) Look how it can be adapted – come up with a few solutions
- 5) Decide how to adapt or add something to the system
- 6) Adapt the system
- 7) Test if the adaptation offers the desired behaviour

Points of intervention



Limited Diagnosis – Solving Consequences



A process (chain of steps) or a set of connected systems, or other chains of similar or physical components can be investigated to find the root cause. This makes sense to correct mistakes, but not to prevent further similar mistakes. It prevents from learning at a deeper level and to improve in a more fundamental way. Often a true crisis or a repetition of similar flaws forces us to leave this investigation path and to search for a new one.

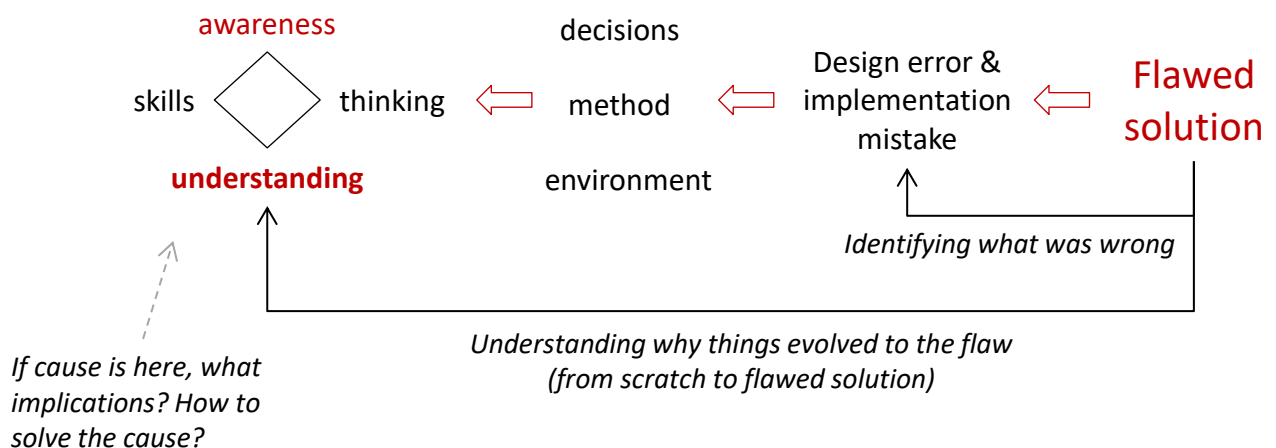
We consider only the question “where did what go wrong”. The question how we evolved to that situation, **what is missing or wrong in the design process that created the flaw is not raised.**

Finding the True Causes

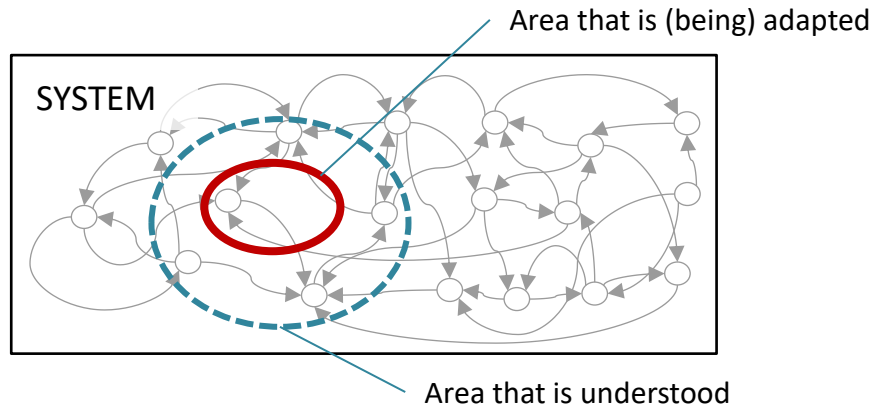
All kinds of flaws come in a design because of bad choices and decisions, incorrect methods and unsuitable work environment. Work environment include here things like collaboration, communication, mutual support as well.

The cause of unsuitable decisions, methods and environment can be traced back to the lack of skills, too little thinking, insufficient awareness and a lack of understanding. The latter one is the most important. Learning is key.

Backwards investigation (analysis) towards the root cause



Risks in Local Intervention (Local Adaptation)



Knowledge after local investigation:

- Global understanding of the system and environment
- Investigated local area
- Still many vague or unknown areas and interactions
- Requires (also) knowledge of the internal organisation of the system

Risks:

Adapting or influencing the functioning of a system or its environment, may have undesired effects (not always immediate, not always visible) due to pre-conditions, interactions, limits, relations, behavioural aspects, not taken into account (ignored)

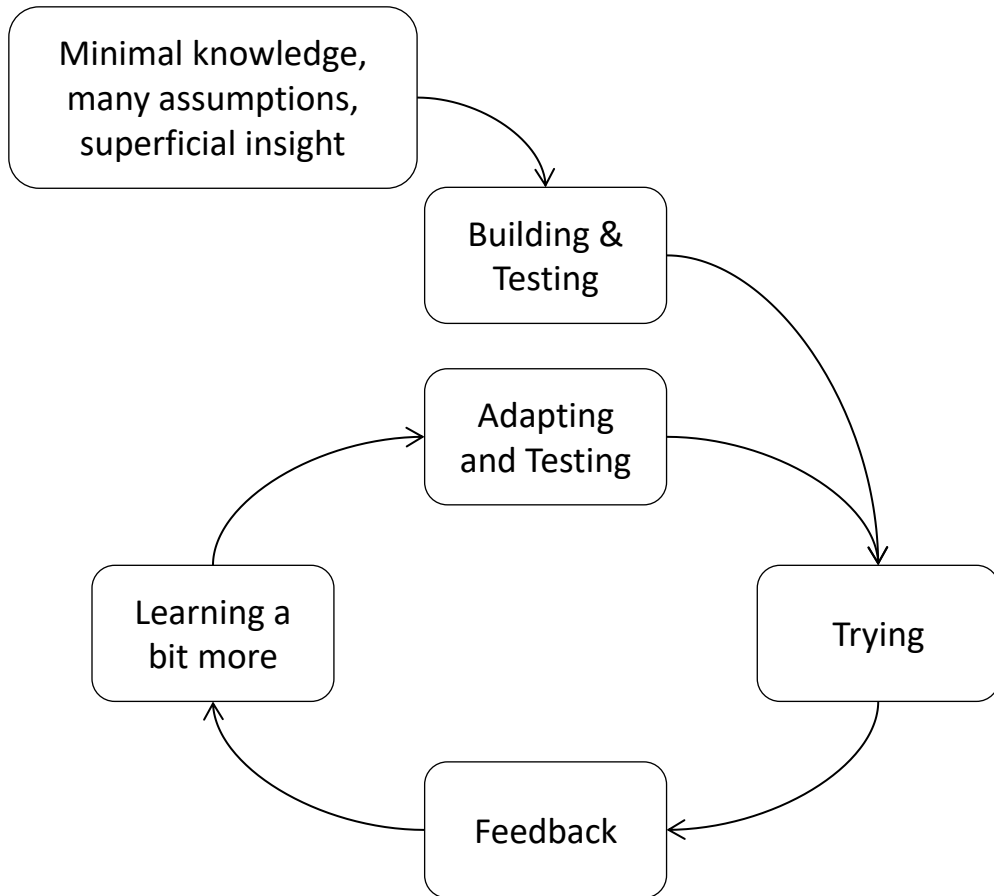
Intervening on a complex system which is not fully understood is taking a huge risk.

The more we know the system, the fewer problems will occur due to ignored interactions.

Analysis, controls and simulations may help to reduce these effects.

Mediocre Problem Solving

18/12/2018



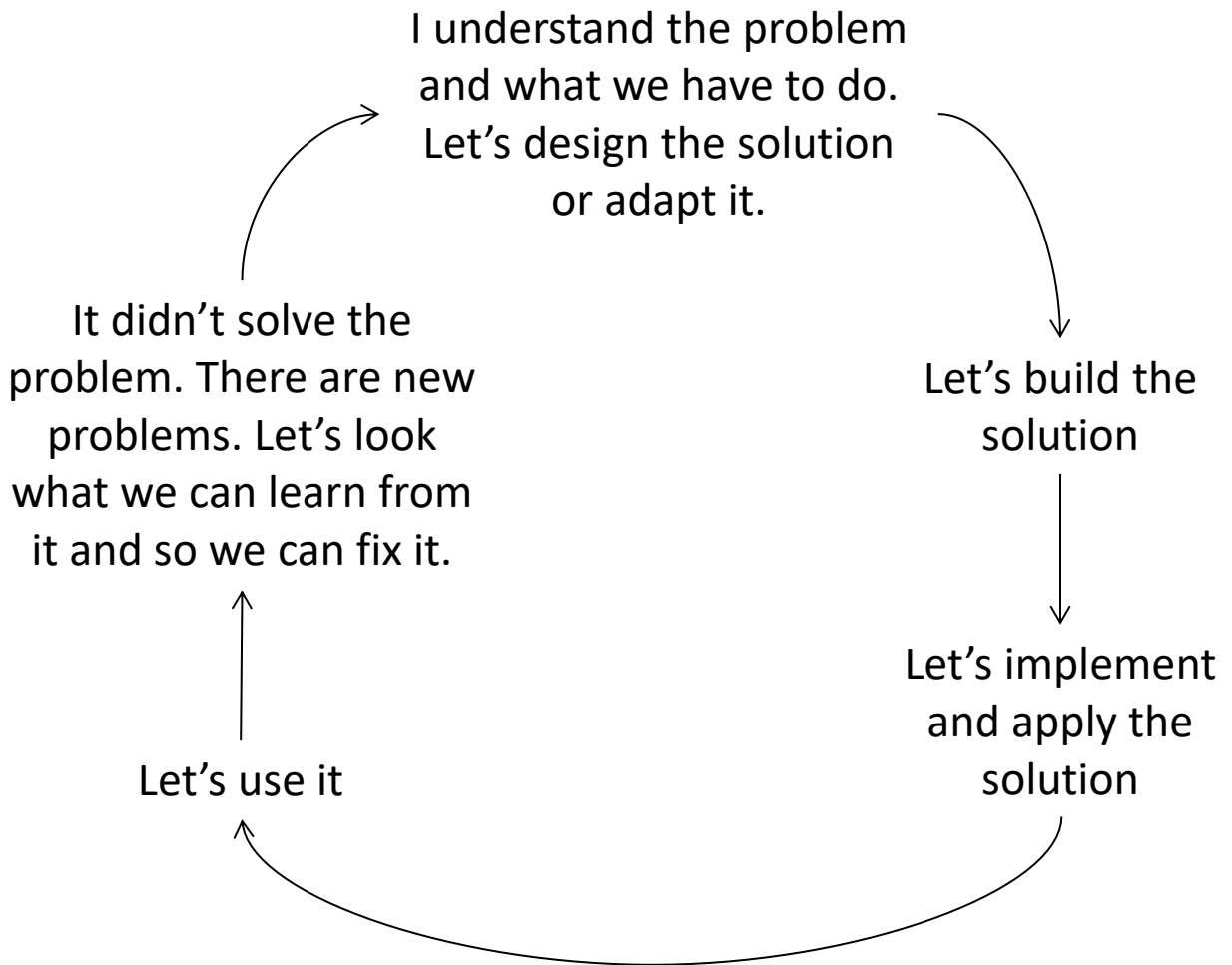
Notes :

- Useful for matters which are difficult to predict like response of people (preference, interests, ...)
- Useful for concrete, local, practical aspects like User Interfaces improvements
- Trying is good if no other alternative does exist.
- Not very useful when a broader, deeper more holistic insight is required.
- Adaption: Redo some work; some work has to be thrown away and replaced by new → includes a waste of time and resources
- What is the guarantee that what we learn is valuable and that after learning it, no other iterations will be required?
- How much do we learn compared to the time spend on building a non-solution? Is there a faster way to learn that? Can we learning by trying and by getting feedback in a system (the whole organisation) where everything is connected, but where knowledge and insight is more specific and local ?

Problem Solving Traps

30/10/2018

Rushing into problem solving



Problem Solving Traps

30/10/2018

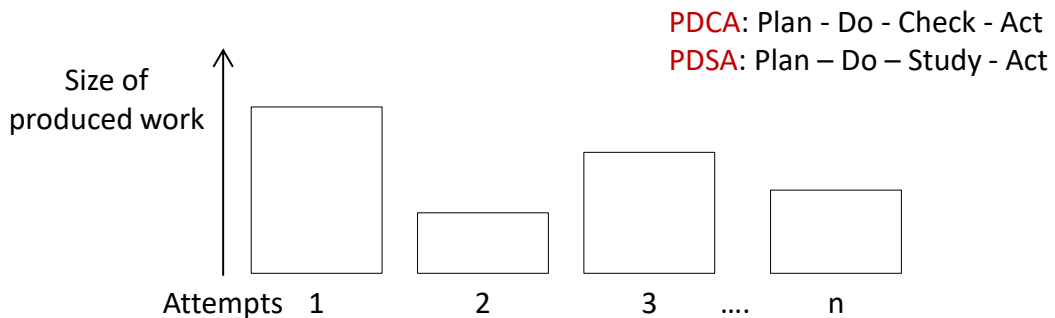
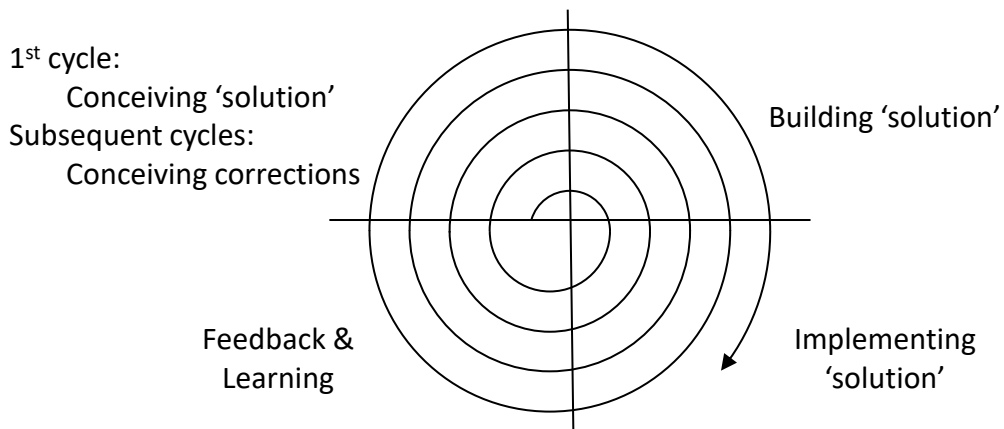
Mind is driven by

- “What solution is required?”
- “What should the solution look like?”
- “What solution will solve the problem”

Focus

- on “solution”
- on “action”

Rushing into problem solving with too little insight is not without consequences.



Problem solved after n attempts

BUT

- Will it ever be solved (certainty)? Predictability of required cycles? Predictability of timing?
- How well solved? How much rework will it take?
- If solved, then cycling until we learned what is necessary... or not even that.
- After n cycles, possible conclusion that the wrong problem is being solved
- What is learned is certainly useful. But no guarantee that we learned what we need to learn.

Learning comes late in the process.

Building solutions with superficial insight is inefficient and risky.

Limiting the Iterations

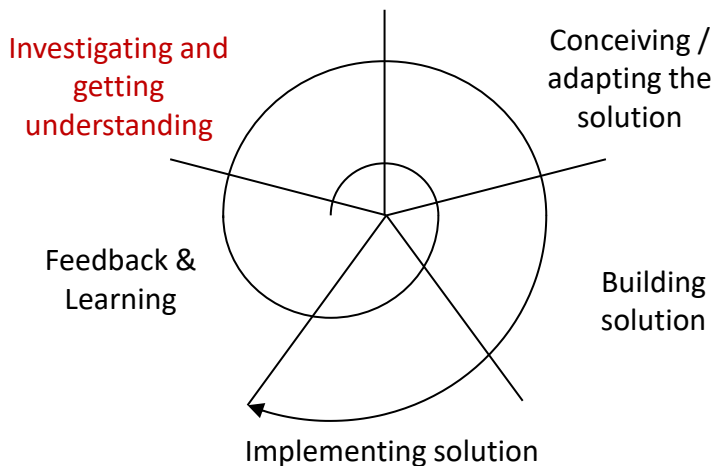
30/10/2018

Cause of number of attempts:

Lack of understanding when building

How to limit the number of attempts?

- Put learning, acquiring understanding early in the process
- and learn from the result



Two prerequisites:

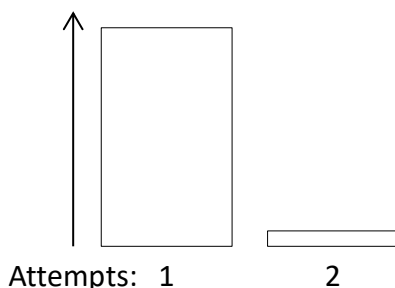
Identify the right problem

Understanding the problem, the supra-system, the problem area, the context, ...

Conceiving the right solution to solve the right problem

KEY: HUMMILITY – CURIOSITY – QUESTIONING

Size of produced work



Ensuring to tackle the right problem
Problem solved (preferably/more likely) after 1st iteration.
Ensuring problem is correctly solved.
Doesn't exclude (minor) corrections or future adaptations.

- Confusing the “understanding of documents, of demands, of requirements” with “the understanding of the situation”
- Confusing ‘knowing’ with ‘understanding’
- Not identifying the unknowns & the assumptions; not verifying the assumptions; filling in the blanks with own answers; guessing
- Choosing instead of deciding (based on unimportant aspects; based on an interpretation, assumptions and preferences; taking uninformed / ill-informed decisions)
The choice based on preference (like for taste, smell, colour, fashion, meals, deserts, ...) is fundamental different from decisions about systems, solutions, and so on.
- Oversimplification (= ignoring some existing complexity)
- The brain doesn’t like complexity and effort. This is why a simple but wrong solution (or idea, concept, ...) has far more chances to be accepted and to be spread than a complex but right one. Anything, like abstract thinking, requiring an effort of the brain is unpopular and everything requiring no effort is popular (neuroscience).
- Avoidance by externalisation (= avoiding to have to deal with existing complexity). Avoidance is a missing chance of strengthening the own skills.
- Going along with the first answer, accepting the first solution (fear disappear as soon as we have a solution. Proposed alternatives creates a new problem (which one to chose, which one is better, new discussion, ... a new effort for the brain)
- Fake alternatives: Often a set of alternatives are proposed. Only one of them is / looks like a real solution (which is the one someone wants to sell), the other “solutions” are obviously worst. This is a fake choice among fake alternatives.
- People often are not aware of what they need. They confuse what they want and what they need.
- People may define the missing solution as being the need, instead of what it is supposed to solve. “We need a new database. So, the missing database is the need.”
- Believing the labels: “solution”, “business”, ... Accepting anything that has been labelled as “solution” or which just looks like bringing progress and is related to the problem.
- In a choice between the truth or conforming with the choice or idea of the group, people will often chose for the latter. Choosing for an answer (the truth) different of that of the group may lead to exclusion, rejection (survival strategy). Result: weakest ‘solution’, lesser innovation, ...
- “We have acquired a solution. Let’s now look how we can solve the problem with the solution. (no understanding of the problem, no cause identified, ...)

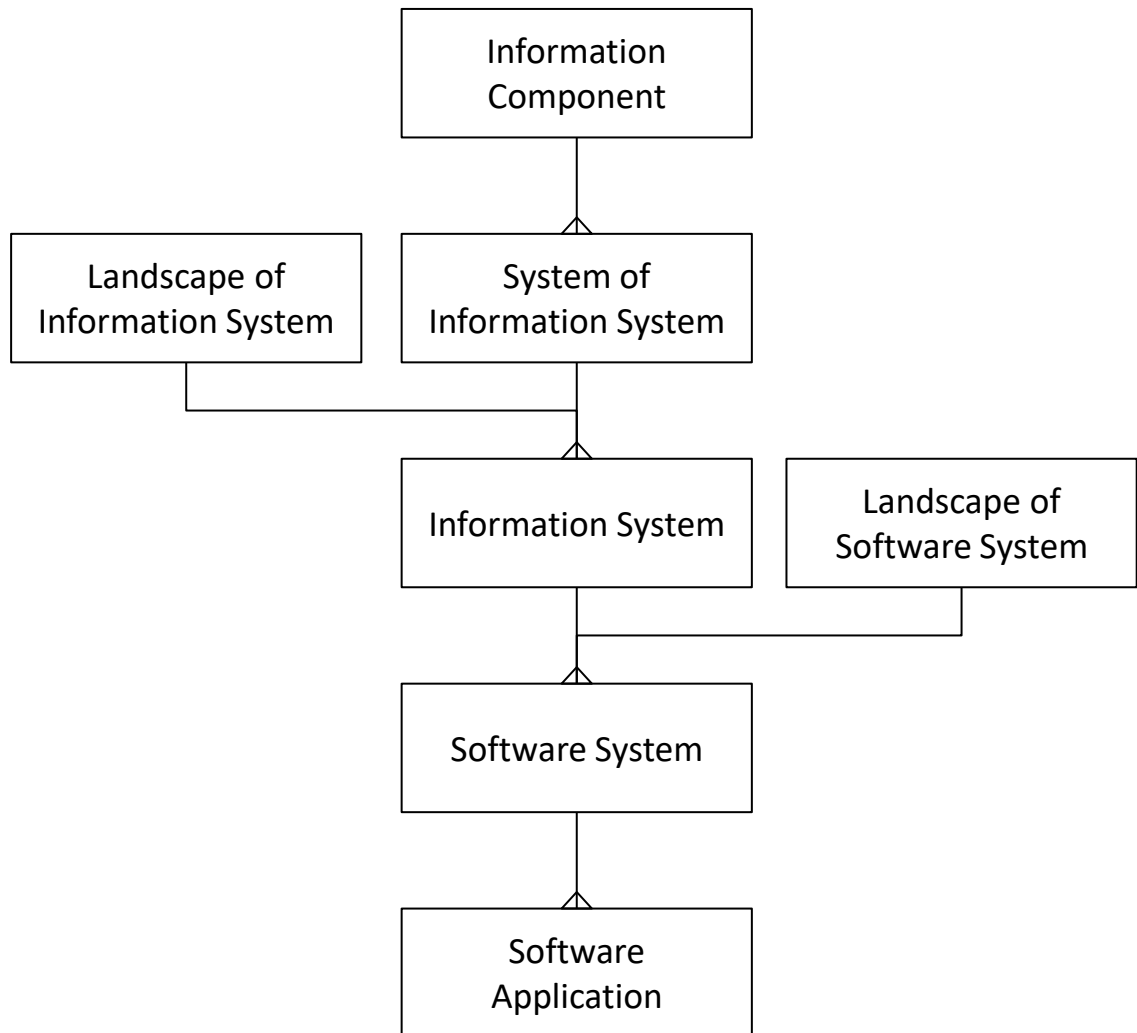


INFORMATION COMPONENT

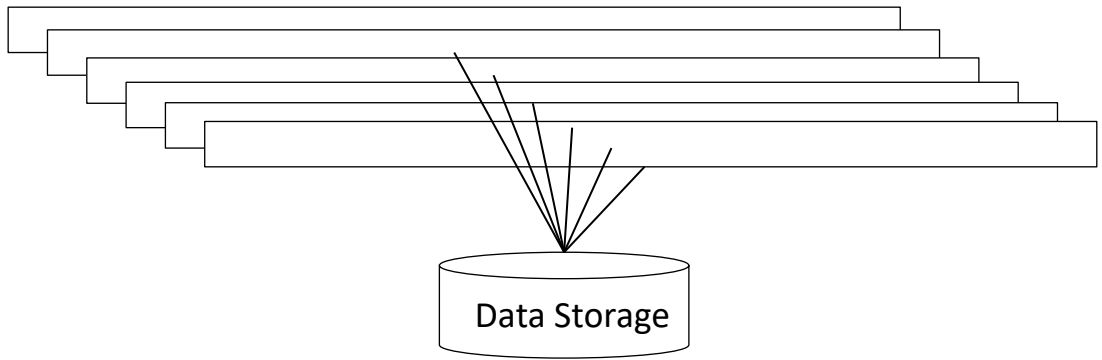
Systems in Information Component

30/10/2018

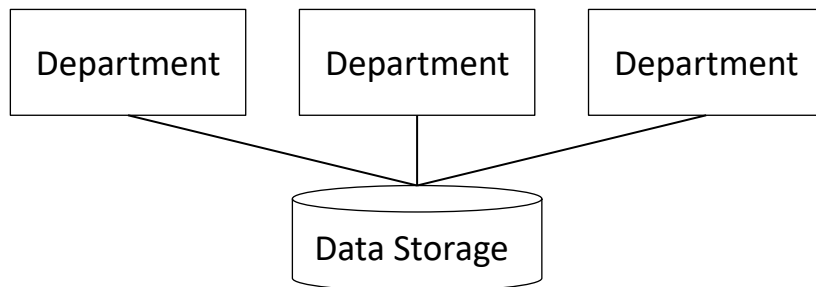
The information Component is that part of the organisation that organises, regulate and formalises information and the way information is processed and managed.



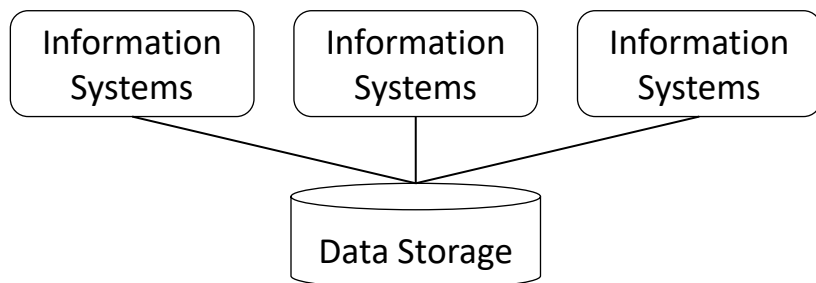
Business Processes



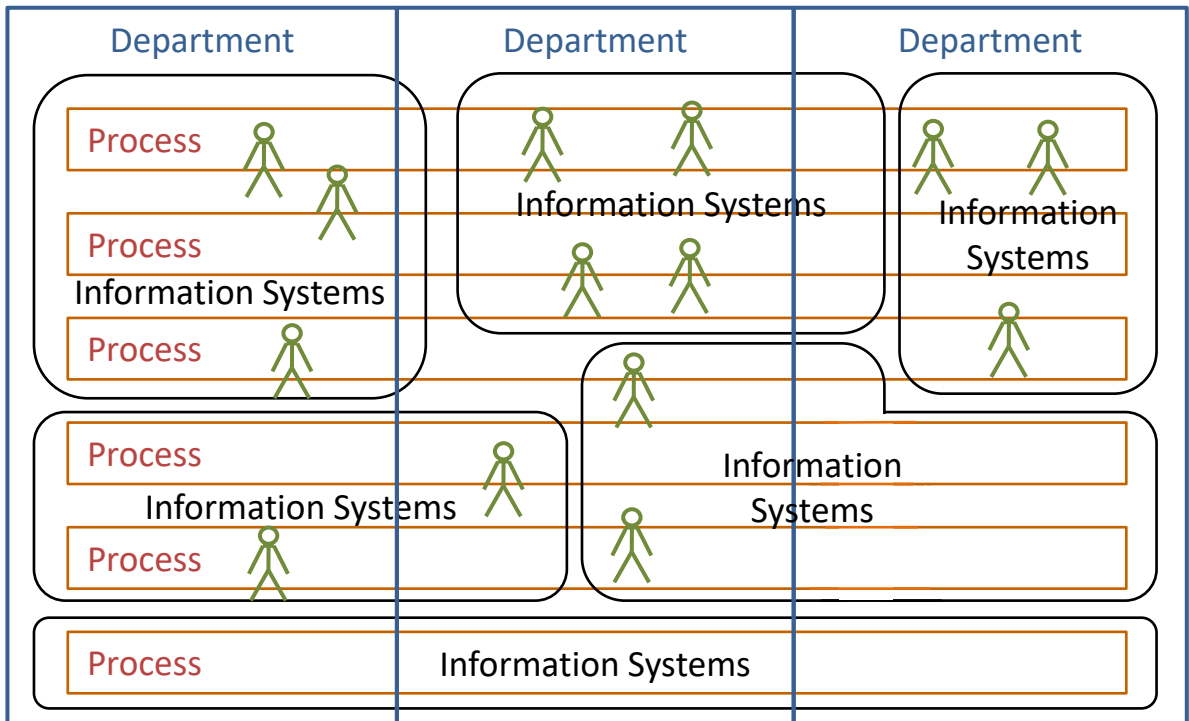
Departments



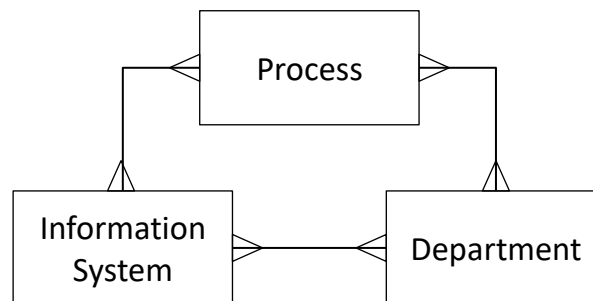
Information Systems



The IC: a heterogeneous multi-dimensional system



- Roles
- + Purposes
- + Collaborative relations
- ...
- + Communication channels



Notes:

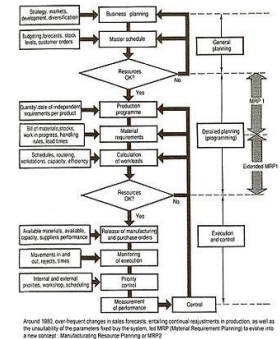
Indicating all parts, motives, aspects, dynamics in a single model would make it too complex. However, it gives a general idea, although it is still simplified.

This forms a whole in which harmony has to reign.

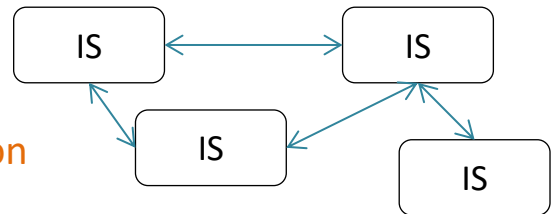
INFORMATION SYSTEM



Information and
Communication tools



Processes and
Procedures



Communication
between IS's



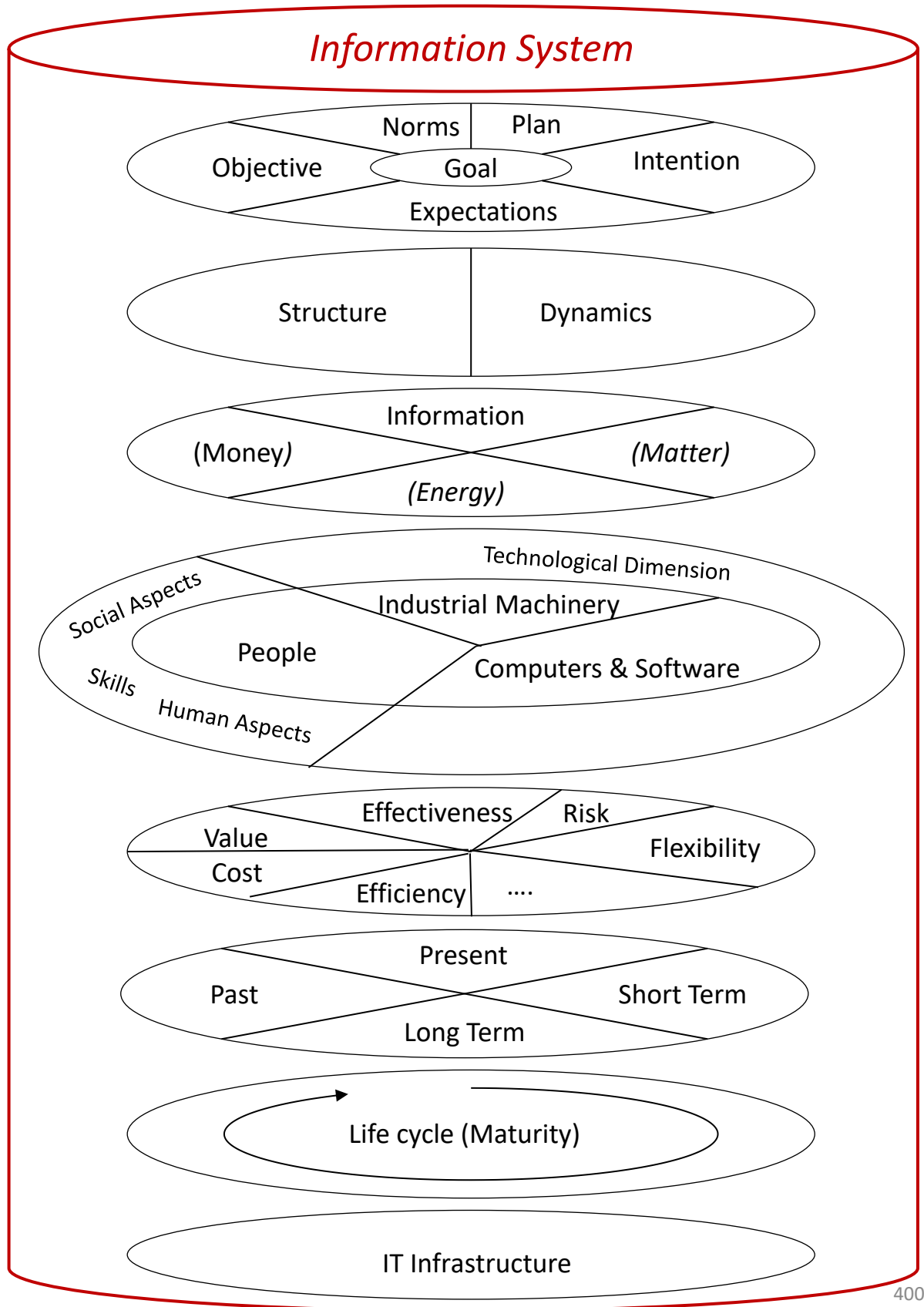
Users



Documentation

SOFTWARE SYSTEM(S)

Today, the focus is essentially on engineering software systems. Engineering the whole Information System is rarely considered.



‘Entropy’ is here defined as “the gradual decline into disorder of systems or societies”.

(Remark: different meanings and definitions do exist for the term ‘entropy’)

Information systems (and software systems) are subjected to the tendency of entropy:

1. Degradation of Design

Wrong abstraction have been used, changes of purpose, inelegant design, wrong concepts, ...

2. “Degradation” of source code

Bad programming: elegance of code; organisation of code; containing inefficient code; unused remains of old code cluttering the source code; inappropriate object, variable, function and method names; ...

3. Information contained in the system

Unreliable information, not updated, incomplete, encoded differently, not respecting standards and formats, old information, abuses of the system, ...

4. Technological fragmentation and chaos

5. Increased misalignment with co-systems, supra-system(s) and environments

Reflection:

“How do you combat entropy in Information Systems?”

Find as many principles, approaches, techniques, and other ways as possible.

- Mainframe System
- Client-Server System
- Single/multi-User System
- 2, 3, Multi-tier System
- (Near) Real-time System
- Batch systems
- Distributed System
- Networked System
- Agent-based System
- Web-based System
- Intelligent System
- Embedded System
- Back-End / Front-End System
- Enterprise-Wide System
- Communication System
- Centralised / Decentralised System
- End-User Software
- Operating System
- Cloud System
- Workflow System, Routing Systems
- Messaging Systems
- Peer-to-peer System
- Pilot system, Prototype System, Test System
- ...
- Executive Information/Support Systems (EIS)
- Management Information Systems (MIS)
- Decision Support Systems (DSS)
- Knowledge Management Systems (KMS)
- Expert Systems
- Content Management System (CMS)
- Operations System
- Office Automation Systems (OAS)
- Transaction Processing Systems (TPS)
- Supply Chain Management (SCM)
- Customer Relationship Management (CRM)
- Data Warehouse System (DWH)
- Enterprise Resource Planning (ERP)
- Just-in-Time Systems (JIT)
- Monitoring Systems
- Alarm Systems
- Financial, Accounting, Inventory, Human Resource, Logistics, ... Systems, Electronic Health Record System, GIS, ...

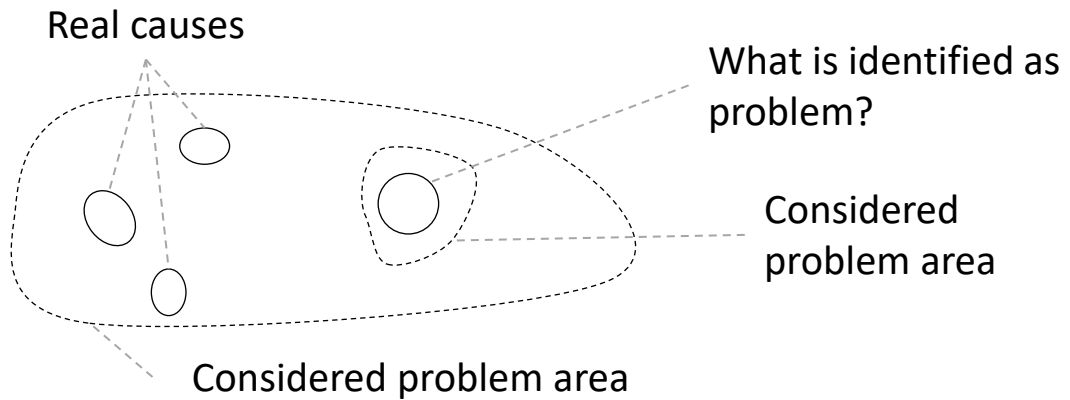


BUSINESS KNOWLEDGE & EXPERTISE

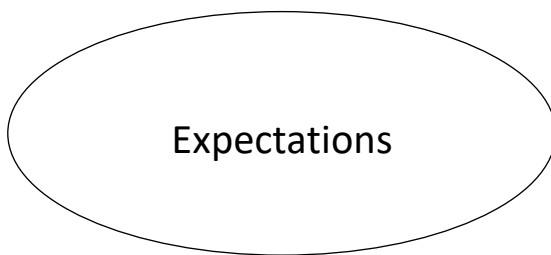
Common Issues in Contemporary IT

10/01/2020

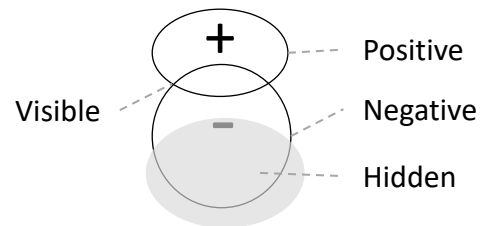
Problem Definition



Expectations versus Results



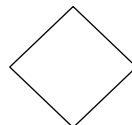
With this solution a lot will be solved and the business will function much better.



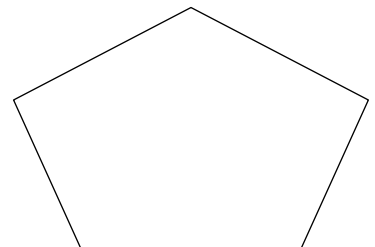
Demand, Necessity and Ignored Opportunity



The Demand



The minimal necessity

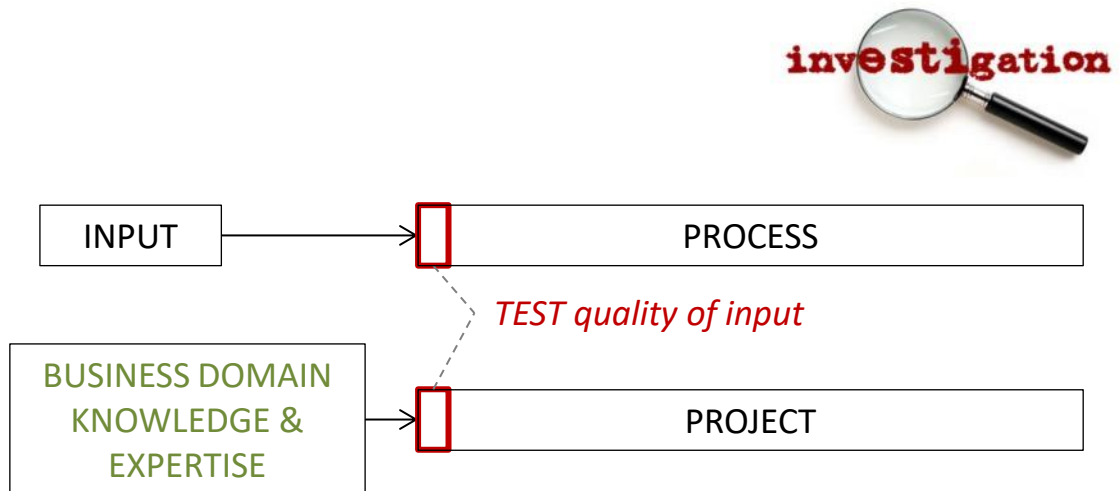


What is possible and advantageous

Let's investigate some causes

Business Domain Knowledge and Expertise

Business Domain Knowledge (BDK), Business Knowledge and Business Expertise (BKE) is **critical**. It is the **INPUT** of the project.



Need for understanding “Business Knowledge and Expertise”

- What is BDK, BKE, ...? And what doesn't belong to these fields?
- What level and quality of this knowledge and expertise is required by the project?
- What is the quality of the available business knowledge?
- What is the quality of the source?
- What are the limits of competencies of Business People in relation to systems development?

Assumptions about Business Knowledge

10/10/2018

In general, nobody knows the business better than the business experts.

Assumption 1

Business experts know the business very well / well enough

Assumption 2

Business experts are the only source of business knowledge and expertise

Assumption 3

The business knowledge is sufficient and readily usable for information systems development

Assumption 4

Business experts are able to estimate what business knowledge the Analysts and other IS engineers need to know, when they need to know it and, more importantly, what they don't need to know (filter). They are rightly controlling the flow of knowledge to the IS Engineers.

Assumption 5

Business experts are able to communicate all the required knowledge (completely, unambiguously, degree of certainty, guesses in a methodical way, ...)

Assumption 6

Business logic has to be determined first. This creates a first global picture of the solution. Non-business logic can be determined later and added to the solution.

Organisations, approaches, collaboration and decisions are based upon these assumptions. We can't simply continue with assumptions. They must be investigated, clarified and checked to know whether they are true, to what extent, and so on.

BUSINESS DOMAINS	
A) Industry Car manufacturing Car Leasing Telecom Railway Travel Agency Logistics Building Company Insurance ...	B) Business Function Management Marketing & Sales Research and Development Operations Customer Support Administration ... Financial Resources Human Resources
C) Academic Fields: management, mathematics, psychology, operations research, economics, ...	

For every statement (or decision) it is important to understand to which discipline it belongs.

Who captures information, where and how it is captured, where it is stored, how it is processed (partly), what information is kept, what information is measured, how it is shared, ... all this doesn't change how business is conducted. It is not business domain. It's informatics.

Note:

EXERCISE (useful!): For some industries and for every business function think of statements which do belong to the industry or business function and also to which they don't. This helps in identifying the limits of these areas. It will help to identify when an SME in one area talks outside his/her area of expertise.

Beware of labelling any logic other than user interface management or data writing or retrieval logic as "business logic" !!

Business Knowledge vs Non-Business Knowledge

Examples of **Business Knowledge** and Expertise

- Everywhere we do business we use as much as possible local products and raw materials and do business and collaborate with local companies.
- No ship shall enter or leave the port without the permission of harbour master's office. This regulation does not apply to tug boats and pilot vessels while conducting ships entering or leaving the port.
- New orders above the 10k € have to be validated by a manager if the client has still overdue invoices.
- Our company sells standardised items and customised items. Customised items require meeting the client in person and discussing the customisation before any order is placed.
- In dealing with immigration, the first step in the administrative process is to identify the person.

Examples of **Real World** knowledge

- A family is a group consisting of at least two persons and can be described as one or two parents and their children living together as a unit.
- Every city is located in a single country.

Examples of **Informatics**

- The client information has to be sent to the account manager.
- The clerk has to capture the information.
- This information will be stored, but we won't keep that information.
- We need to be able to search for these orders with these criteria in the database.
- This feature needs to do this and that.
- The resulting features have to appear in this screen.

WHAT BUSINESS KNOWLEDGE PROJECTS NEED



SME

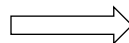
- 100% of the Business Knowledge and Expertise
- Multiple business domains knowledge
- Enterprise-wide knowledge
- Multi-level: from operational to strategic
- Relevant
- Complete
- Coherency between predefined logic and applied logic
- Clear, Unambiguous
- Detailed
- 100% certainty
- Structured, organised
- Mainstream and exceptions
- Realistic, applicable, achievable
- Settled in time: difference between what is valid now, what is valid later.
- About “As-Is” and “To Be”
- Coherence among the stakeholders
- Immediate (right) definitive decisions (no guesses, trials, partial, conflicting, ...)
- Accessible
- On time
- Stable
- ...

REQUIRED?

YES

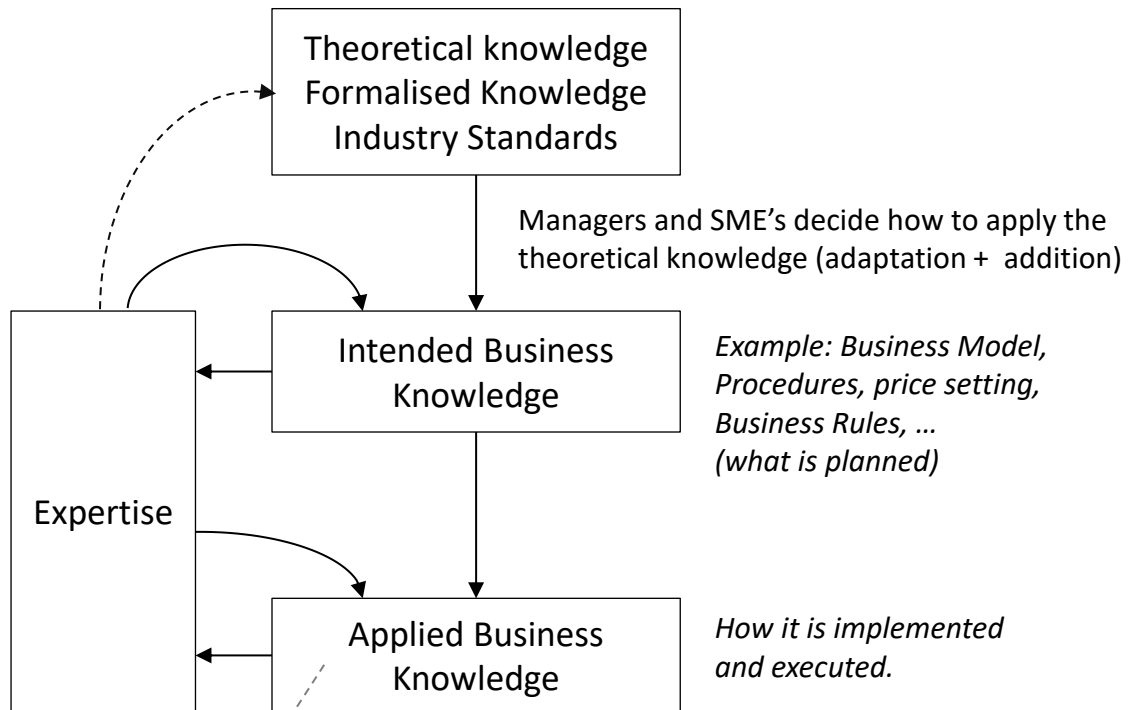
REALISTIC?

NO



PROBLEM!

(= job of the Analyst !!)



In practice, the intended business knowledge to be applied is adapted and expanded

- Unforeseen events, situations and exceptions happen
- Problems have to be solved
- People can improvise to cope with lack of details
- Learning path
- Interpretation of the intended business
- Memory (knowledge is variable)
- Not all individuals do the job the same way
- Some degree of freedom in the application. (Example: determined price isn't used, but is negotiated with the client)

Proposed definitions:

Business Domain Knowledge: Academic field, business theory, generic knowledge that can be used by different companies as a foundation.

Business Knowledge: The specific BDK and business logic implemented in the company.

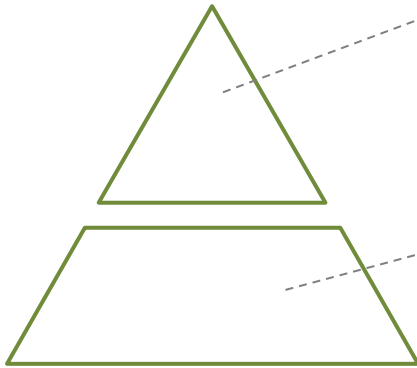
Business Expertise: Knowledge and experience in how the company conducts business and how it turns out in practice

Business Knowledge and Expertise

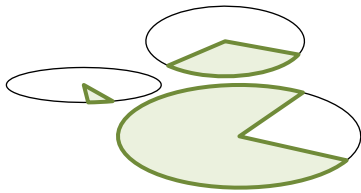
30/10/2018

Managerial level:

The knowledge at this level is broader, global and may concern the past, present and the future. Knowledge of objectives, intentions, plans, policies, ... Knowledge of the details and practical experience, as needed for practical execution, can be fuzzy, unreliable or missing.

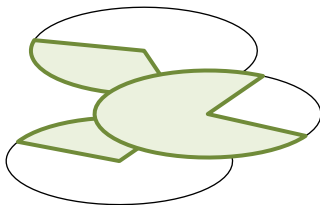


Operational level: The knowledge is detailed and practical. Operational objectives, habits, exceptions, concrete gaps, hindrances, local opportunities for improvements, ...



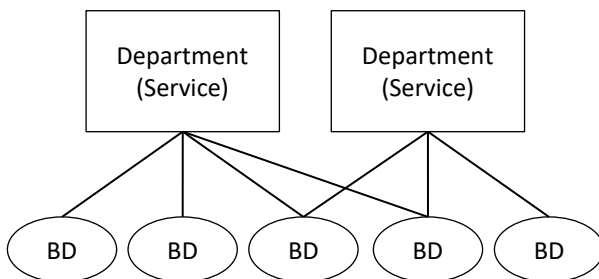
Services, departments

An employee or SME knows his role fairly well. The own service or department is also known. However, this knowledge may be incomplete. Knowledge of other roles, services and departments becomes much more superficial and uncertain. Only the information usage and the parts of the business processes in the own department or service is known. A more global picture can be lacking.



Business domains

A business subject matter expert has expertise in one or in several domains. Business domains often overlap. The business subject matter expert may have some knowledge of other areas as well. His/her expertise in other domains may be lesser. Being an expert is not a generic qualification. Someone can be an expert in an area, but not in other areas.

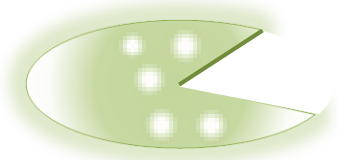


Usually, each organisational unit has (0?,) 1 or more main business domains.

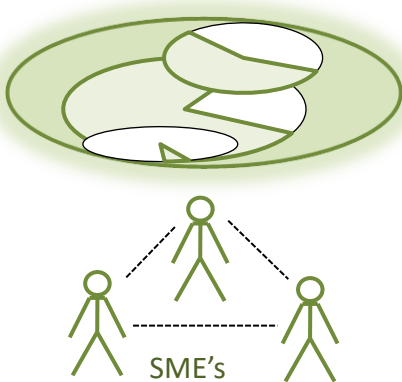
There is not necessarily a 1-to-1 relation between a department or service and the applied business domains. A department may use this knowledge for its own purpose, for a global goal or to suit the needs of other departments.

Business Knowledge and Expertise

30/10/2018



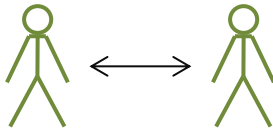
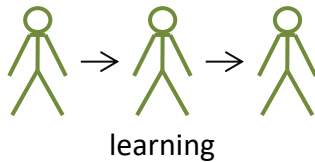
An SME has knowledge of his business domain. Some of this knowledge is well-mastered. **“Knowledge”** (and expertise) is not collection clear, nicely delimited, concrete, factual amount of information. An SME knows some areas of his domains. There might be blind spots, vague areas, unknown areas, distorted knowledge and so on. Knowledge is not black or white, true or false. Many qualities of knowledge is a matter of gradation. Additionally, there is a difference between knowledge and understanding. It is also a matter of perception, perspective and belief. Some knowledge is not always consciously present. Some knowledge can be inferred. An SME can always correct, improve and expand his expertise in his domain. Knowledge is not a static fact. It may increase or decrease. Memory is volatile.



A company has usually different persons with business domain knowledge. So, there is more business knowledge present in the company than there is in the head of a single SME of that company. The business knowledge of each SME of the company varies. They have a lot of knowledge in common. But there might be nuances, gaps and contradictions. SME's may have a different knowledge, a different understanding, a different perspective on a domain or issue. SME's consult each other to exchange knowledge. They don't always agree with each other. The application of business knowledge is often a combination of business knowledge and knowledge of the daily world (the reality).



There is more business knowledge or knowledge useful for a particular domain in the world (universities, libraries, web, companies, organisations, other experts) than present in the company. The business knowledge outside the company, although useful, is not necessarily the knowledge the company uses.



The function or term of “Expert” is not a measure of solid business knowledge. Some SME’s do a genuine educational effort to improve their expertise. They are keen to select the sources they learn from. However, a common learning tactic is to learn on the field. The SME learns from colleagues. These colleagues learned themselves from other colleagues. Activities, work approaches and methods are transmitted. Belief systems and habits are perpetuated in the company. Norms and expectations aren’t raised. They are used as objectives. So, some tend to be satisfied with somewhat better than average, being average or not so bad. The knowledge of the expert may not go beyond the functioning of the own company and is not backed up by a solid knowledge of the theories or broader industry knowledge. Many people don’t deepen their insight and further develop their skills. Some don’t like reading or are demotivated for one or another reason. Some persons are considered as expert already after a few years. Some experts may acquire 1 year of new experience and then repeat this experience for the next years without fundamentally learning something new. These tendencies are not favourable for the development of real expertise. The Analyst has to take such situations into account.

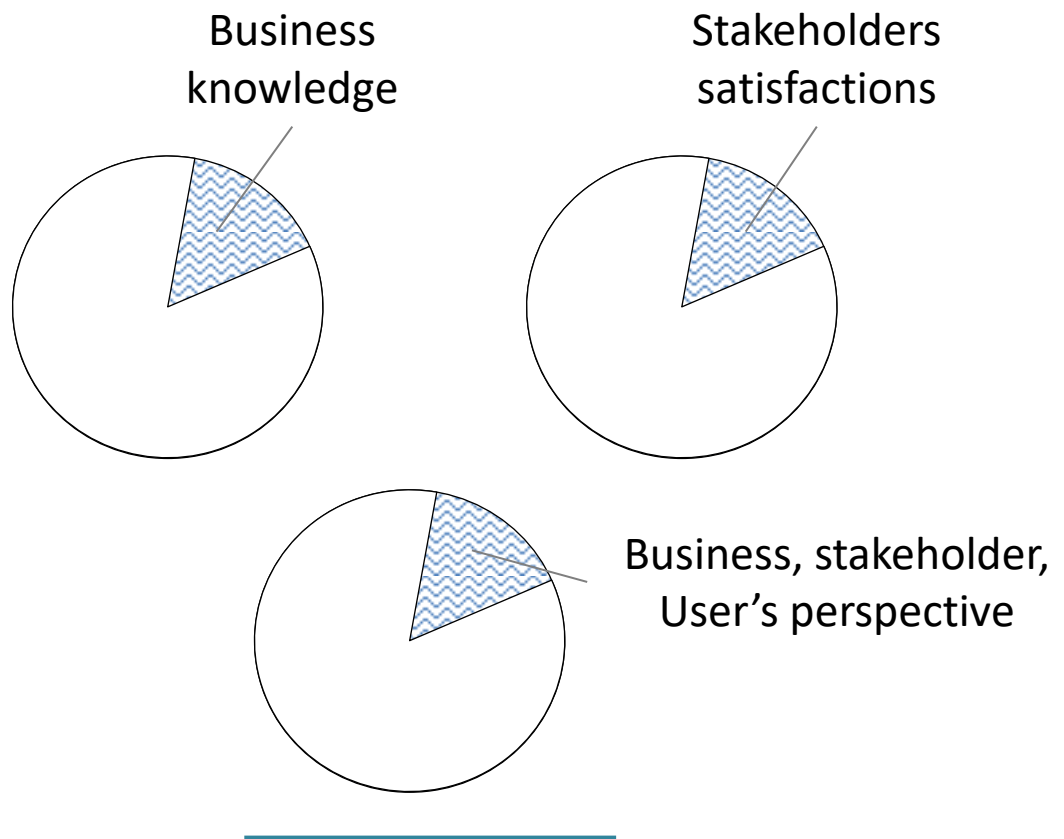


Business expertise is developed and learned to be understood and used by people. Its purpose is to allow people perform their job. People understand high level instructions. They can interpret them, evaluate and decide to act in the best way accordingly to the issue and to the circumstances. They can deal with uncertainty. They can adapt and improvise. This knowledge and expertise is meant for people. It has NOT been developed for software applications and computers.

The quality of information, particularly knowledge and instructions, meant for people is very different from the quality required as input for engineering information systems.

Notes:

The Analyst has to verify the information received from the business community and to deal with these types of difficulties. If the Analyst doesn’t, the system design may be conceived based on weak information.



Business knowledge is a narrow view of all the knowledge required to diagnose and to conceive suitable information solutions, let alone to innovate.

Users satisfaction is, as criteria, by far insufficient to evaluate a system properly. It is based on this same perspective.

The users/business perspective, although essential, is inappropriate to conceive and to evaluate a system.

Note:

Can a driver (user) decide whether a car (system) is safe and roadworthy?

Customer and Business perspectives and their knowledge are important. They are important to be taken into account in the design. But they are insufficient to diagnose, to design (conceive) and to fully evaluate information issues and information systems.

Is it possible that a solution satisfies the business stakeholders, yet hinders, limits or harms the company as a whole?

Engineering vs Usage

30/10/2018

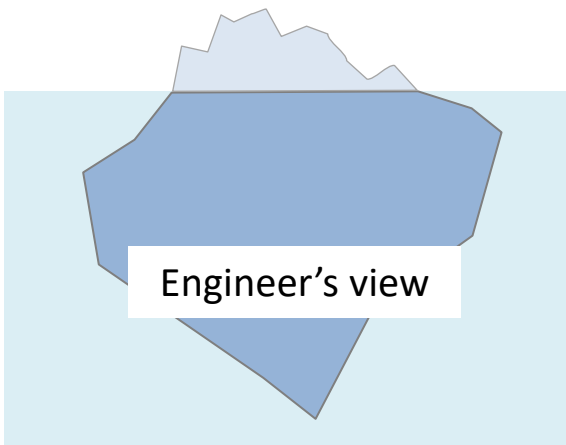
	Engineer	User
Focus	System oriented	Usage & Result oriented
Driver	How can I make the system better? How can I make better systems?	How can I use the system in a better way?
Thinking	Systemic Thinking, Engineers Thinking, Holistic, Methodical	Actionable Thinking

< CONFLICTING >

Iceberg

Ice above and below the waterline

User's view



In IT

The part above waterline is different from part of below waterline. GUI is by no means comparable with internals of systems.

Both parts are different

- in nature
- in possibilities
- in size
- in complexity
- in conceptual approach
- in required knowledge and competencies
-

< MISLEADING >

Reflection:

A User is NOT an Engineer !!

- Does a car driver understand the car engine?
- Can he/she engineer a car or provide car requirements?
- Can he/she lead a team of car engineers?
- Can he/she says when the car is roadworthy?

Car: The dashboard, steering wheel and pedals (users view) is very different from chassis and engine (engineers view). → Ability to drive a car ≠ Ability to engineer a car.

An engineer's brain is differently wired than a non-engineer's brain. They have different world view and have different thinking patterns.

A System is more than Features

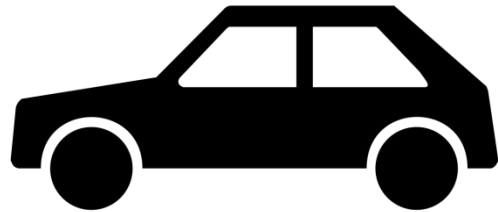
10/01/2019

What the users know of
software applications



“User Interface” and
“features”

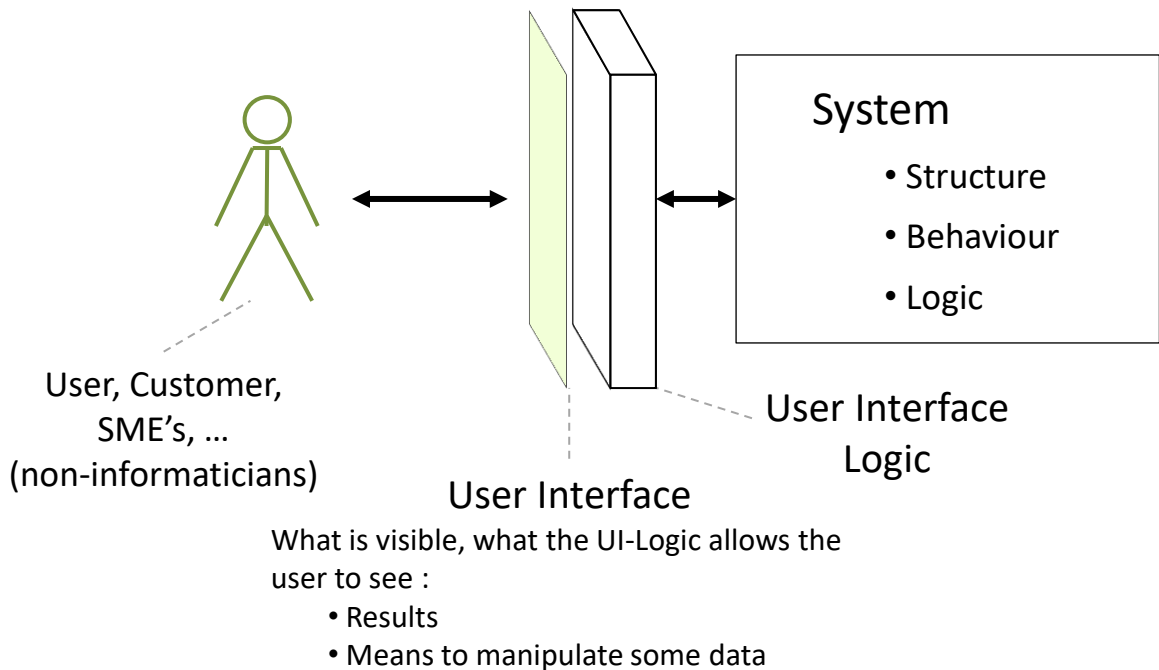
What a software
applications is



- Does a driver (user) understands the car (system)?
- Can a driver judge and value a whole system through its interface?
- Is the dashboard a relevant picture of the car's internals? Does it reliable reflects the engine? Or are both of different nature?
- Is it the role of the driver to engineer the car?
- Can a user engineer a car?
- How would the car look like if it was engineered by a user? Would it be roadworthy? Can a driver even evaluate whether his/her car is roadworthy?

User Interface is not the System

10/01/2020



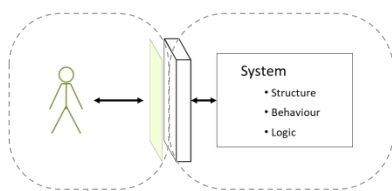
Major Assumption

The UI reflects the system sufficiently and reliably

(it's nature, its content, its functioning, its logic, ...)

“The User Interface is a Lie”

- protects the User from having to know the whole complexity of the system
- protects the User from having to know too much
- provides a distorted and simplified view on the system
- and hides the reality, the true system, the real complexity, ...



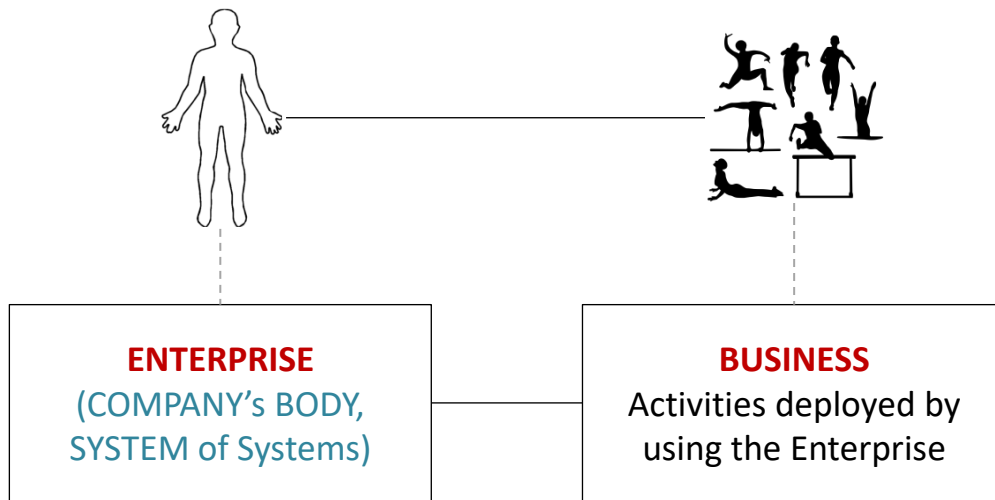
This knowledge ≠ this knowledge ☛☠

Because using a system requires lesser knowledge about the system than conceiving it. These are 2 different jobs.

Consequence:
“The customer doesn't know what he wants”

Business Activities vs Enterprise

30/10/2018



- Business defines the activities deployed by using the Enterprise.
- Business defines some(!) of the capabilities of the Enterprise

Business, as activity and as field of expertise, impacts the design of the enterprise as a system, but it does NOT define it.

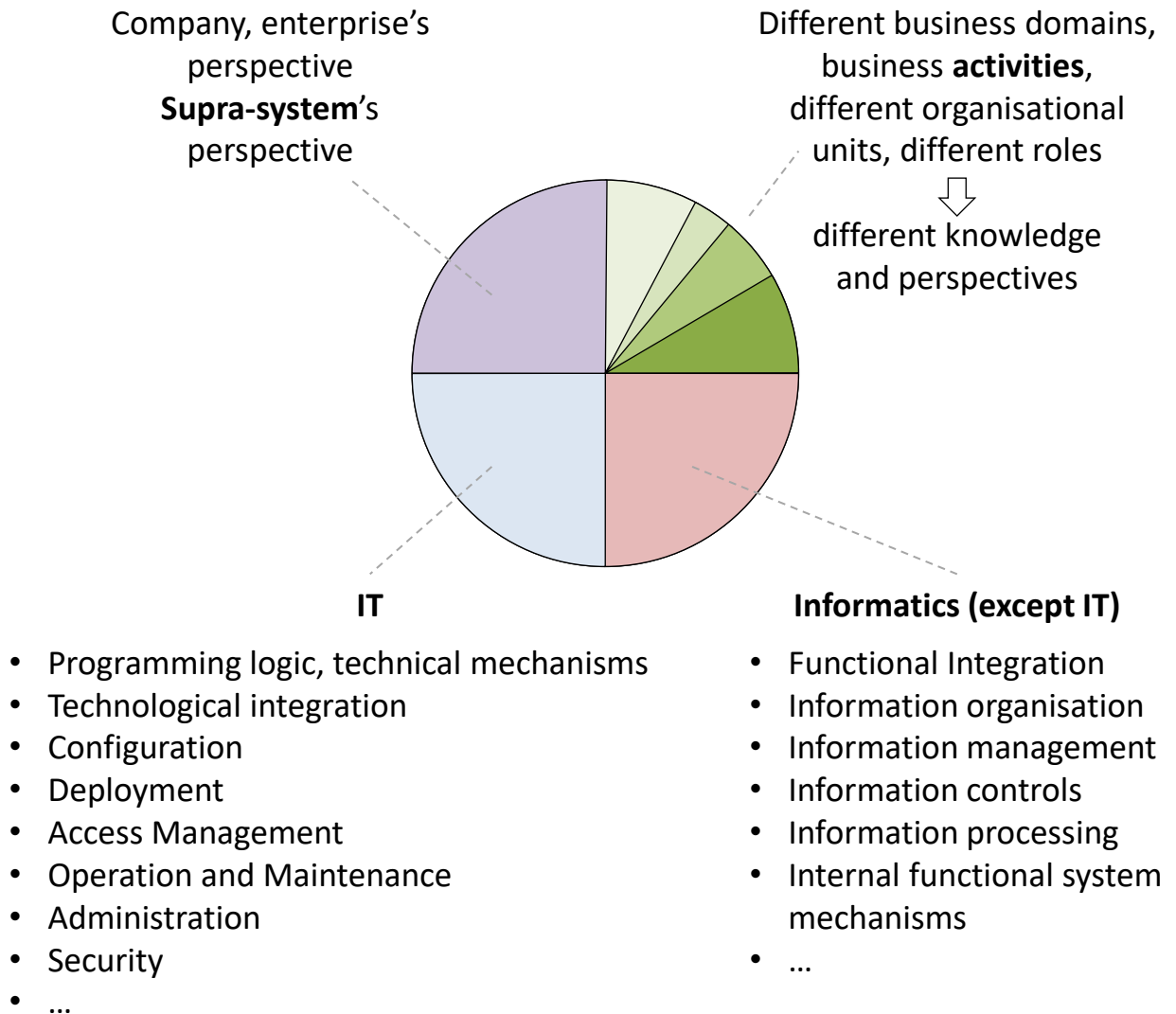
Why? Because the business perspective is only one perspective; a partial, on-sided and limited view. It does not provide a holistic view combining all the perspectives.

Notes:

The physical activities don't define our body. They define the capabilities required by the body. Training creates these capabilities. The need for survival impacts the design of the human body much more.

A same business (same business activities) can be implemented by **very** different systems of systems. These systems implement different concepts, different logics, different architectures, but they implement the same business logic. (like "one problem can be solved by different solutions")

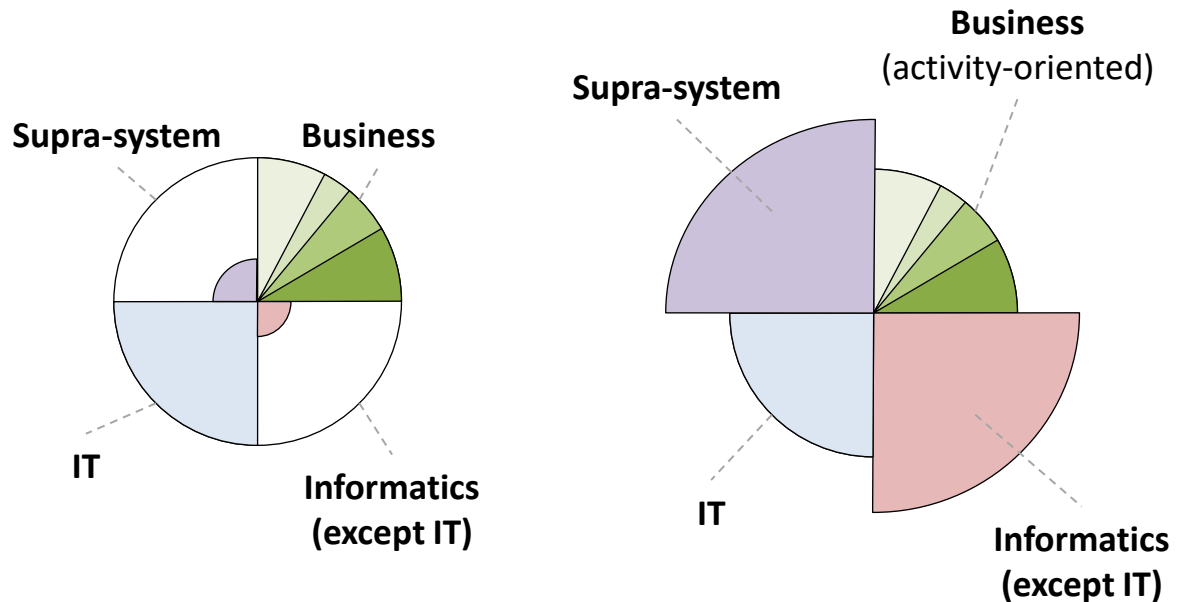
Business Knowledge and Expertise in Software Systems



Besides the business logic, software applications contain a lot of other logic that matter very much as well.

Implementing business logic with just enough of other logic (like screen and menu management and logic to store and retrieve in interactive software application) is unlikely to lead to robust, well-functioning and lasting systems.

Business Knowledge and Expertise in Software Systems



Can be true for simple problems, concrete and local logic, small(er) solutions

Larger, sharing, multi-business-domain, integrated more complex software systems require more of THIS model

CAUSE OF FAILURES OF LARGER PROJECTS

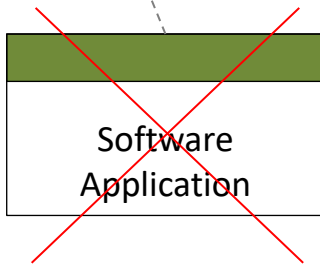
is the underestimation of the required degree of Project Management expertise, Systemology and Informatics (non-IT part) expertise required by the project.

This leads to

- ineffective project organisation and forms of collaborations
- and to the non-respect of principles in the above-mentioned disciplines.

Business Knowledge and Expertise in Software Systems

*Automated Business Layer
(pure business knowledge)*



The implemented (pure) business knowledge does rarely/not necessarily form a uniform upper layer in software.

Implemented business logic appears more as tests, computations, connections, calls, and so on. It is spread throughout the software application's logic and databases. It is intermingled with other types of logic.

Not all logic used to guide activities executed by business people is business logic.



If business logic would fundamentally define the solution then once the business logic is settled, only one possible solution can be created. This isn't the case. Business knowledge and expertise influence the solution, but doesn't define it. It may determine parts of it.

Non-IT people are not used to deal with such abstract, detailed and formalised logic. All the program logic behind the user interface is completely different from the logic end-users are accustomed to deal with. It is the formalisation of a very detailed mathematical precise formalised fixed(*) logic. (*:people can interpret instructions, computers execute the code like it is written). End-users also know pieces of logic implemented in a software application and the result of some functions.

We know (common complain):

“The Business doesn’t know what they want”

- Given the nature, the quality, the fragmentation, the spread of business knowledge,
- Given that the role of the business community is to make the business running
- Given that most people don’t know what computers can do or can’t do
- Given that most people have no clue about IS engineering, Systems Science, ...
- Given that most people’s brain is not wired like that of an engineer

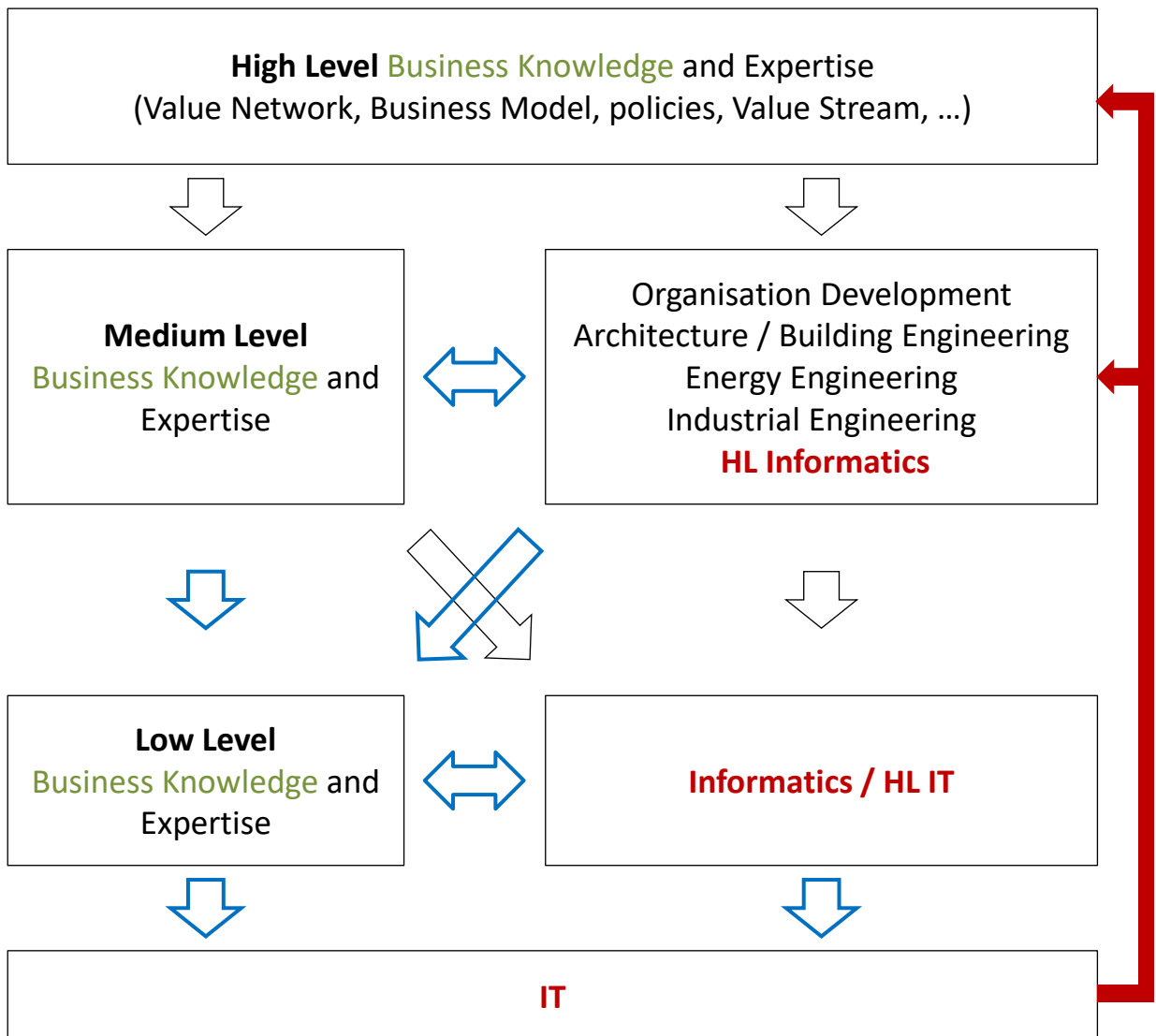
and

- Given the business community is not qualified to engineer information systems

any approach expecting the business community to define its information needs, to specify what information solution and its outline they need, taking informatics decisions and even “guiding” IS projects is a way of working to avoid.

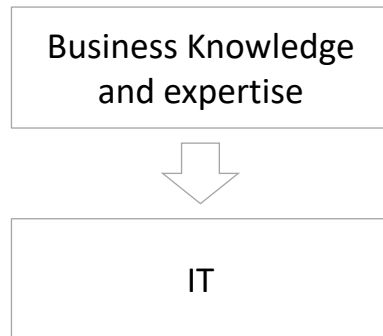
Notes

Business community should take business decisions. Informaticians should take decisions related to information, information organisation, information processes, information systems, ...



- Information products & Information services change business models
- Web, RFID, Smart devices, BI, Big Data, AI, ... change the organisational structures, the collaborations, how business is conducted, and even business models. (←)
- The power of Informatics is at the level of information concepts, information organisation and information processing (example: AI). Although this is “abstract”, this shouldn’t be underestimated. Therefore, one need to know what software and computers can do and can’t do.

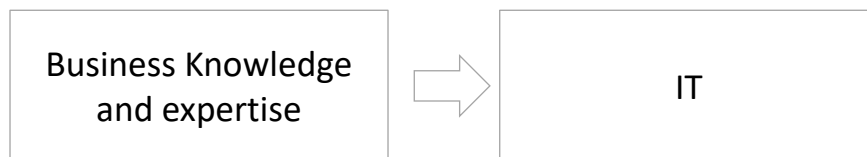
OLD Paradigm



Business knowledge governs the IT expertise.
“IT serves the business.”

Consequently:

Habitual Project Methodology or Approach



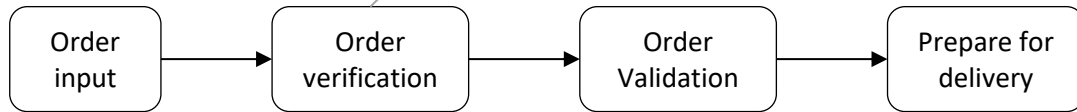
- 1. Business stakeholders guides the IT projects**
(determines, defines, controls, manages, ...)
- 2. “Throw-it-over-the-wall” - approach**

First, determine the Business knowledge and expertise to be implemented.
Pass it on to IT. IT implements it and also adds its own logic.

This belief does not match the reality, yet it is commonly implemented in approaches and applied. It is a source of problems and serious limitations (among others, inhibiting innovation).

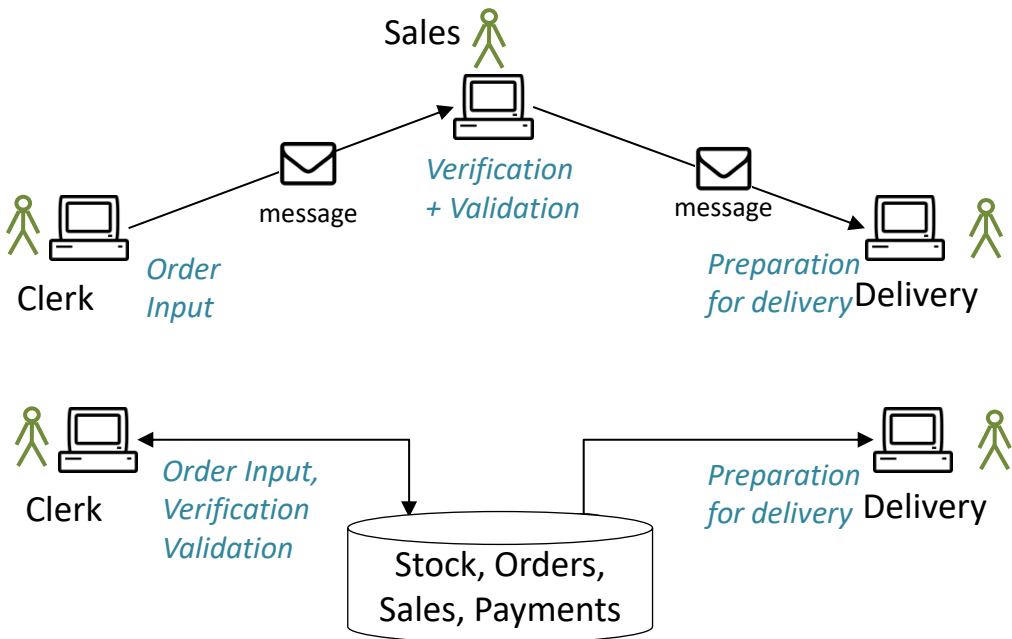
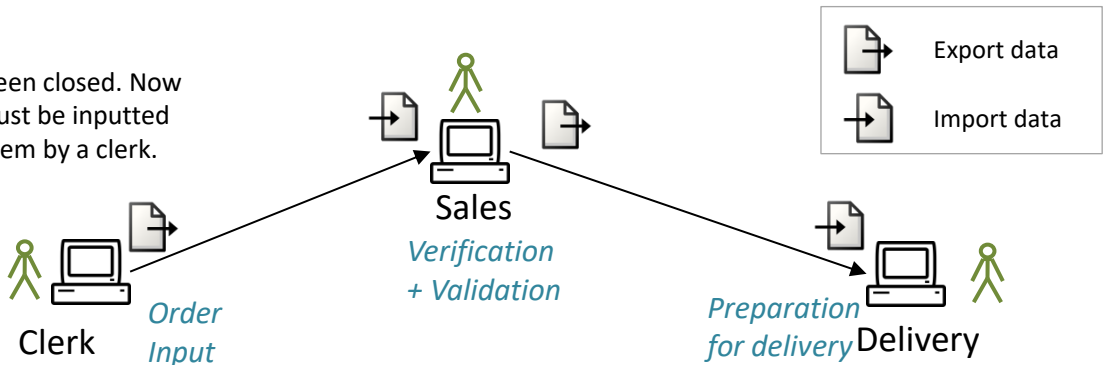
Pure business logic

- Check client history (unpaid orders)
- Check order with available stock
- Optionally provide discount



IT

A sale has been closed. Now the order must be inputted into the system by a clerk.



Verifications and validations are done by software. Sales person doesn't need to intervene anymore between order input and order execution (except in case of negative validation).

Notes:

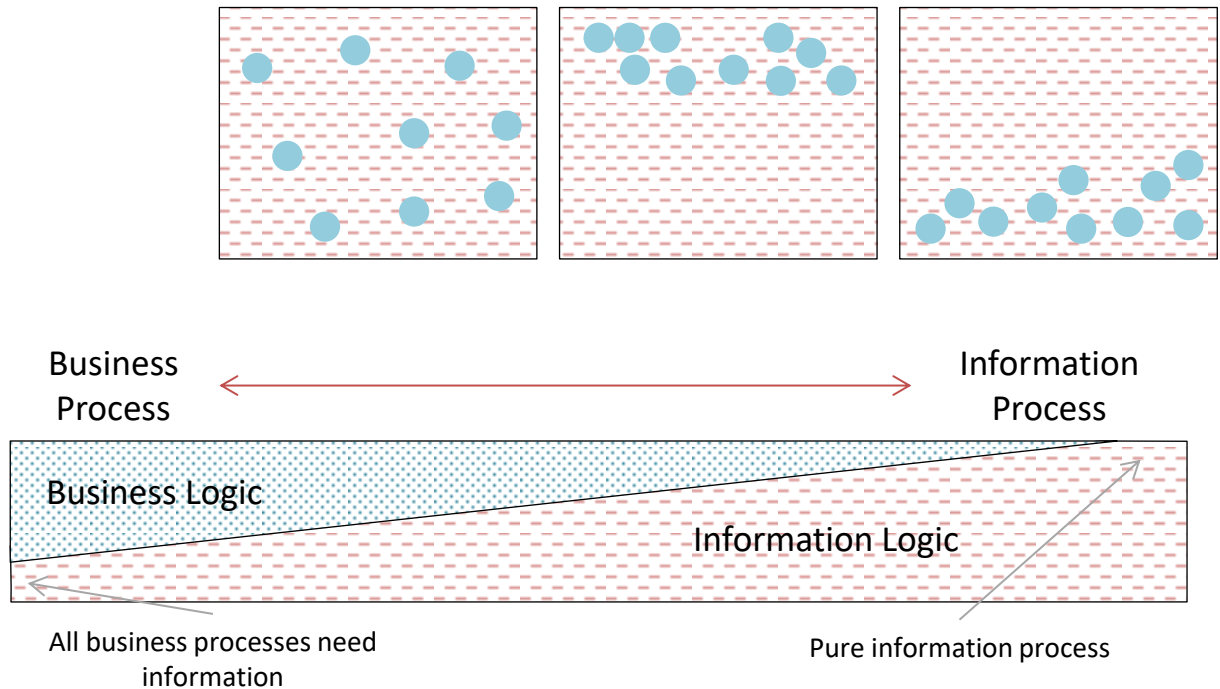
The business logic remains the same in the 3 implementations. But the actions executed by the business community are changed.

Business & Information Logic

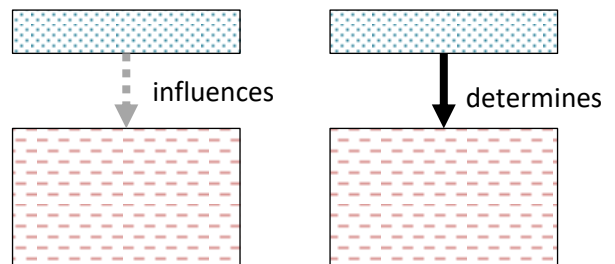
30/10/2018

Business Logic may be dispersed in an information process and can be more or less concentrated in one or another area.

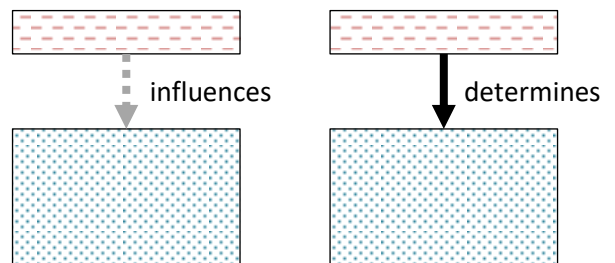
(example: business rules)



Business logic may influence or determine the structure of an information process



Information logic may influence or determine the structure of a business process



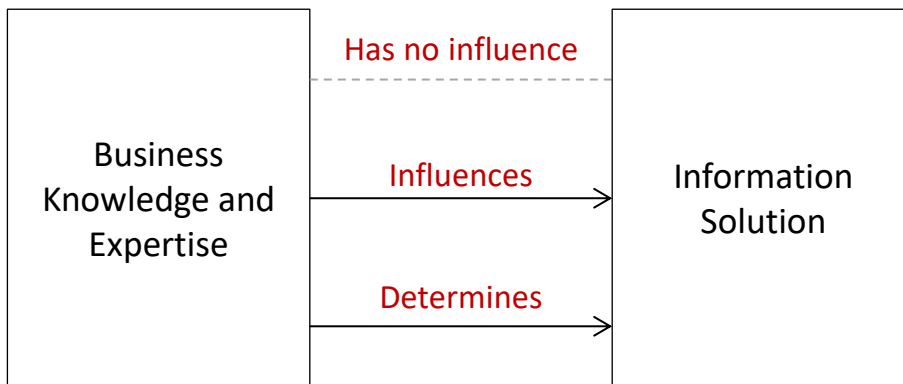
Notes:

This impacts the design approach, the priorities, the focus, the leading competencies, ...

2nd drawing: shows only the amount of both types of logic. It doesn't show 'layers' of knowledge.

Business Knowledge ↔ Informatics

30/10/2018

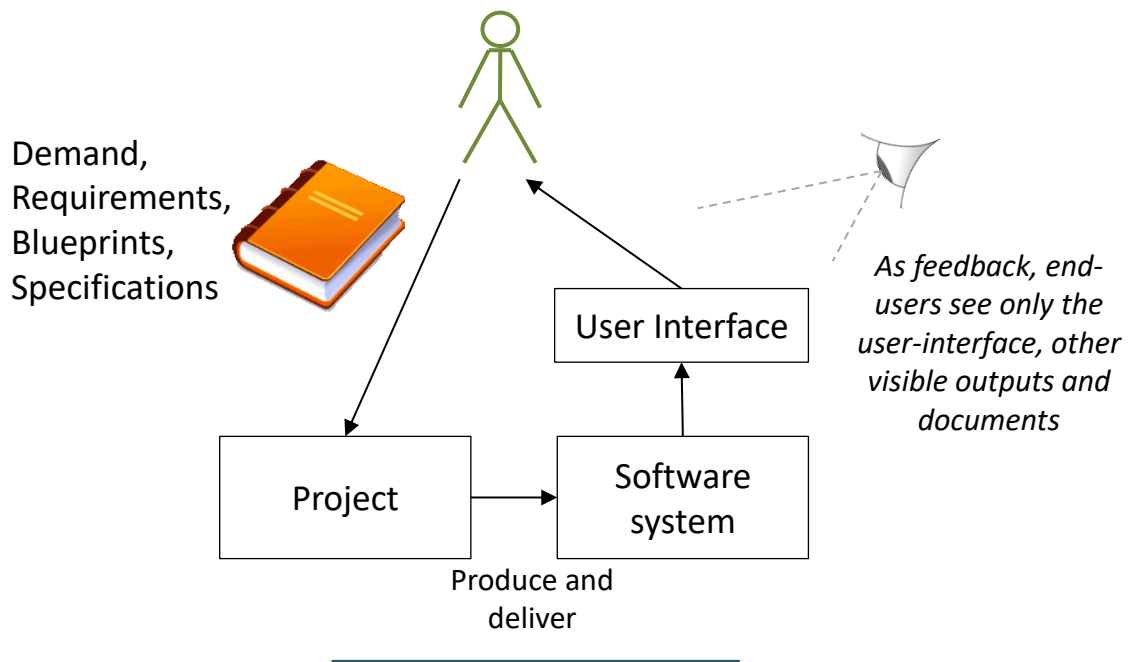


Notes:

- Some business knowledge has no influence on the concerned information solution.
- Or, its influence is local and limited to the execution of it (like a specific formula).
- Some business knowledge influence the information solution. For example, some parts of the business may be very strict and formalised and use procedures, while other area's are more free and function based on *cases*.
- Some business knowledge determines how information is processed (like the computation of price). However, not every information processing is determined by business knowledge.
 - Example:
 - Selection algorithms
 - Sorting algorithms
 - Matching algorithms
 - Identification of data elements
 - Data quality verification
 - Prioritisation algorithms
 - ...

Software Applications Under Control of Non-IT People

10/10/2018



1. The user may assume the demand, requirements, blueprints and specifications are implemented in the software application.
2. The user may deduce the internal logic and behaviour of the software application through its user interface and other observable outputs.

AND

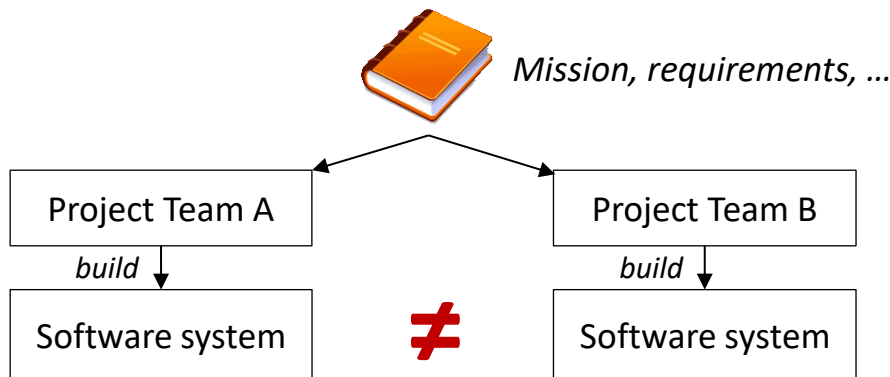
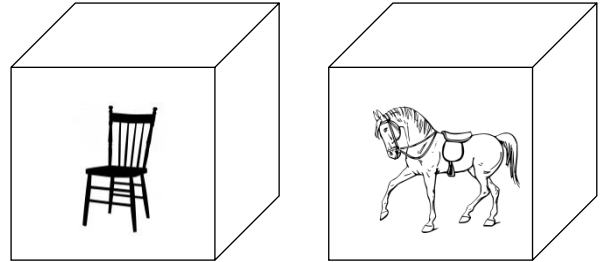
3. The user may imagine how the logic expressed in the transmitted documents have been implemented.

THE CONTROL IS VERY WEAK,
NOT TO SAY ILLUSIONARY

Software Applications Under Control of Non-IT People

What is in the black box?

“It must have 4 legs and it must be possible to sit on it.”



Same requirements for same problem, same environment, same ... going to 2 different team lead to 2 different solutions implementing the same business logic.

Why?

- A same business logic can be executed by different mechanisms.
- A software application is also guided by non-business principles and contains a lot of non-business logic and even complete non-business mechanisms.
- Business logic often influence, but does not necessarily determine the architecture.

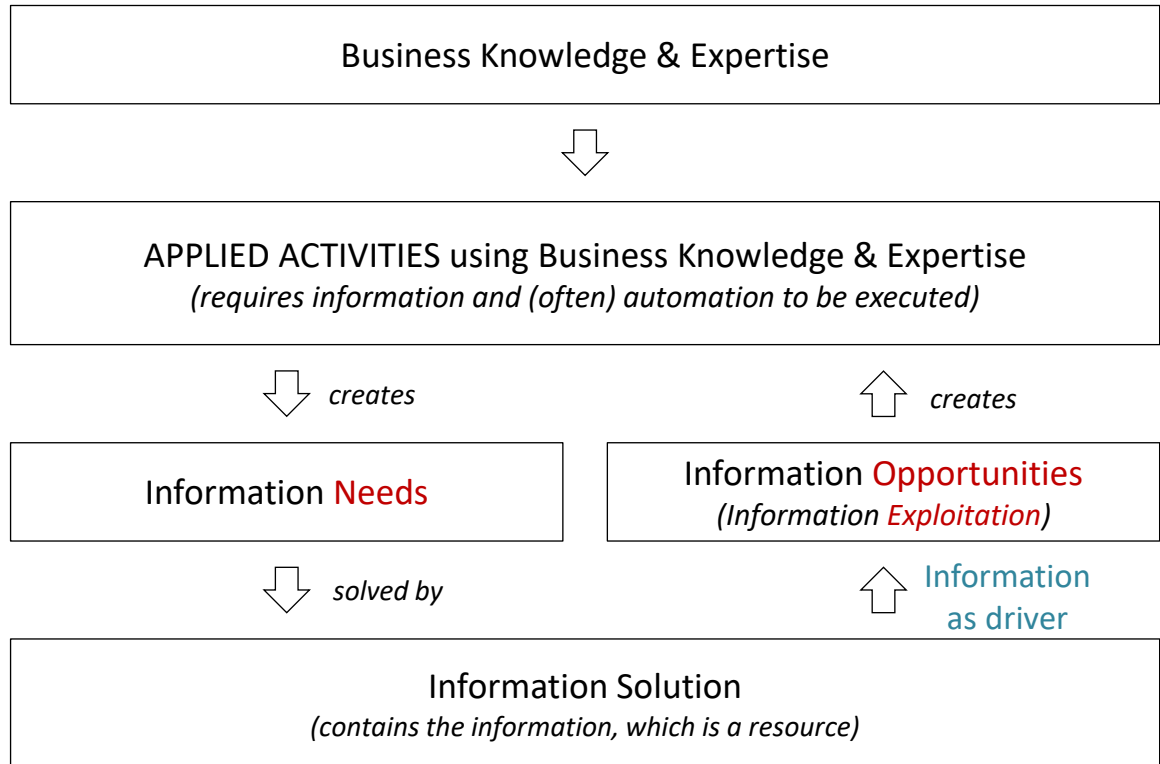
Although useful, requirements are insufficient to conceive and to build the right solution. And it is still a control, but it is a control which is still too weak.

Notes

If requirements are given to the teams, the differences may be important. On the other hand, if specifications are provided to the teams, the differences will be smaller.

Business Knowledge ↔ Informatics

30/10/2018

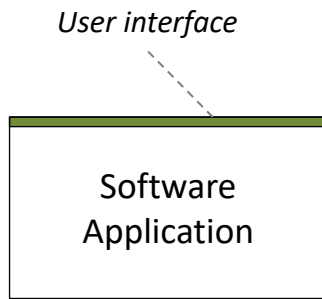


Notes:

- Without information, business activities can not be applied. The application of business knowledge and expertise creates information needs. Solving these information needs can only happen with information capture, processing, transport and storage and retrieval.
- While the information aspect has to respect the business logic, one should not confuse both. They are NOT the same.
- But information, through its intrinsic value, provide also opportunity for business activities. Available information and potential of informatics creates opportunities to improve business activities.

Knowing Software Applications ?

30/10/2018



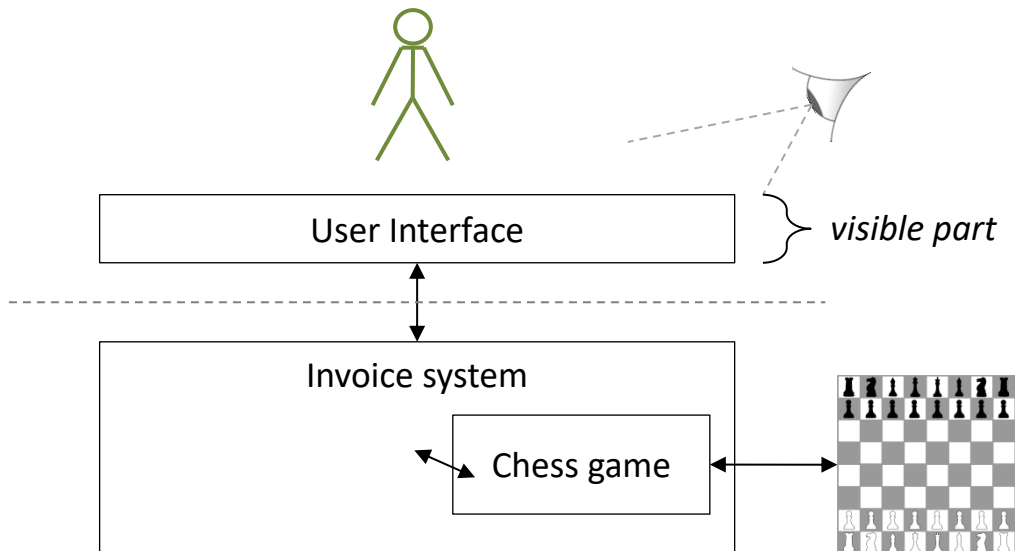
The user interface (together with some concrete outputs) is the only part of the software application visible to end-users. The end-users know and can use the user interface.

The user-interface forms a barrier between the end-user and the rest of the software system.

It prevents the end-user to see the internals of the software application. It is like a concrete layer on top of the software application. They have no access to anything beneath the user-interface.

The user-interface is like the dashboard of a car, a drink distributor or a movie. A dashboard is very different from the car engine. It tells very little about it. Knowing how a drinks distributor looks like and knowing how to use it, does not imply knowledge of the internals of a system. Or a movie tells little about the movie set.

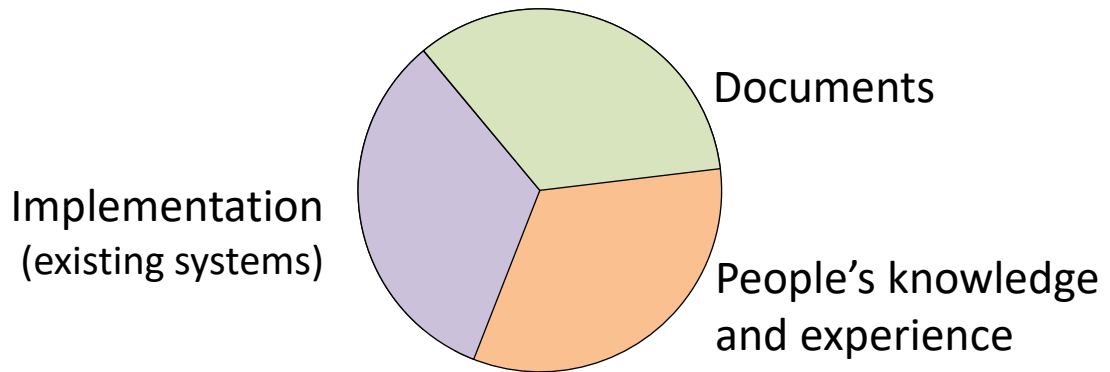
Knowing and being able to use software through the user-interface is very different from knowing the internals of the software application.



What prevents developers to convert invoice data into chess moves and to send them to a chess game? The end-user wouldn't notice anything.

Sources of Business Domain Knowledge

Three Types of Sources



Notes:

Documents

- Policies, documentation, procedures, ...(present); Mission, vision, strategies, tactics, objectives, plans, novelties, opportunities, ...(future)
- Formalised, but can be outdated

Implemented and Execution

- Existing systems, existing procedures, communications, actions, work, organisational structure, decision making processes, structures, concepts, principles, ...
- It is the present, the “as-is” ... and it is an important source for diagnosis

People's Knowledge and Experience

- Knowledge, understanding, insight, experience
- Thinking patterns, beliefs, norms, values, priorities, expectations, fears, tendencies, habits, decision forming, unspoken objectives, personal drivers, cultural aspects, limiting beliefs, ...
- Through communication
- Subjective, interpretative
- Relies on memory → variable, different perspectives, different understandings
- Different persons may have contradictory ‘knowledge’

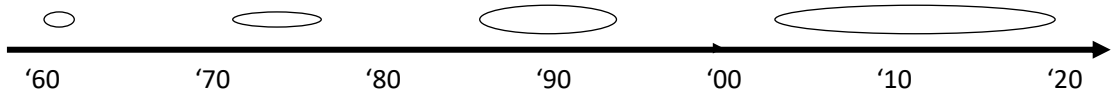
We may find information in one source that we don't find in another. It is up to the Analyst to put it all to sort it out (and to get confirmation and consensus).

1. Business knowledge is
 - intended to people and human activities, not to software and computers.
 - high level, partial, ethereal, uncertain, fuzzy, variable, not always complete and coherent, ambiguous, interpretable, ...
 - hidden in many forms, places and of different natures
 - not readily available for usage
 - constitutes only a part of all the required knowledge
2. Information solutions don't solve business problems (unless the business is about information). They solve information problems experienced by the business community, allow information needs to be solved and it is a way to seize opportunities offered through information (improving the exploitation of information). This is why business knowledge is only a part of the knowledge required to diagnose issues, understand environments and design solutions.
3. The required competencies to diagnose, fully understand and conceive information solutions and software systems are: information, processes, Systemology, engineering, Systemic Thinking, Information Science, understanding of possibilities and limits of software and computers, ...
4. Software applications are, basically, black boxes to the business community. The mastery and understanding of software application is based on assumptions and confused with the usage. The control of their internals, which is what matters, is rather an illusion. IT people are also users of software applications, but they know the software applications they developed themselves.

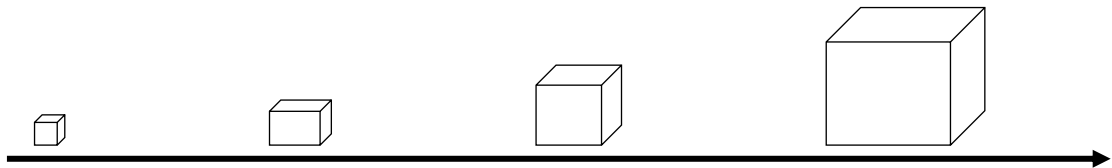
There is a huge work for the Systems Analyst (or BA) to search, to gather, to study, to consolidate, to stabilise and to verify and to select the necessary business knowledge, as one of the inputs of the required knowledge.

HISTORY

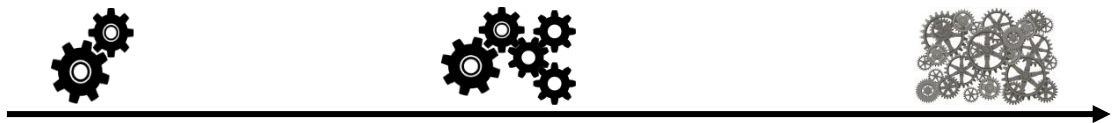
Problem Scope & Functional Breadth



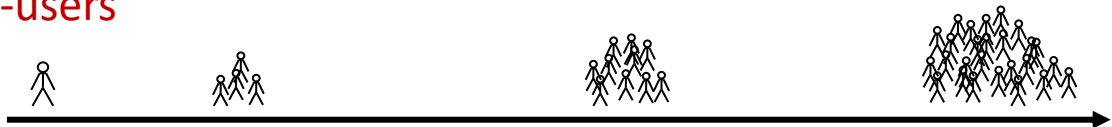
Size



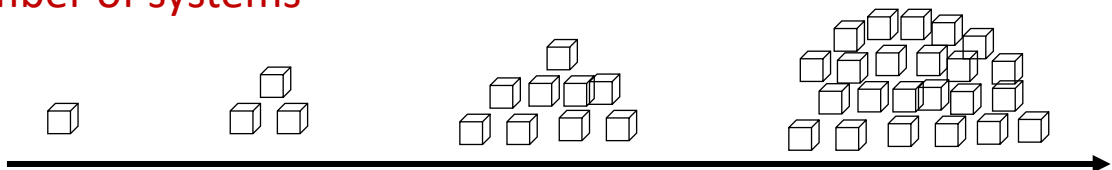
Complexity



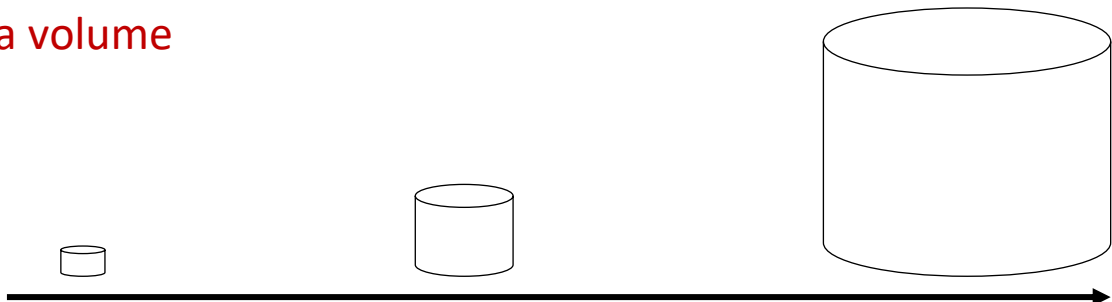
End-users



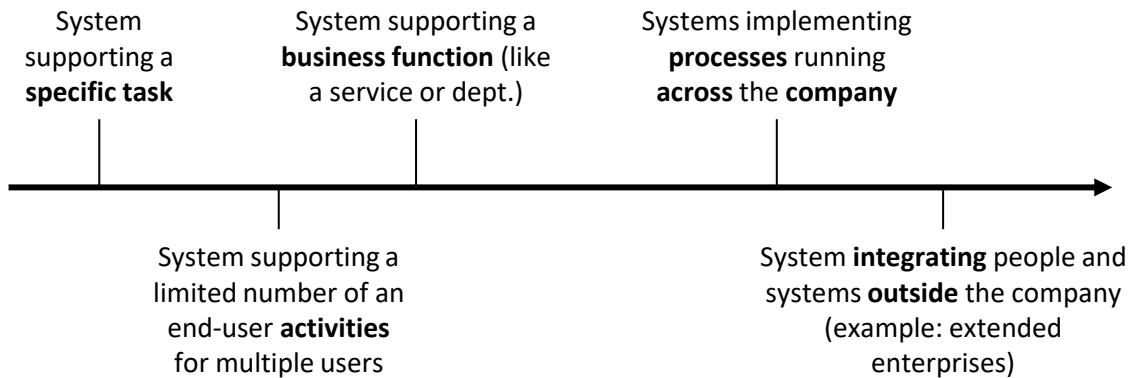
Number of systems



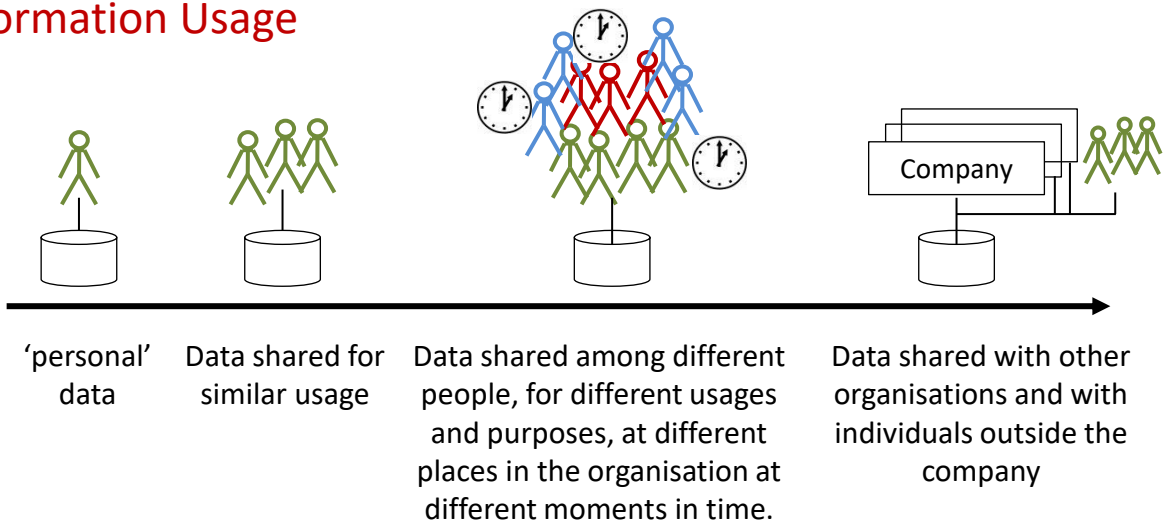
Data volume



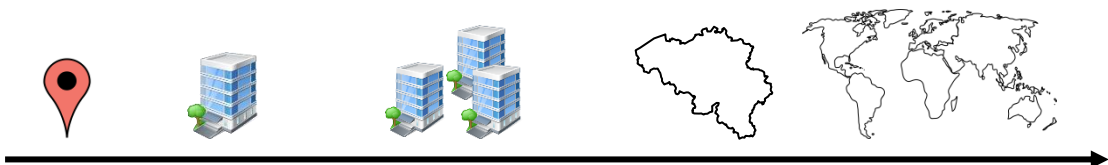
Relation between automated work and organisation



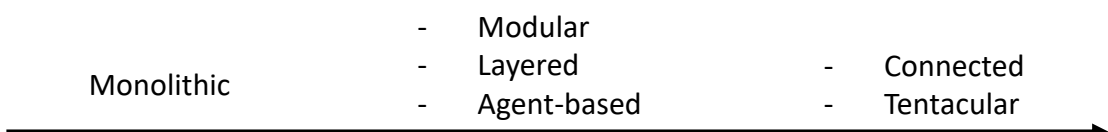
Information Usage



Geographical spread



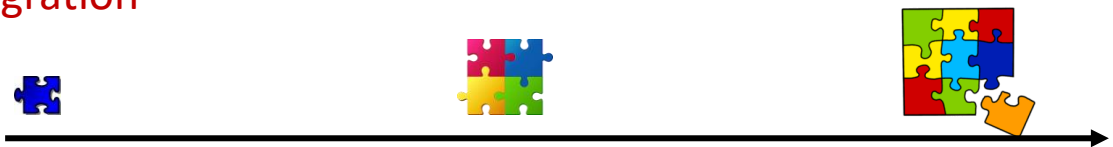
Architecture



Inter-systems Interactions



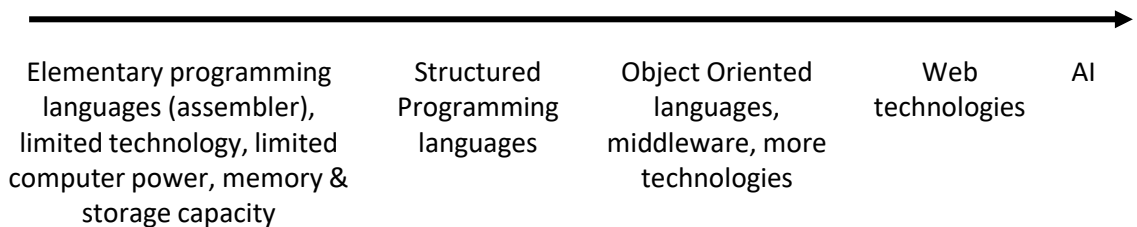
Integration



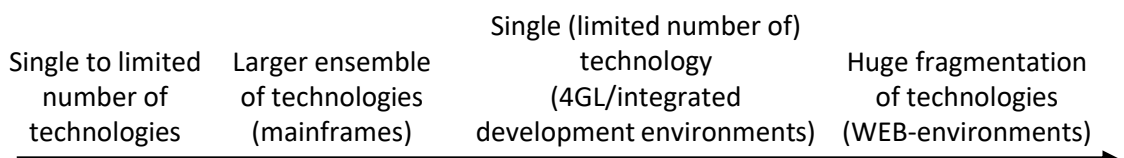
Focus of Engineering



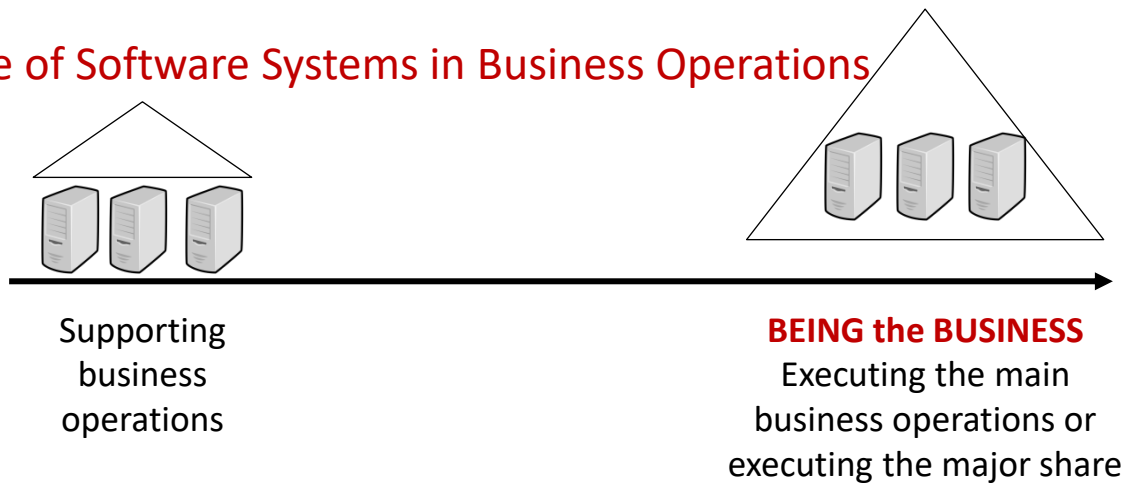
Development Languages and Techniques



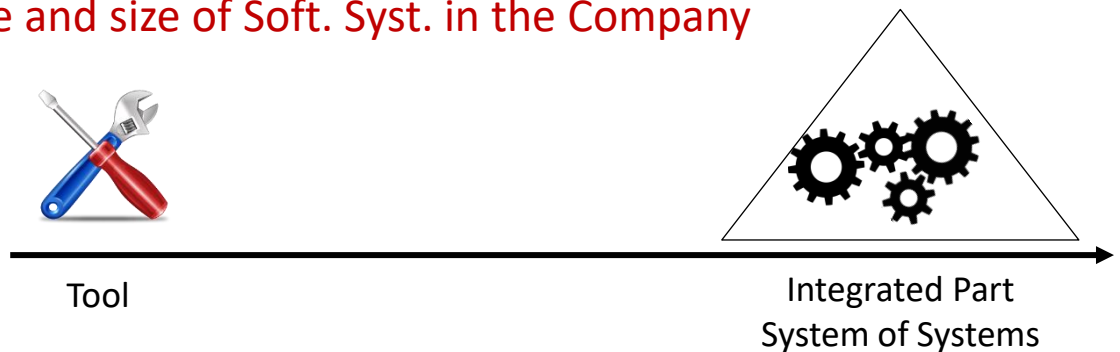
Technological Fragmentation



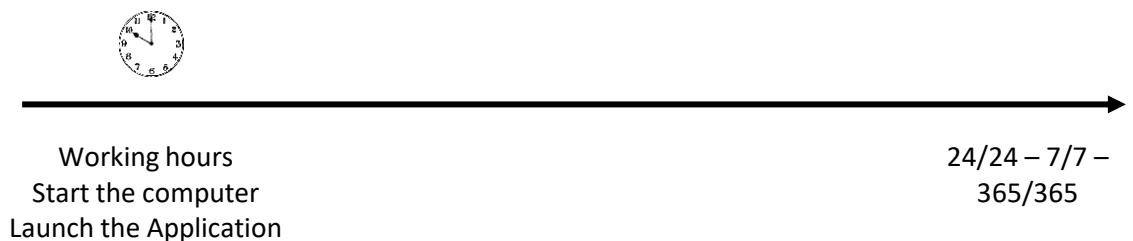
Role of Software Systems in Business Operations



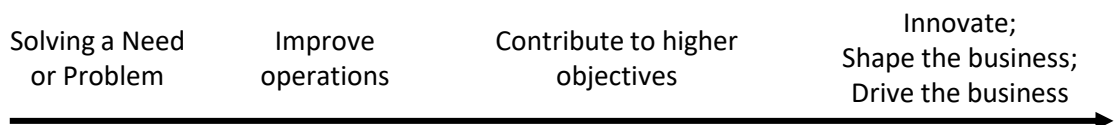
Role and size of Soft. Syst. in the Company



Timing

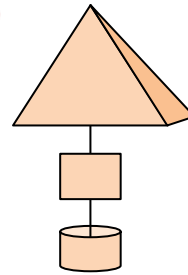


Contribution to the Business

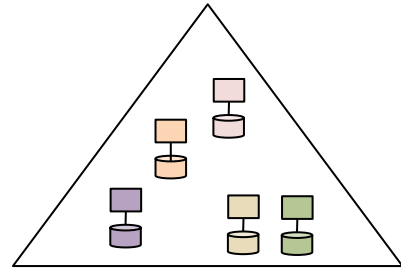


Evolution of IT Systems in the Company (1)

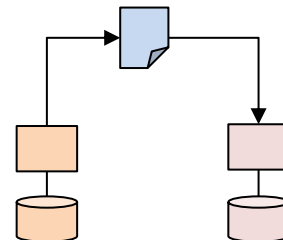
1) A new system supports administrative tasks or executes some administrative tasks of an organisational unit.



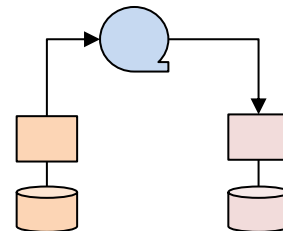
2) More systems are built independently from each other. Issues: different technologies; duplicated data; no consideration of reuse, enterprise-wide processes, sharing data, ...



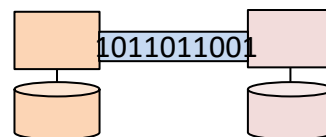
3) Communication between the systems happens by printing data on paper and re-encode the data in the other system.



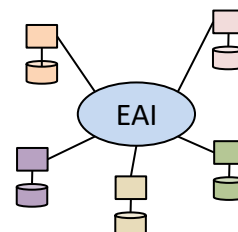
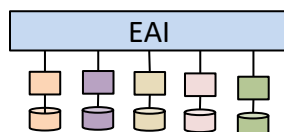
4) Communication between the systems is automated: file transfer (E.g. FTP)



5) Middleware



6) Enterprise Application Integration

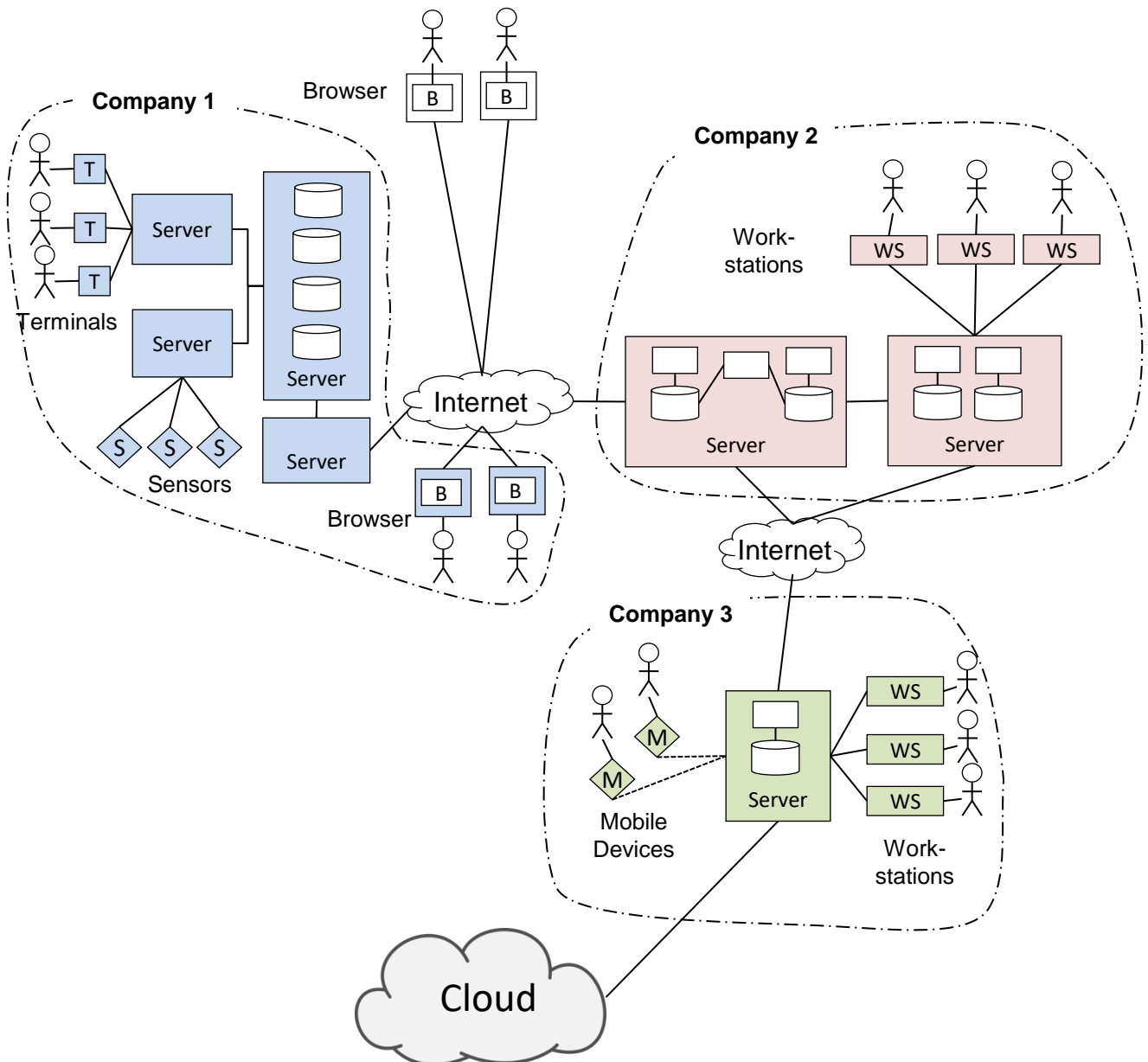


7) Enterprise Systems Bus

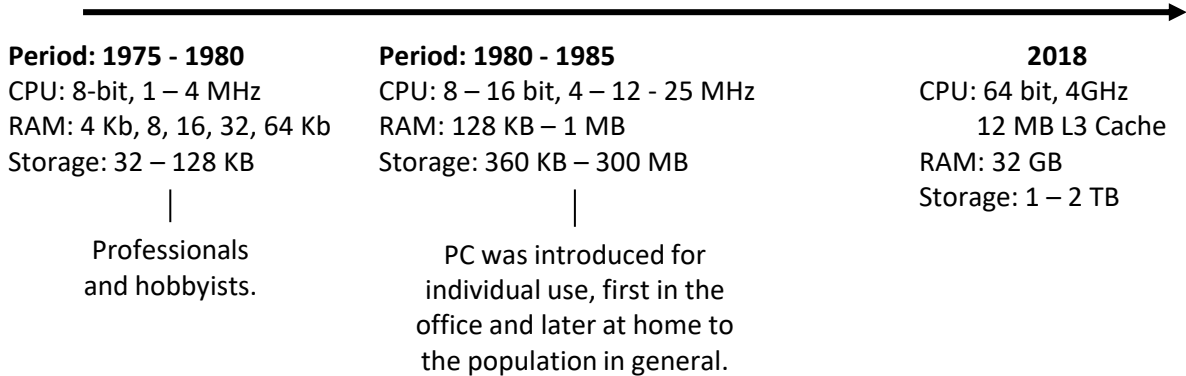
Evolution of IT Systems in the Company (2)

7) The **Extended Enterprise**:

systems goes beyond the company's borders, connecting systems of different organisations with each other.



End-User Computing Power



The early software systems consisted of a software application running locally on one computer using data stored locally and was operated by a single end-user. Likely, this software has been written in Assembler. It is very demanding to develop software in Assembler. The computer hardware had very limited computing power. The first software applications were rather small.

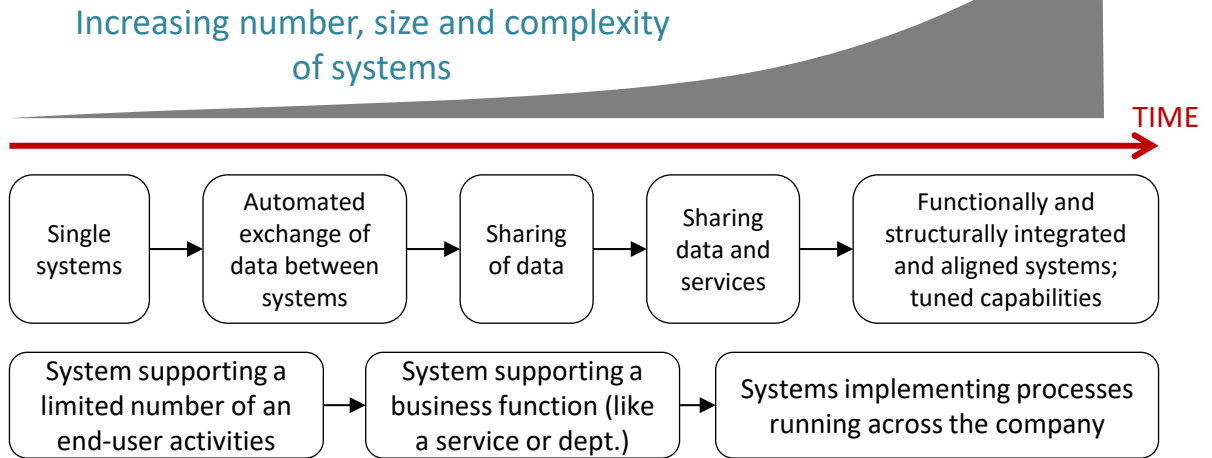
Better programming languages appeared. Among them the so-called structured languages and later the 4GL languages. They facilitated the development of larger software systems. At the same time, hardware became more powerful. The number of users for a single software application increased. First, multiple users using each an occurrence of the same software application and later the multi-user systems appeared.

The Analysts and software engineers faced challenges that couldn't be solved with the regular approaches and techniques used by then to conceive software applications. New analysis methods were invented to surmount those challenges. Now, even larger software systems could be developed. The object-oriented programming languages and middleware appeared. New architectures could be developed. Again, the habitual methods were not adapted to the OO-philosophy. New analysis and design methods were needed. Case-tools followed the development of new analysis and design methods.

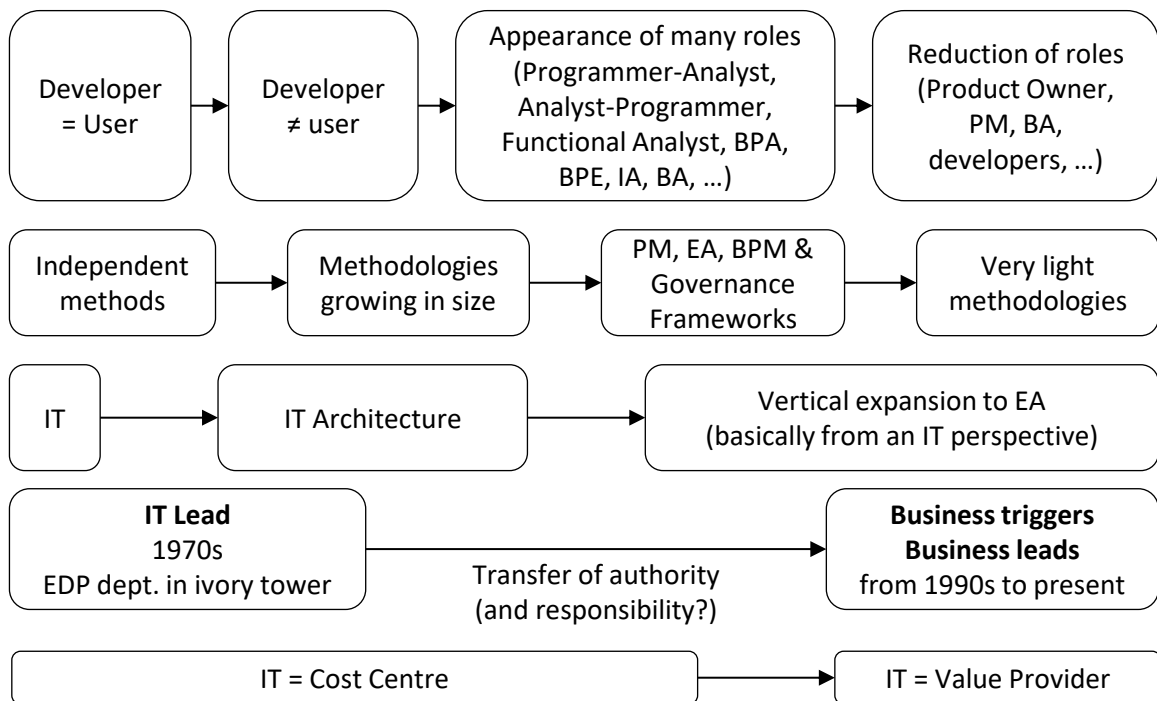
The size of developed software system continued to increase. Project management and methodologies facilitated the execution of such larger projects. Their importance increased. They became a necessity to respond to the size and complexity of the systems.

Needs drove innovation. These inventions increased the scale and complexity of the systems. The inventions and the larger systems brought new issues in matters of engineering approach and methods to the engineers. New methodological solutions were found to cope with them.

EVOLUTION OF SOFTWARE SYSTEMS

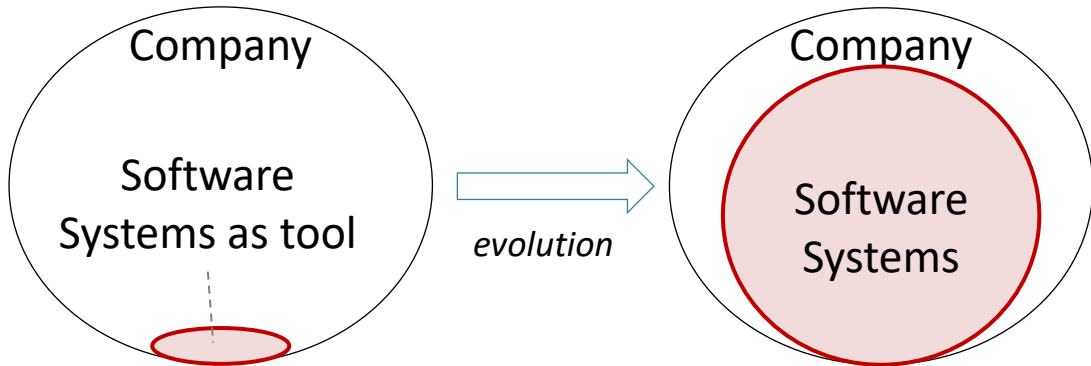


EVOLUTION OF SOFTWARE DEVELOPMENT



The evolution doesn't stop here ↗

- Evolution of competencies? Transfer of competencies?
- Building a small piece of software application is relatively easy. Larger software systems are a challenge. They have their own specific issues and needs.
- What triggered these evolutions? Is this evolution logical? (we need to think way further than the conventional answers)
- Does the evolution of systems and the challenge of developing them match the evolution of software development?



- Small
- Lesser critical
- Limited complexity
- Specific role/function (limited local scope)
- Supporting end-user's activities
- Serving 1 user, later a few users
- Without inter-system information exchange

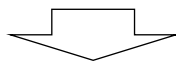
- Large
- Critical
- Complex
- Broader goals
- Performing the core activities
- Performing a huge share of the activities performed by the company
- Interconnected
- Crossing boundaries
- Inter-organisational information exchange

From building set of tools
to

Conceiving the 'core engine'
of the company

+

Explosion of information, communication, systems and connections



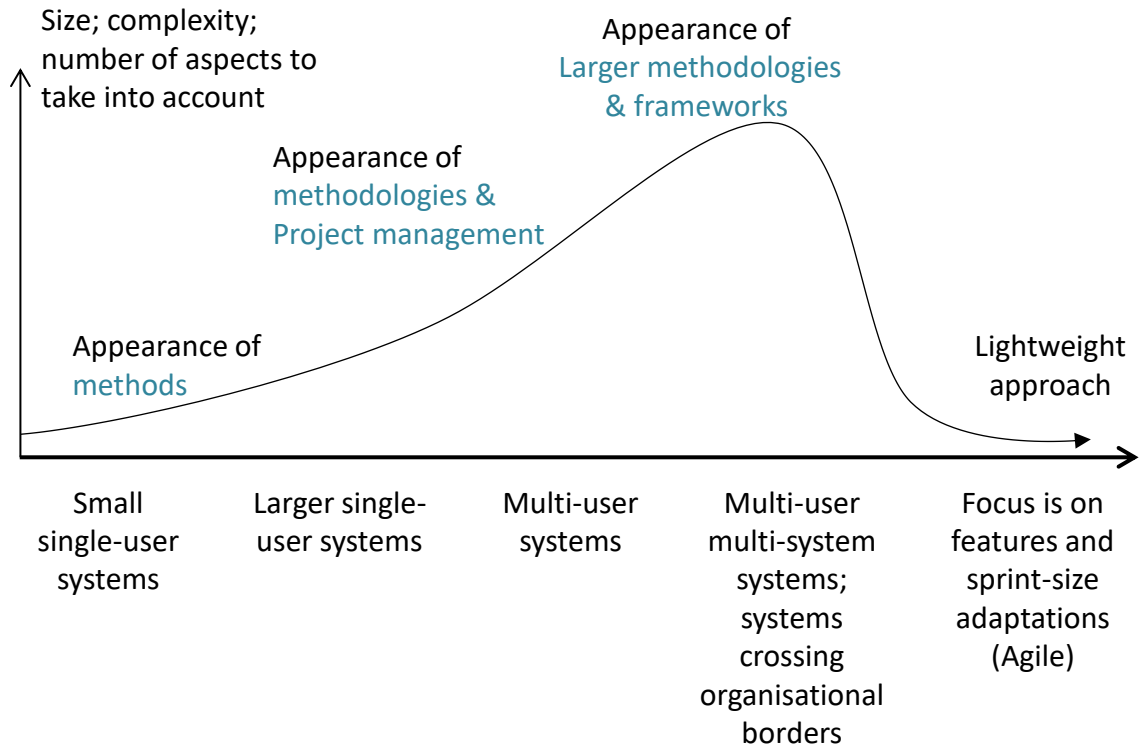
Fundamental Change

The Importance of the Role of Analyst INCREASED
and became CRITICAL


This is absolutely not about more of the same or the same at larger scale. It is a much deeper change requiring a different way of thinking, different competencies, ...

Evolution of Methodologies

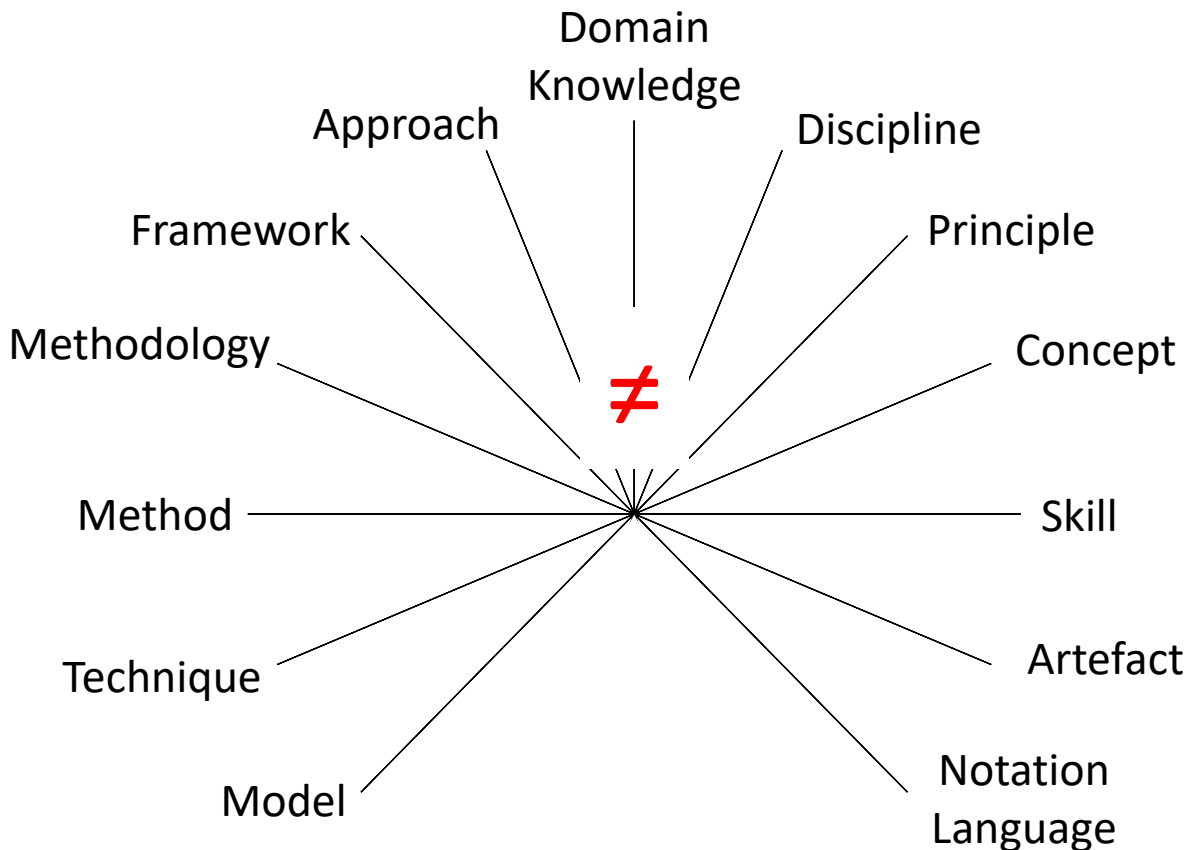
30/10/2018



Methods, methodologies, frameworks, project management developed in response to issues and needs created by the increasing size, complexity, risks and cost of software development projects and to increase the efficiency. Each step in the evolution was a learned lesson and it enabled them to cope with the encountered issues.



GENERAL SUBJECTS AND VIEWS



It is important to use these terms correctly and to classify elements correctly (to avoid confusion).

Examples:

KPI is a concept, tool, metrics; not a method.

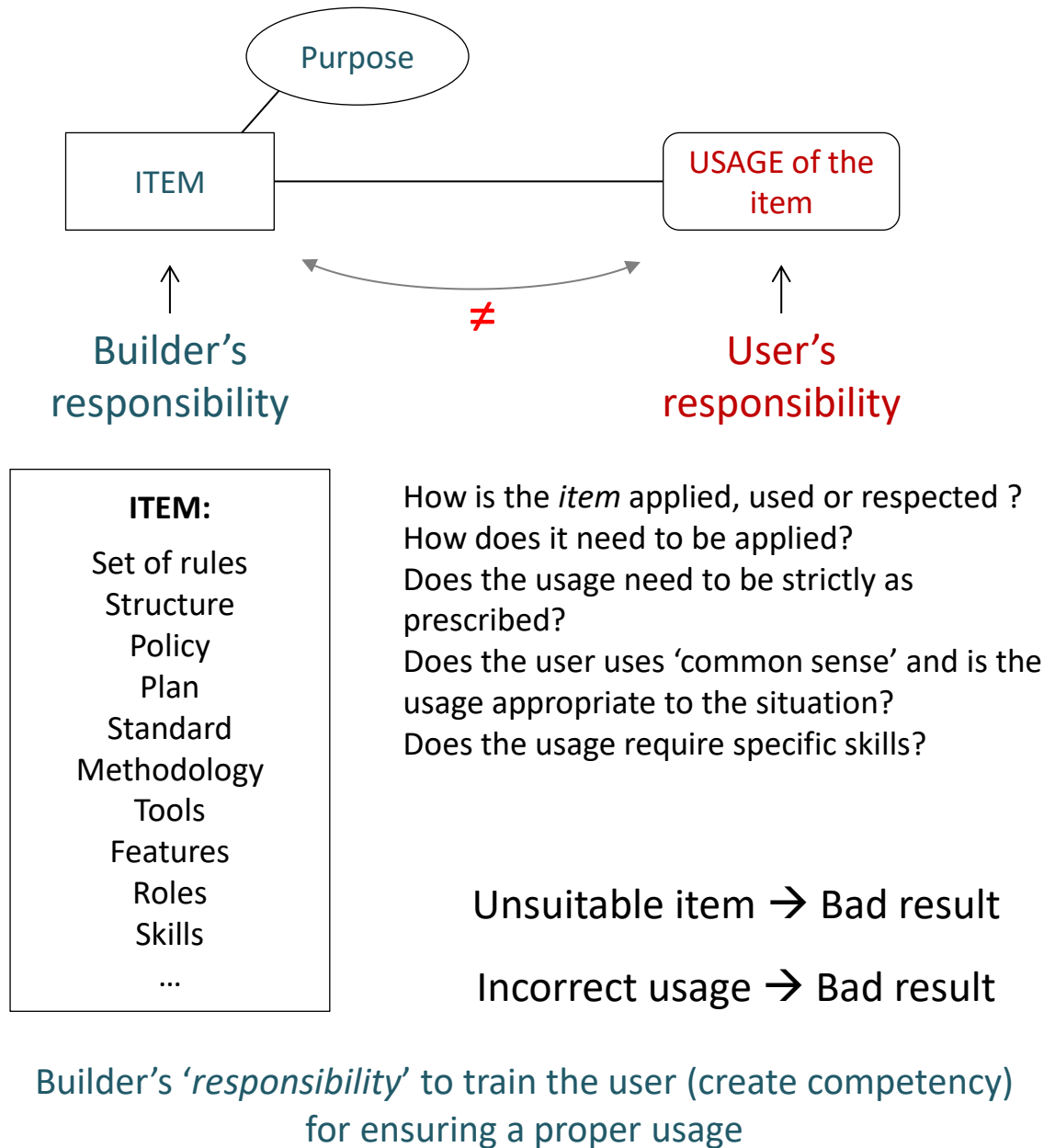
UML and BPMN are languages, not SE methods or methodologies. It can be a method of representing something.

Agile is a philosophy, not an SDLC or a methodology.

SDLC is not a project management methodology.

Waterfall is an SDLC, and as SDLC it is a methodological concept. But it is not a methodology. Some methodologies are based on the Waterfall SDLC.

Process Analysis is a discipline, not a method.



The item or its usage can be adapted to suit a specific situation or to solve a specific issue. The decision can be made that “the best thing to do” for a specific issue or a particular situation is more important than to stay between the lines and follow the prescribed rules.

Always questioning: Wrong application/usage or unsuitable item? It is useless to blame the item for an inappropriate usage.

A MODEL

- is an abstractions
- is perspective
- represents a part of a world, real or imagined
- represents a concept, and idea

All models are incomplete (thus wrong).

However models are or have to be useful

- to master complexity
- to support thinking
- to facilitate communication

A MODEL is Worth a Thousand Words

Using Models

A model represents an idea, a concept, a process, a structure or a mechanism.

Models are useful because

- they provide (some) insight
- they help to master complexity
- they guide the mind during a thinking process
- they support the communication of an idea or mental construct.

A model reflects a part and/or an aspect of reality or of a mental construct.

It represents an abstraction, a generalisation, a perspective or a part of the reality or of a mental construct.

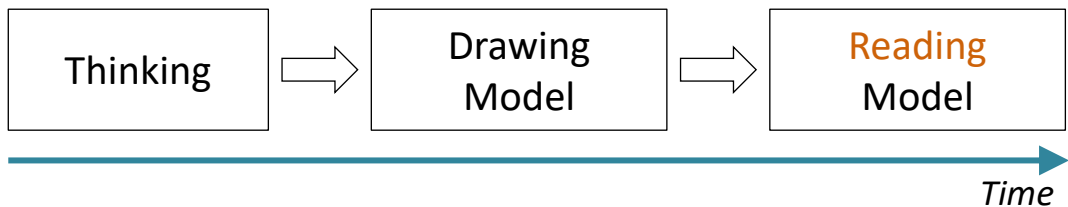
But it always represents a simplification. There is always/often a difference between the model and the reality.

Consequently, strictly spoken, all models are wrong. They have to be interpreted correctly and (often) adapted and applied in a way that suits the specific situation. Or, they may be interpret inappropriately. Neither may we assume they always can/have to be applied “as is” (literally). It is the information or idea that they convey that matters and which should help to take the decisions.

This applies to the models the Analyst makes,
but also to the models in this course.

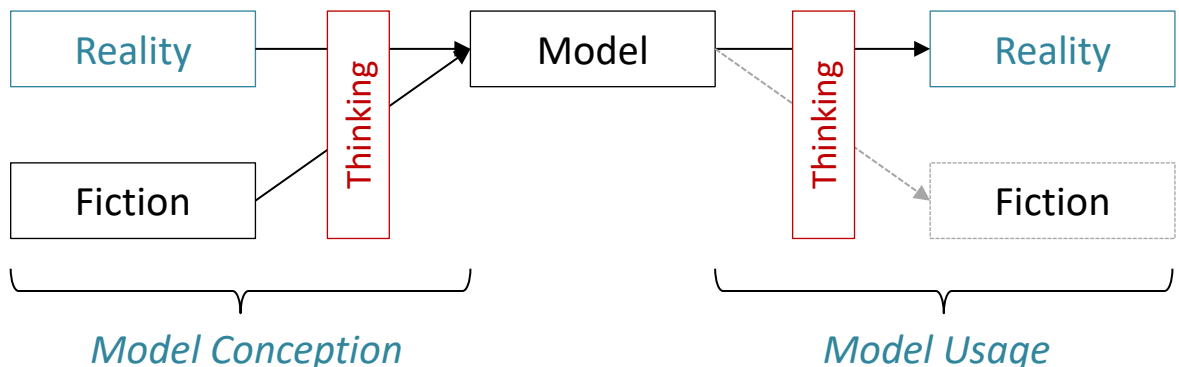
Models is a Basic Technique for Analysts

Model used **as a Communication Tool**



Models are **PRIMARILY a Thinking Tool**

This usage allows to tap into the real power of Models

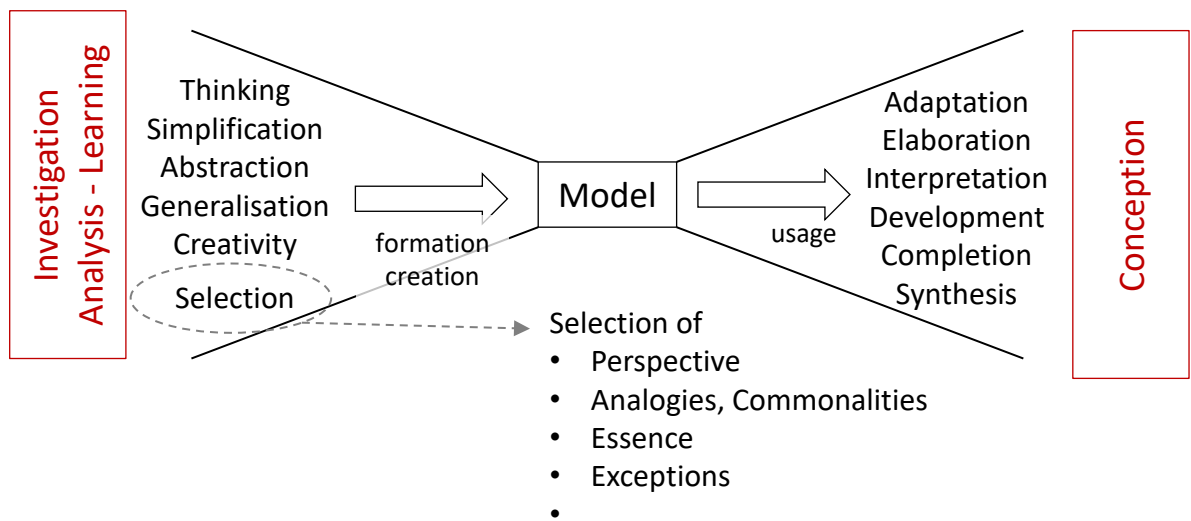
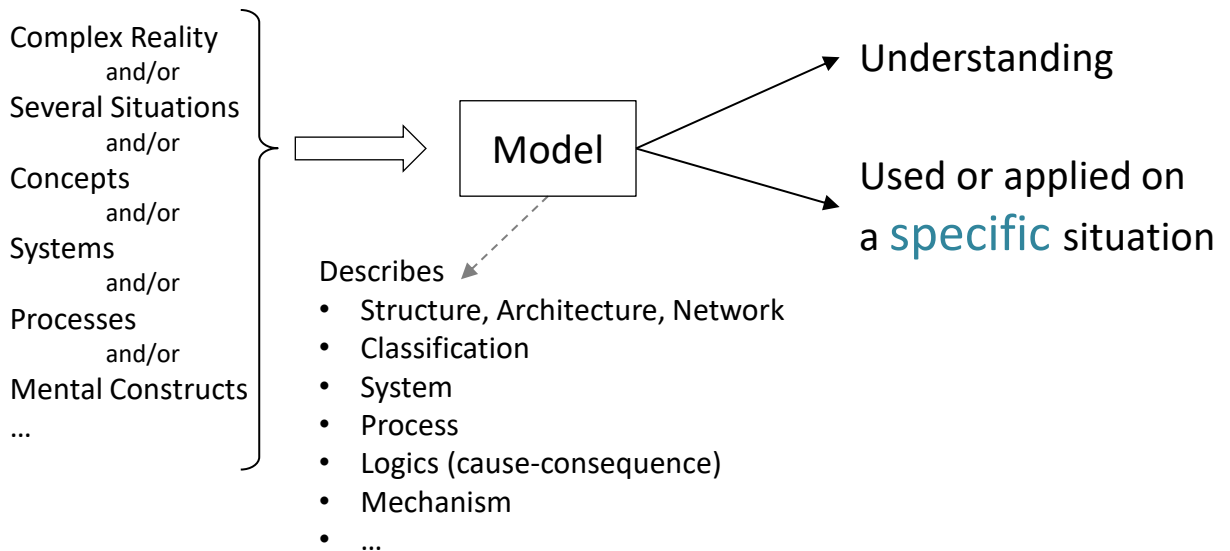
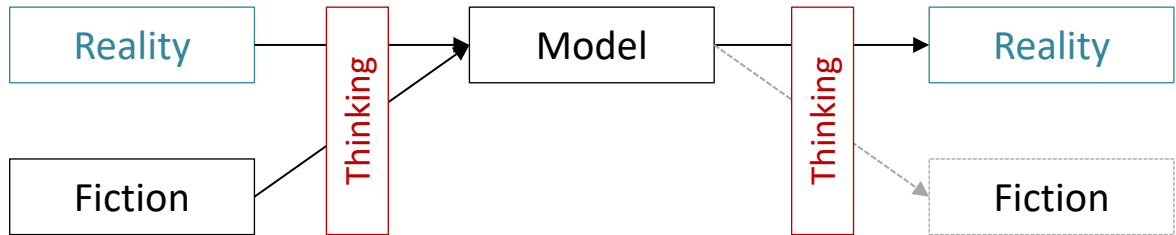


This is a dynamic, an active, usage of the model blending investigation, thinking, exploration, reading and other mental activities with the drawing or 'reading' the model.

Using Models

10/01/2020

Model used as a Thinking Tool

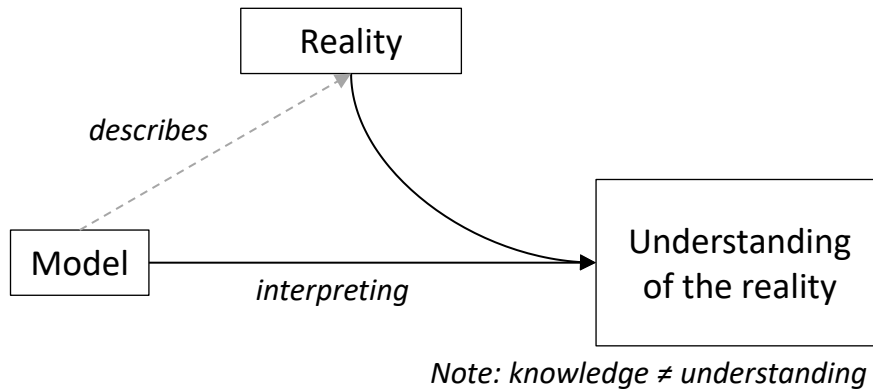


Using Models

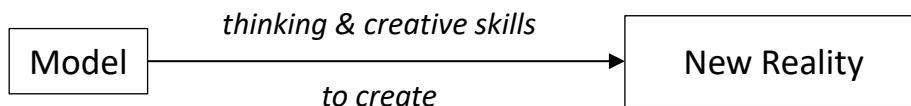
10/01/2020

Model used as a Thinking Tool

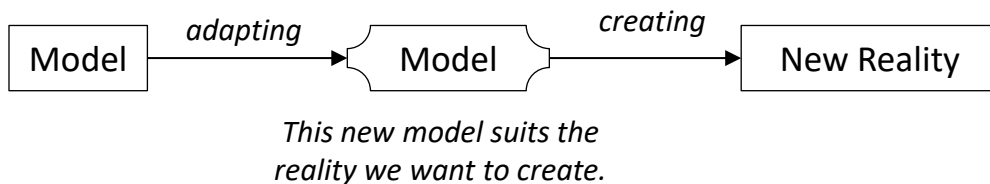
Understanding of a more complex reality



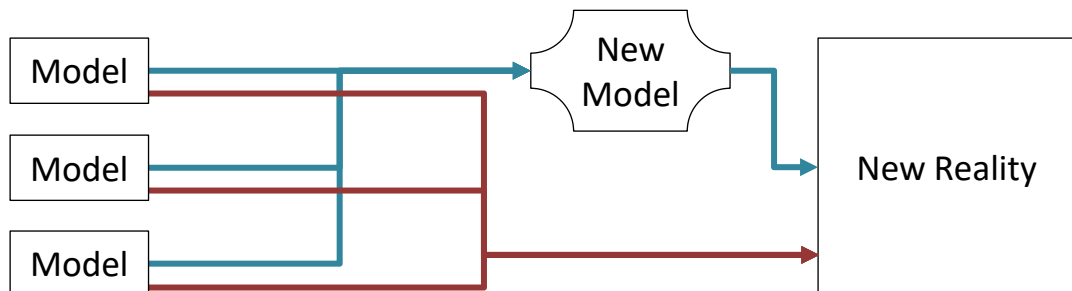
Creating a reality

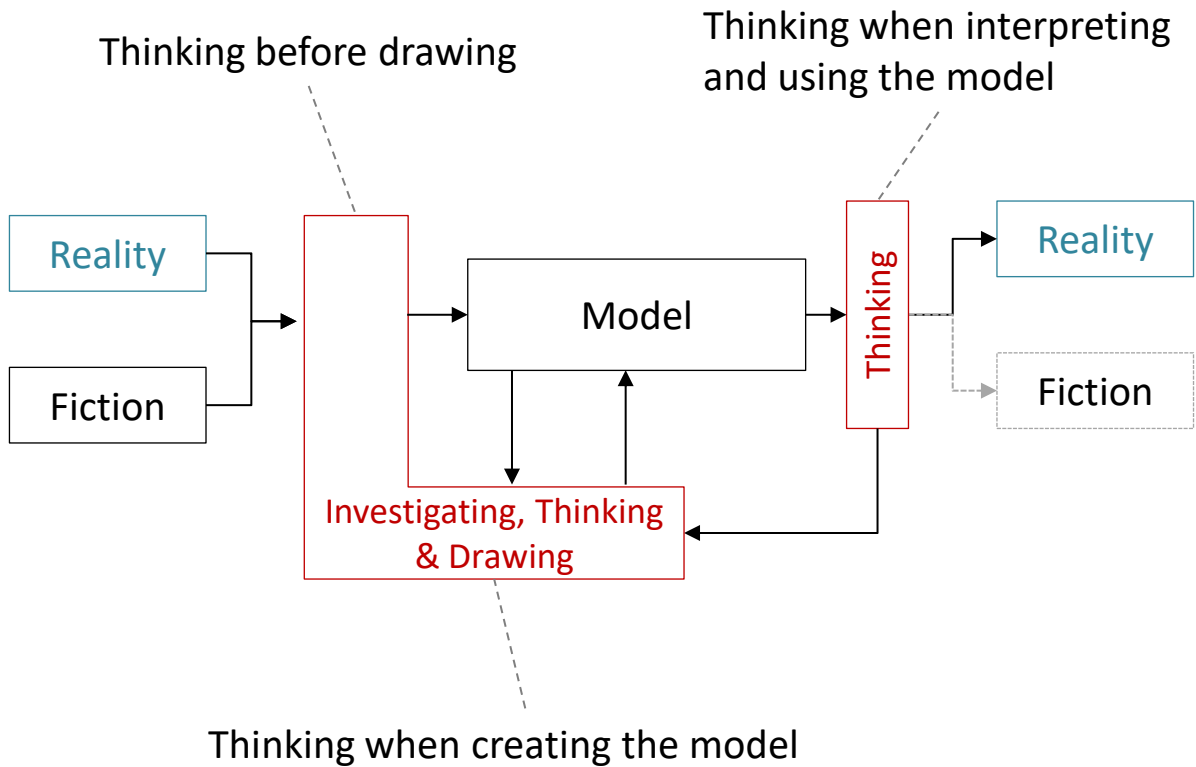


Adapting Models

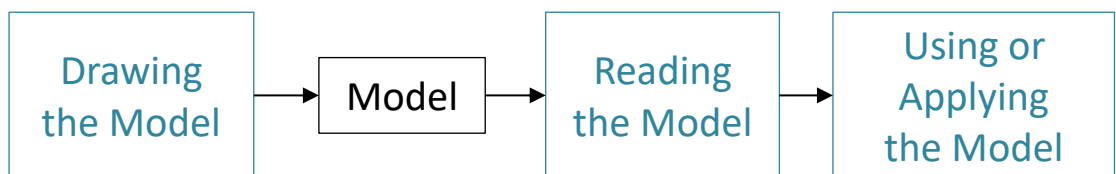


Synthesising Different Models to Create a New Reality





Superficial & Inefficient use of Model



Quality of Application and Usage

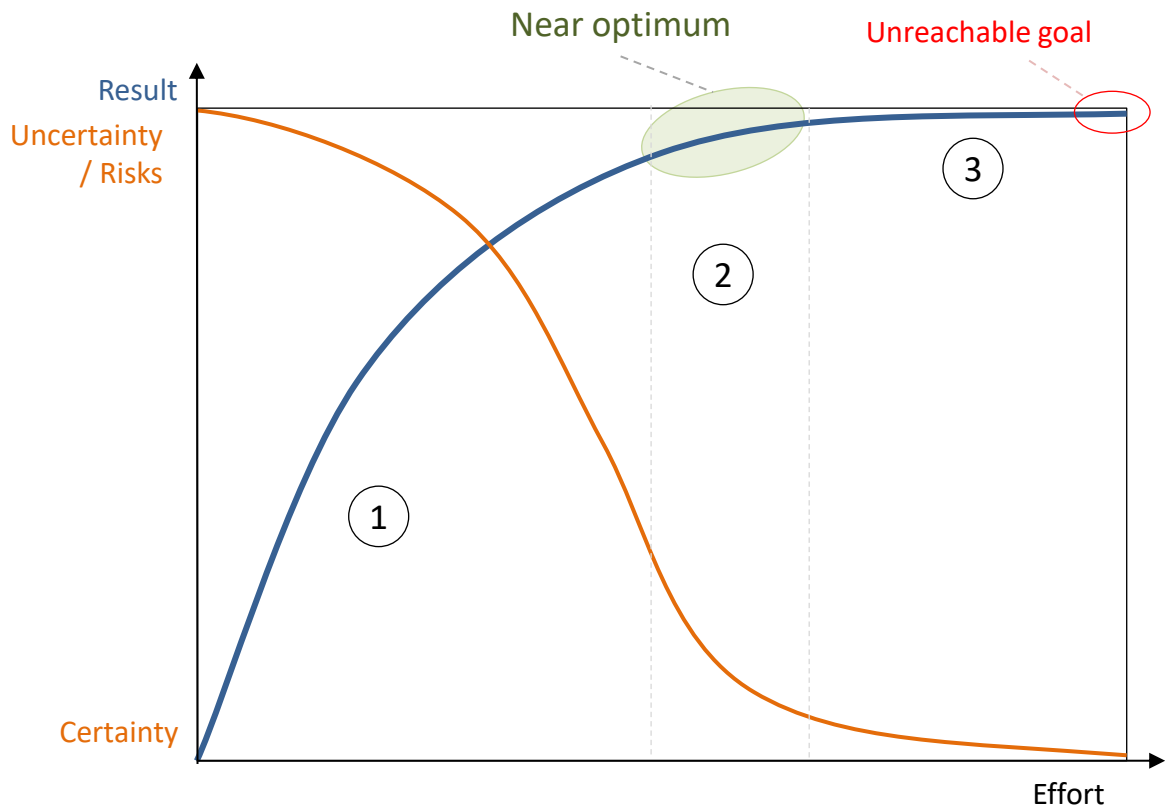
30/10/2018

Application of Theories, Principles, Concepts, Methodology, Methods, Systems, Tools, ...



Principles of the Genuine Effort

30/10/2018



1) Too little effort,
Small result
Inferior result
High uncertainty
and risks

2) Good balance
between effort –
quality results –
uncertainty and
risks

3) Huge additional
effort for little
additional result
and small decrease
of risks and
uncertainty

GOAL:

Executing with the firm intention to get a good result,
not simply for the sake of doing.

Balance between effort and good quality with limited risks and uncertainty

Some major risks can often be eliminated fairly quickly. But a lot of smaller risks (smaller impact) may remain for a while.

At a certain point, most new information will confirm the present understanding and little new insight is acquired or the new insight concerns details. This is the point where we can go on.

The arrival of new information invalidating major insights or having a major impact on the design can never be ruled out. But the likelihood that this happens becomes gradually smaller and smaller.

MYTH, ILLUSION or HYPE	REALITY
<ul style="list-style-type: none">• Easy• Fast• Cheap• Highest Quality• Linear	<ul style="list-style-type: none">• Complex• Arduous job• Slow• Difficult and Risky• Non-Linear

Small and simple software applications or smaller parts, such as software features, are easy, fast and cheap to develop. This is not the nature of most present-day systems in corporate environments.

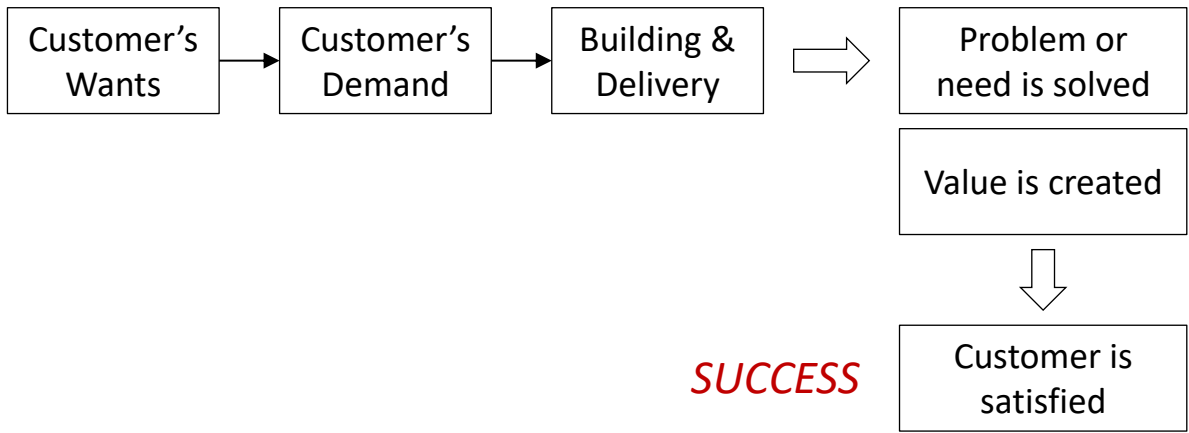
Accepting reality and finding wise ways to deal with it

Danger: oversimplification, simple but wrong answers, short-cuts, ...

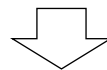
Notes:

Novice may not understand the purpose, value and right use of decisions, techniques or tools. They may unwittingly take wrong decisions and short-cuts creating issues, increasing risks, limiting the solution, technical debts,

This is even more true for laypersons.

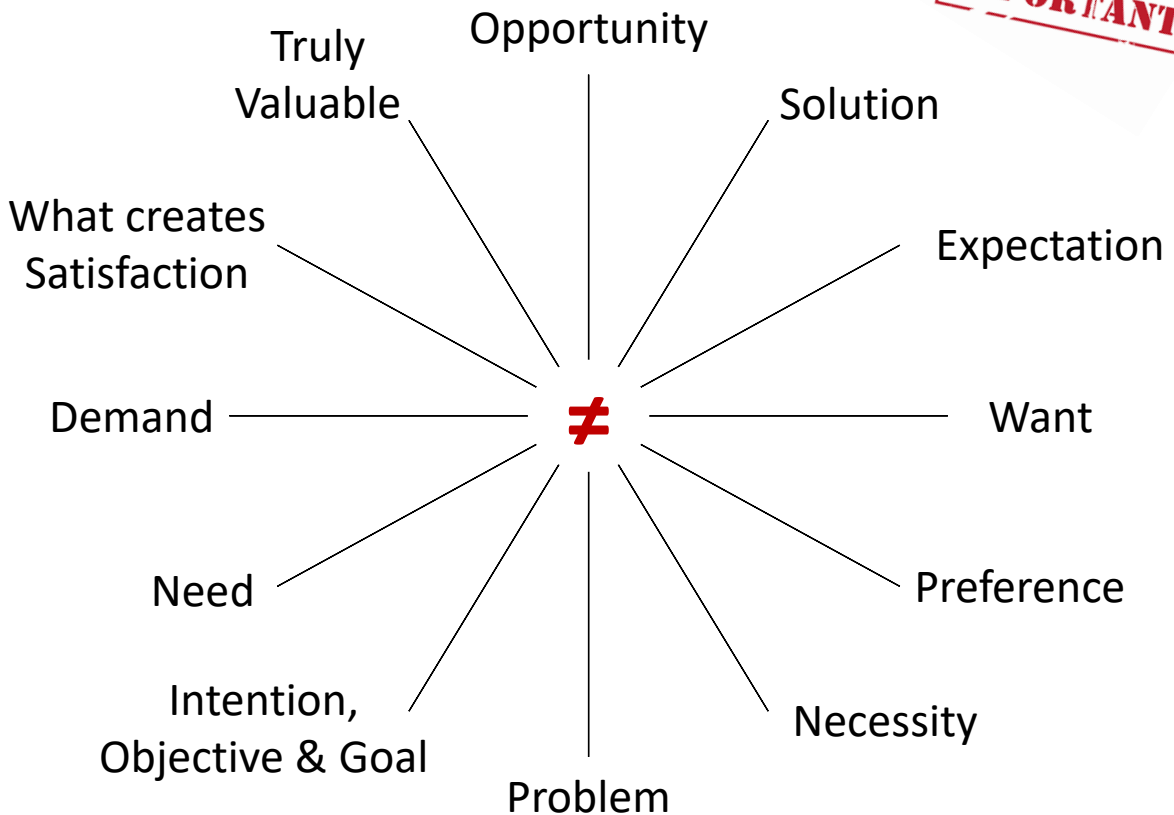


If a customer can articulate what he wants, he is more likely to get it. What the customer wants is articulated in the demand, as well as his desires and preferences. This demand describes the needs and what is needed. Therefore, it expresses what is required. The customer's problem or need is solved by building and implementing a solution which complies with his demand. Value has been created. It is by responding to the customer's demand that his needs and expectations are met and customer's satisfaction is achieved. The product will contribute positively to the overall organisation.



!! MANY ASSUMPTIONS !!

IMPORTANT

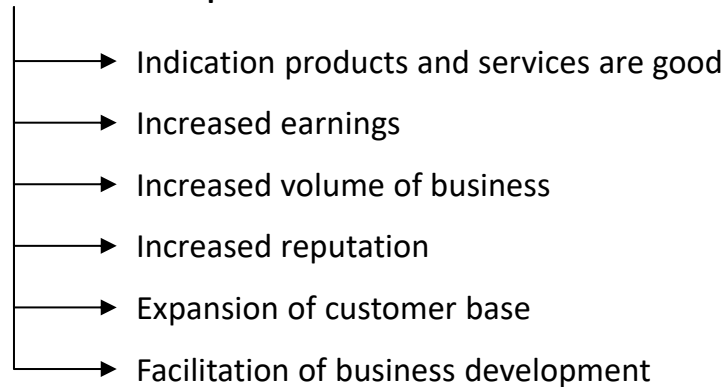


These are totally different concepts !!

AND the correspondence between these elements
may never simply be assumed.

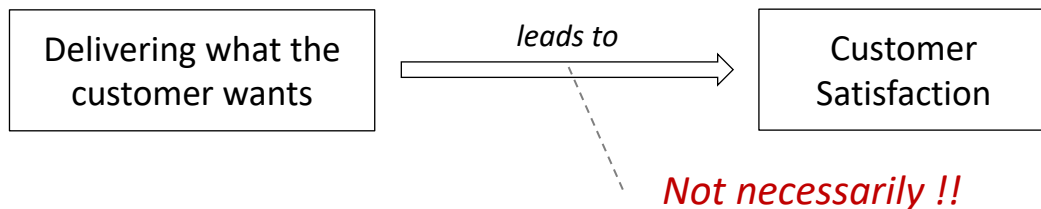
What is wanted is not necessarily what is needed. What is demanded doesn't always reflect what is required to solve the problem or need. It doesn't always allow to meet the expectations. Building accordingly to the demand doesn't always lead to satisfaction. The envisioned solution doesn't always increase the value for the company. And so on.

Customer satisfaction is important for a few reasons:



Nature of Customer Satisfaction is

- Intangible
- Personal
- Variable - Temporary
- Vague
- Unexpressed or partially expressed

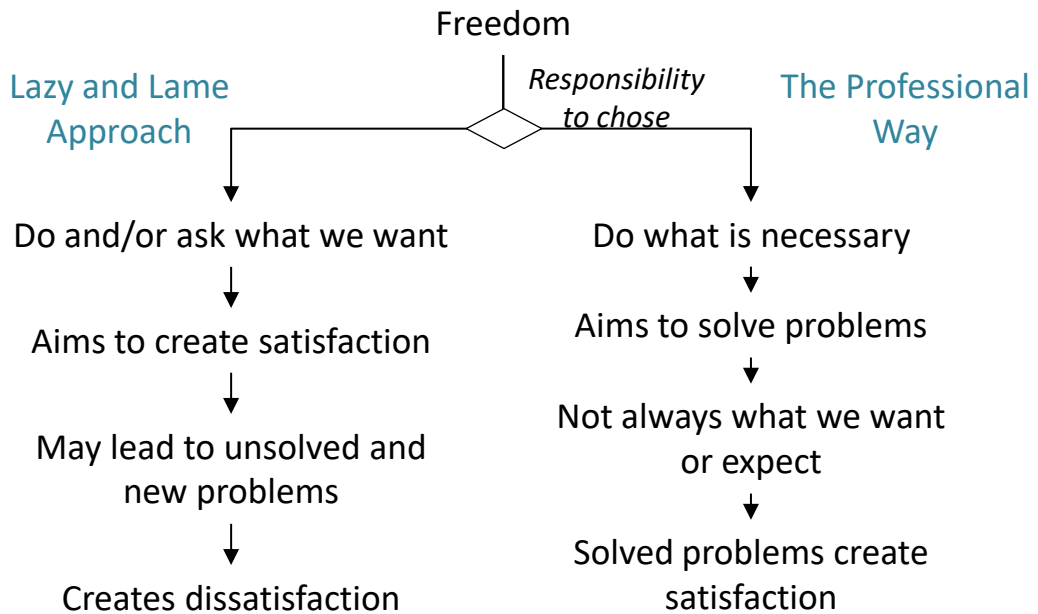


What a customer wants and what will satisfy a customer
are **two different questions**.

Satisfying a customer does not necessarily lead to future decisions or actions from which the supplier can benefit.

If knowledge or insight is missing → the desire is incorrectly identified, still immature, unstable or not well expressed → no customer satisfaction

**“The client doesn’t know what he wants”
Or “he knows what he wants, but it is not what he needs”**



Example 1: Do-It-Yourself Store

Vendor 1: Sells the product the customer asked for, chooses or wants. If the client buys the wrong product, it's his/her problem. The client might be satisfied. If it doesn't solve the problem, the purchase of the product was unnecessary and dissatisfaction will follow.

Vendor 2: Asks the customer what problem (s)he wants to solve with it and what the situation is. The vendor uses his/her knowledge to advice more appropriate products, better solutions and tips. The vendor may sell a different product than asked. Once the problem is well-solved, the client is satisfied about the solved problem AND about the delivered service.

Example 2: Diet and Lifestyle

Many people prefer to eat unhealthy food. Over time, they may start feeling in a bad shape and get sick. They will feel better or heal once they eat the right healthy food and switch to a healthy lifestyle (doing what is necessary, doing the right thing).

Example 3: Planet

We may like the modern life standards and try to maintain it. However, it is unhealthy and unsustainable. It is destructive for the planet. As long as we ignore to do what is necessary, reasonable and responsible, the effects will backfire and we may not like that.

More than the desired "solution",
people want the problem being solved without new problems or annoyances being created.

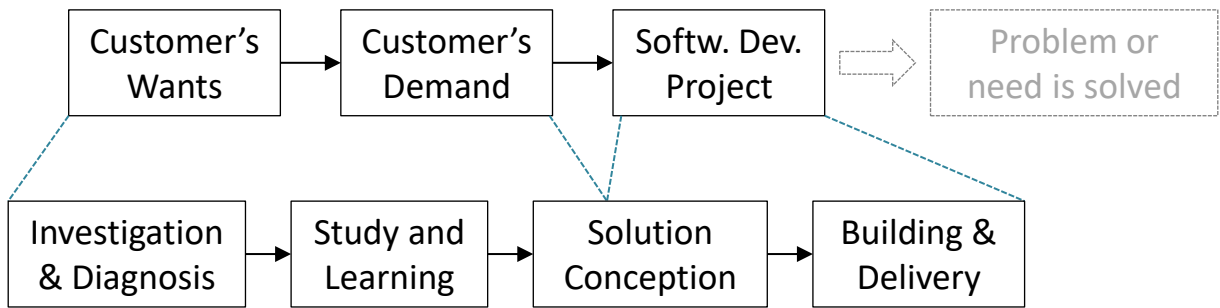
Doing what is necessary is important, doing what is wanted is supplemental.

Both are not necessarily mutually exclusive. Sometimes they can be combined.

THE BUSINESS DEMAND

Expresses Needs and Demands, Wants, Desires, Expectations

- Subjective: Based on an interpretation of the situation and 'diagnosis'
- Often symptomatic (Consequences of problems are noticeable. The experience of the consequences are upsetting. They have to disappear.)
- Can be influenced by emotions; like impatience and frustration
- Different visions may exist (different persons may have different wants, expectations, desires, ...)
- Can be the result of a compromise (ends up loosely related to the real problem)
- Lack of alternatives (have they been investigated?)
- Hard to capture
- Can be misaligned or conflicting
- Based on uncertain facts (partial insight)
- Can easily be changed, even in a blink of an eye (while a system's lifespan is several years)
- Can be unrealistic
- Demand may not be inline with intentions, wants, expectations, ...
- Vague and partly unexpressed (not all wants, intentions, reasons, desires, expectations, ...)
- Based on personal insight of the situation, limited perspective and scope
- Based on understanding of the possibilities and limitations of software systems, which may be a flawed
- Based on understanding of how to deal with information, which may be based on limited insight in this matter, false beliefs and assumptions
- No guarantee that it constitutes a solution
- No guarantee that it leads to added value for the company
- Can be desired, wanted and asked, and, once built, not used.
- May undermine the company (truly damaging)



Responsibility of the Customer in this line of thinking

- Posing the **diagnosis**: detection of problem, limitation, opportunity
- Determining the **objective**, intention, goal and scope
- Articulating the “**needs**” and/or “**wants**”:
 - **Requirements**
 - **Conception** of nature of the solution, the role and main functions, solution description in business terms
- **Evaluating** the solution to accept it or refuse it

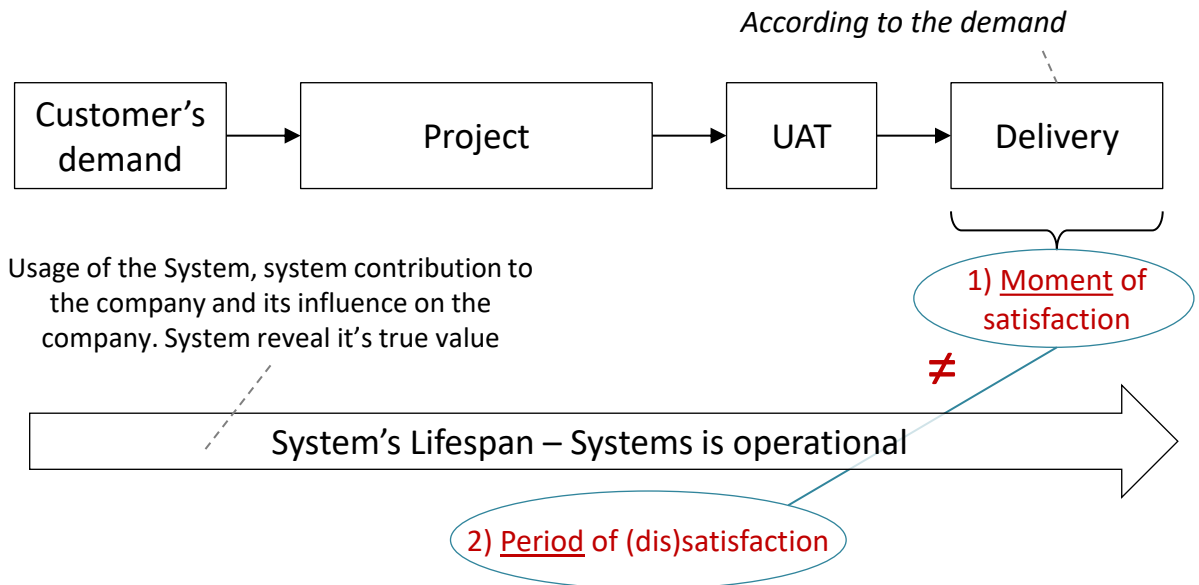
= ANALYSIS !!



*Misalignment between
responsibility and competencies*

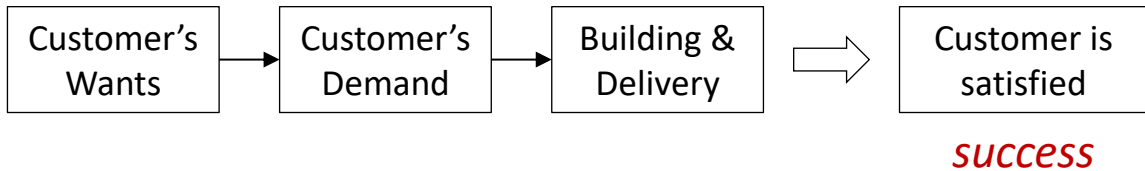
Brief overview of required skills to pose a diagnosis and to conceive a solution (in information issues in corporate environments)

- Business knowledge and situational expertise
- Overall business and company knowledge
- Insight in the company's systems and processes
- Profound understanding of “Information”
- Profound understanding of “Systems”
- Real Problem Solving skills
- Thinking like an Engineer



- Some stakeholders may be satisfied, while others are not.
- Stakeholders may be satisfied, while the 'solution' doesn't solve the problem, create new ones, bring in risks and limits, is inefficient, is awkwardly designed or does not fully exploit possibilities.
- The competencies of stakeholders is only one perspective to judge a solution. It is limited.
- The satisfaction at the delivery (1) does not guarantee or imply a satisfaction during its operational lifetime (2)
- Even if customer satisfaction is obtained at the delivery, the day after the customer may change his mind and not be satisfied anymore. Systems exists beyond customer's satisfaction.
- A system, a good system, is tangible. Customer satisfaction is intangible. Value?
- Successes are too sporadic and won't last. Impossible to meet on regular basis to create permanent satisfaction.
- Focussing on satisfying a "delivery according to a demand" is way more easy than seeking to get value and satisfaction over the long term.
- Seeking to have appropriate systems and solutions requires a different way of thinking, organisational setting and approach from what is required to respond to a pre-established demand.

CREATION OF CUSTOMER'S SATISFACTION

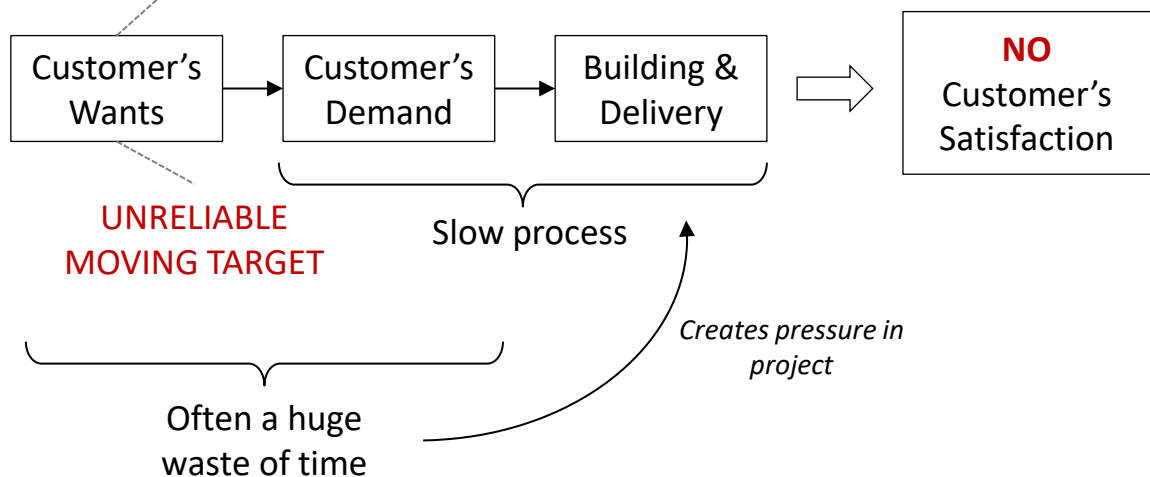


Successes are too sporadic and won't last. Impossible to meet on regular basis to create permanent satisfaction.

Even if customer satisfaction is obtained at the delivery, the day after the customer may change his mind and not be satisfied anymore.

PROBLEM:

Vague, intangible, changeable, linked to emotions, different visions (no unity), unreliable, interpretation, alternatives, hard to capture, can be unrealistic, can be misaligned, uncertain, can be conflicting, local, short term, thinking small, personal, subjective, based on interpretation of reality...

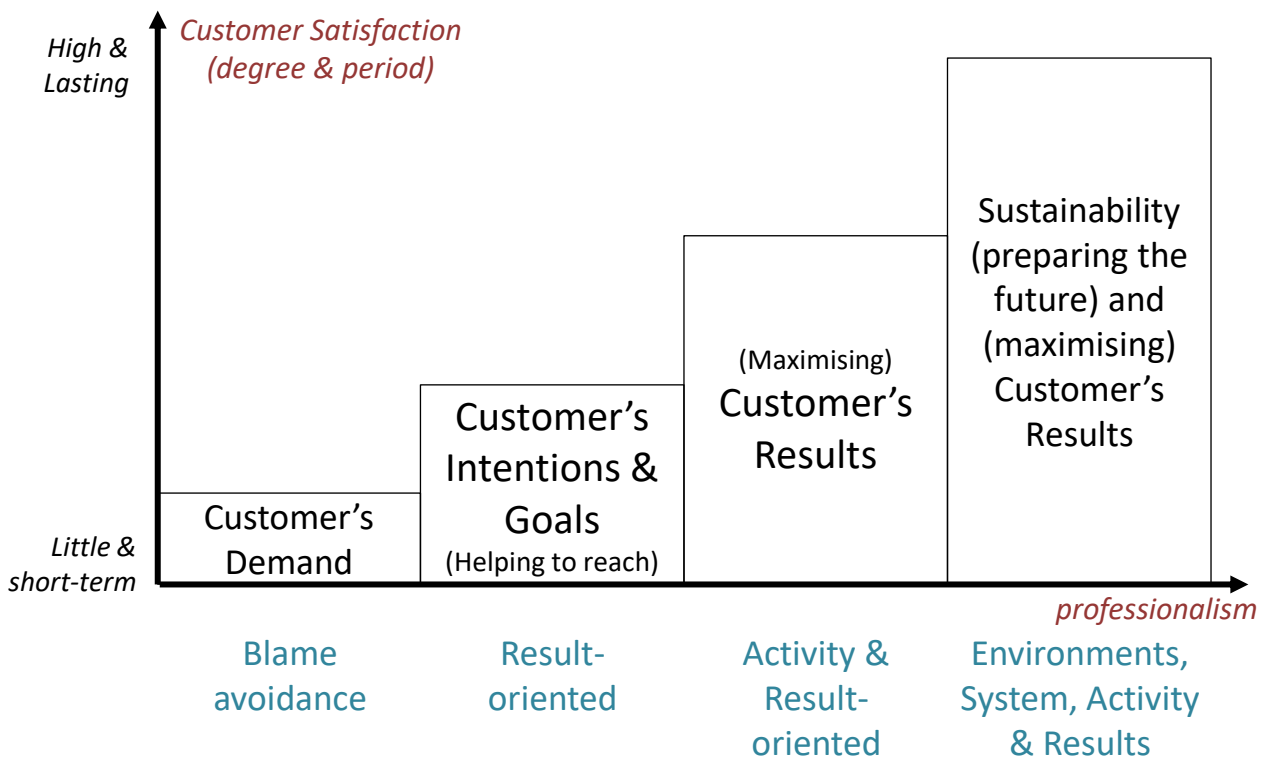


"If I had asked people what they wanted, they would have said faster horses."
- Henry Ford

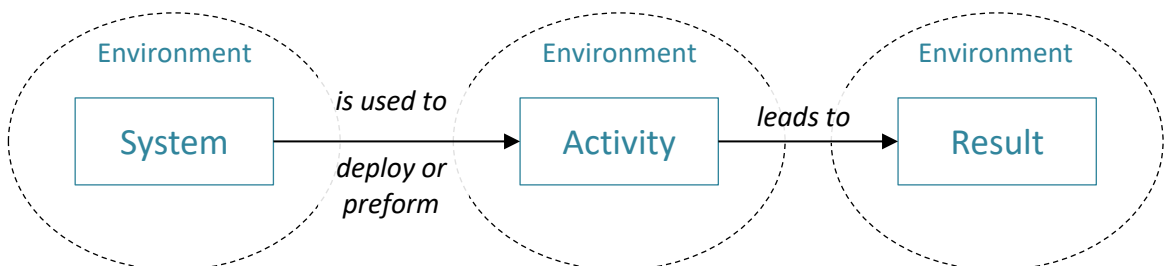
WHAT DRIVES THE PROJECT TEAM?

- On what is being focussed?
- How are the mission and project goal defined?
- What is expected?

WHAT SHOULD DRIVE THE ANALYST ?



Result? Activity? System? Environments ? ... see diagram below



WHO TO SERVE?

Is the **User** the client?

No. Software applications may facilitate the work of software users. They benefit from software application to do their job. However, they are part of the Information System. Their work needs to create value for someone else.

Is the **Sponsor** (sponsoring business stakeholder) the client?

Yes and no. Projects can be funded with the budget of a business unit, department or service. Would it be different if the money came from a budget from the top of the company or from the IT department itself? The money, wherever it comes from, belongs to the company and the company can fund projects in a different ways. The product of the project should benefit to his/her organisational unit. Information system should belong to the company, particularly if they or their data is shared across the organisation. A sponsor is rather a key stakeholder than a client.

The client is not simply the one who asks or fund.

Who Are the Real Clients?

30/10/2018

1. EXTERNAL CLIENT

The client outside the company to whom the company delivers products and services. The real client can be a person, or organisational unit, member of the organisation. The company's main role is, its main mission, is to serve society.

1. THE COMPANY (also 1)

The company, as the overall entity, is the second real main client. This overall system has to function well and to thrive.

How useful is it to have one organisational unit that is doing well, while the entire company goes bankruptcy ?

All the funds come from the company. The organisational units received budgets from the company.

2. INTERNAL CLIENT

Higher organisational units defining the purpose, values, directions, strategy and higher goals for the business are also clients. (ex. Business Units)

2. INFORMATICS DEPARTMENT (also 2)

Huge parts of larger companies are automated. This tendency to automate as much work as possible will keep on. The role of the Informatics Department will thus increase. This also means that it has to prepare and to strengthen itself. An Informatics Department finding itself in a position where it is unable to conceive appropriate information solutions and implement them or to manage and adapt the implemented information systems constitute a major danger for the company.

3. LOWER LEVEL ORGANISATIONAL UNITS

4. USERS GROUPS AND USERS

Users shouldn't be ignored. They possess valuable knowledge. Their needs and ability to work are important. However, their needs, suggestions and requests should be taken in consideration together with all other necessities, objectives and priorities of more important clients.

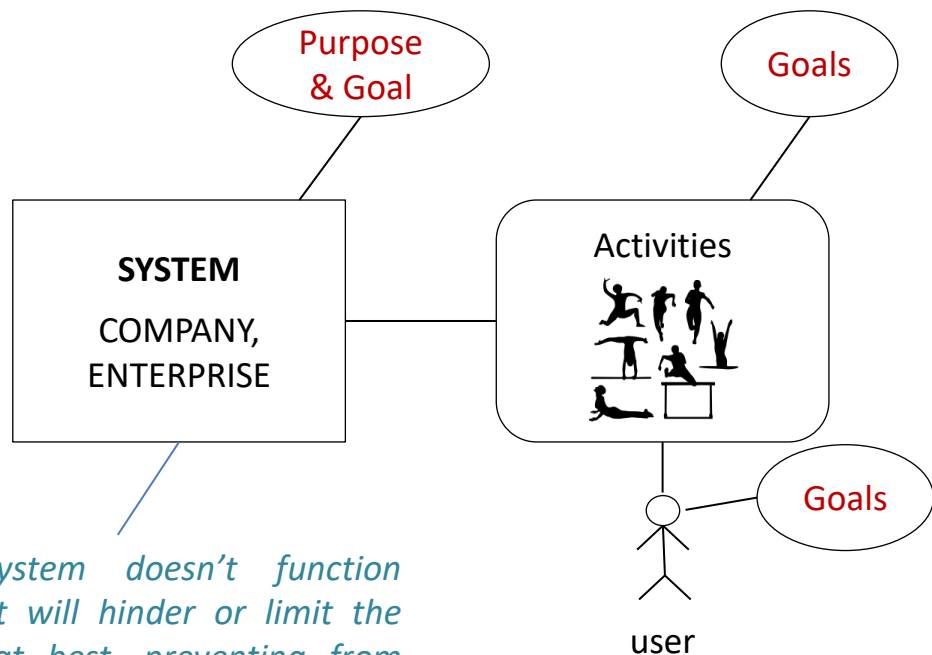
Higher Goals

30/10/2018

CUSTOMERS ↔ **GOALS**
have

Main Customers =

- **ENTERPRISE** (company's body)
 - **COMPANY**
 - **COMPANY's CUSTOMERS**
 - **SOCIETY**
- Often and mainly hierarchical
 - Some subset of goals may contain a network or circular pattern



If the system doesn't function properly, it will hinder or limit the activities at best, preventing from reaching the objectives. It may lead to bad products and services for the external clients.

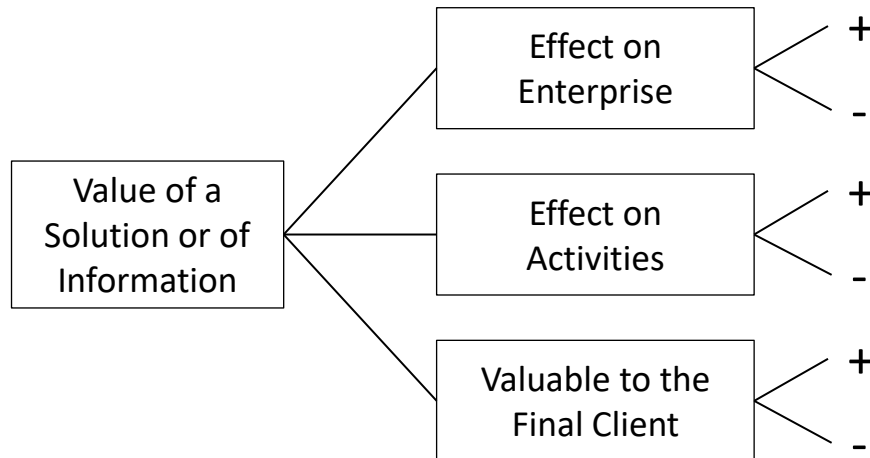
Notes:

A taxi company can't thrive if the car (taxi = system) is old, dirty, worn out, regularly broken,₄₆₈

Value Creation (1)

30/10/2018

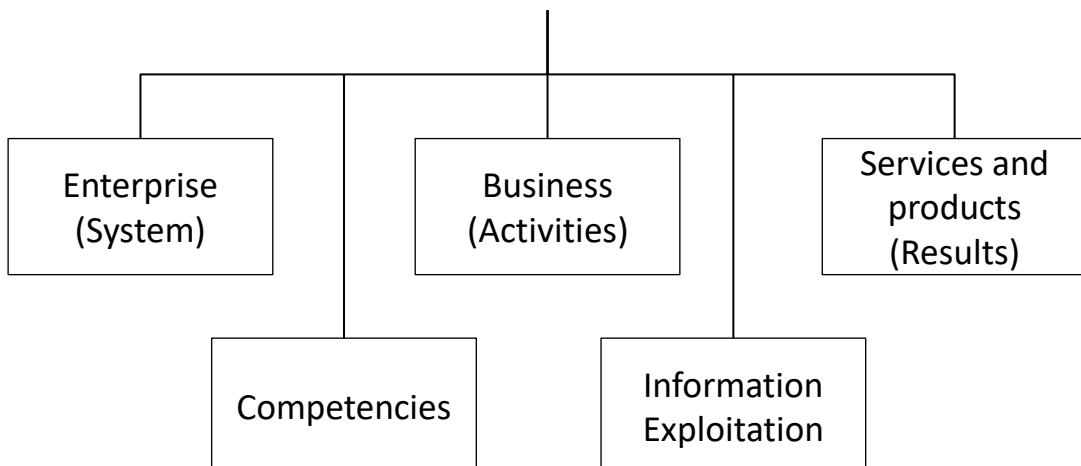
Value is defined by the final **net positive contribution**



The **value for final (external) clients** can be estimated by interest, sales, price ready to be paid, ...

Evaluating the **“value” for the company** requires a deep **understanding of all the effects** an item or solution has on the company.

Major Areas of Value Creation



Ideas with little solid ground FALSELY affecting the estimation of value of a system

The product/solution/feature has value because:

- it is the goal and mission of the project
- it is the delivery of what is agreed on
- the product correspond with the demand
- the software application works as expected, looks nice, is attractive, ...
- it is our product, the project's product, ...
- a lot of effort spent building the product
- building it or acquiring it costed a lot
- the company has now new features, new processes, new software
- the system is adapted or expanded. Features have been added.
- It can do more.
- the new is better than the old, the old was outdated
- the product is desired
- the product or new solution facilitates the job of the end-user
- the end-users, stakeholders, sponsor or business community are satisfied
- the product is useful to the company
- our people, the project team members, are competent
- everybody implements it these days
- it has been advised by solution providers
- the builder is proud of his work
- ...

Possible Obvious Effects

- New functionalities
- Information needs being solved
- Additional Capabilities
- Increased Capacity
- Improving the products and services to the final clients
- Improved information exploitation
- Innovation
- Better KPI results (like reduced throughput time)
- Better alignment with the environment
- Lowered cost
- Reduced waste
- New competencies
- ...

More Possible Hidden Effects

- Increased maintenance and/or operational cost
- Increased fragmentation
- Inefficiencies have been introduced or shifted to elsewhere
- Incoherencies, conflicting logic and bugs,
- New weaknesses, bottlenecks, SPOF's, risks have been introduced
- New hindrances, difficulties and obstacles in usage
- New hindrances and limitations (impossibilities, closing doors) in evolvability
- Increased waste
- Solving consequences, masking the causes (real problem)
- Increased complexity
- Loss of knowledge and comprehension
- Decreased manageability
- Hidden mechanisms undermining the system from within
- Risks of misuse and abuse
- ...

We can't judge a gift based on its wrapping paper.

A solution or system can't be judge on superficial facts, impressions or unfounded criteria.

Main Aspects to Value of System (non exhaustive)

- Purpose is ethical, useful and accepted
- Fulfilling its role swiftly and with sufficient capacity
- Solves root causes or satisfies true needs
- Producing benefits and minimising costs
- Design intent in line with purpose
- Elegantly designed
- Contributes positively to the supra-system
- Respects supra-system
- Respects the environments
- Respecting harmony in the system and in the supra-system
- Respects system's qualities like manageability, evolvability, ...
- Acceptance by people
- No degradation mechanisms or effects

ISED MAM

Information Systems Engineering Discipline Maturity Awareness Model

V1.1 - 05/10/2017

Download ISED MAM:

<https://goo.gl/noPwQp>

TYPE PRODUCT

Information component

Structurally integrated system of software systems

Software systems landscape

Software application

Software code

Business info. (usage is known)

IT data - Knowledge & Intelligence

Source of innovation

One of the main critical resources (usage unknown)

INFORMATION

TRIGGER

Leading: Mainly planned and proactive

Mix of planned and problem, need & demand

Reactive: Problem, need, demand

End-user's (EU) objectives

Personal objectives

Deliver software

Acquisition

Predefined solution

Post-implementation

Design features

Key design issue

Local innovation, little & local impact

Innovation in global functioning, in product and in services

Architectural integration

Functional integration

Holistic integration

INTEGRATION

SOLUTION DEFINITION APPROACH

End-to-end analysis based approach

Objective-driven analysis approach

Continuous holistic analysis triggering initiatives

OBJECTIVE

Company's long term prosperity

Business success

IT dept. success

Stakeholder's success

Product's obj.

Project's obj.

OUTCOME

Strengthening of the company for long term

Streamline cross-organisation operations

Facilitated EU's work

Facilitated Local operations

Driving the company

30/10/2018

INNOVATION

Strategy: Innovation the drives business

HUMANS

- Able to set and define goals
- Have intentions
- Able to execute high level instructions
- Able to deal with imprecision
- Able to interpret
- Able to evaluate
- Able to adapt
- Have specific skills and knowledge
- Can learn and accumulate experience
- Are creative
- Are irregular
- Have preferences and dislikes
- Is physical active and mobile
- Have social relations, collaboration
- Have emotions
- Have intuitions
- Have a behaviour, a character, ego
- Volatile memory
- May shifts in focus
- May have misunderstandings
- May make mistakes
- May have antipathies, conflicts
- Can be tired or sick

COMPUTERS / SOFTWARE

- Number crunchers
- Executes complete, detailed, formal sets of elementary instructions
- Perform repetitive work
- Is extremely fast
- Requires a huge degree of coherence
- Very precise
- May contain very abstract structures and mechanisms
- More constant, linear
- Need pre-defined logic
- Is engineered
- Can gathered immense amounts of information
- Can interact with many systems and people at once
- Can have bugs (bugs are undesired and unintended programmed logic)
- Overloading can “make it slow”
- Can work 24/24 – 7/7
- Can be very reliable (more than people)
- Can break down

Reflections

What are the consequences of these differences?

Can we simply copy (w/o adaptation) work done in the human world to the computer world?

Can we design computerised solutions by not taking the characteristics (strengths, weaknesses, possibilities, limits, risks, ...) of computers and software into account? Without a thorough understanding of the specificities of this world?

Do we need a different way of thinking when designing computerised solutions?

Notes

Computers may be equipped with logic simulating human qualities, like rules system, fuzzy logic, back tracking, Artificial Intelligence (AI). But most systems aren't that far yet.

An IS has a “human part” and a “computer/software” part. Both have to collaborate. Both have to be engineered to form a single system and to work in harmony.

Why Goals ?

- Easier to agree on goals than on processes and solutions
- Allows to **focus** on a **desired state**, while leaving the way and the **solution** to reach it **open** (in opposition to defining the desired solution)
- Different paths may lead to same objective
- Allowing everyone to move in same direction, to work on the same
- Maintain focus on what matters
- Allowing to move faster, using lesser resources because it increase effectivity, efficiency and coherency

Which Goals ?

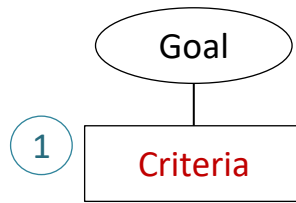
1. Lifetime goals
2. Long Term goals
3. Short Term goals

1. Fixed / Lifetime goals
2. Strategic goals
3. Tactical goals
4. Operational goals

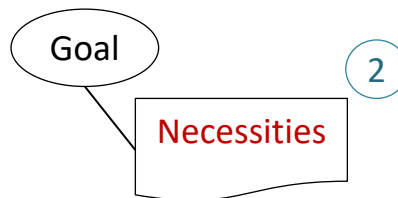
Further description

- Concerned area
- For / linked to specific entity/-ies
- Concerned aspects (quality, communication, pricing, ...)
- Difficulty level
- Impact ('amount' and where)
- Importance & Priority
- Risk level
- Pre-conditions
- Forces (helping forces and opposed forces)
- Obstacles
- Immutabilities
- Limits
- Estimated time to reach
- Estimated necessary resources
- Estimated benefits

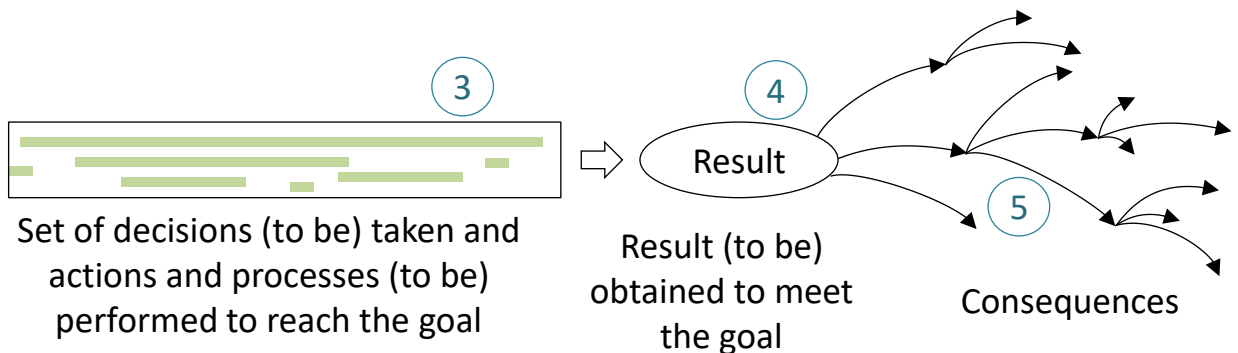
5 Key parts of a goal



A goal has to be **described**
Some has (SMART) **criteria**



Committed Goals creates necessities;
(critical) requirements

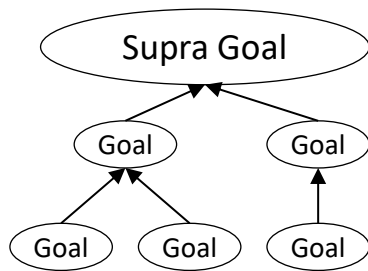


Reaching the goal means obtaining a result. This brings about **consequences and effects** in different areas, in different ways at different time. Intended or not intended; predicted or not predicted; known or unknown; with variable duration; reversible or not; opening new opportunities, hindering or closing doors; ...

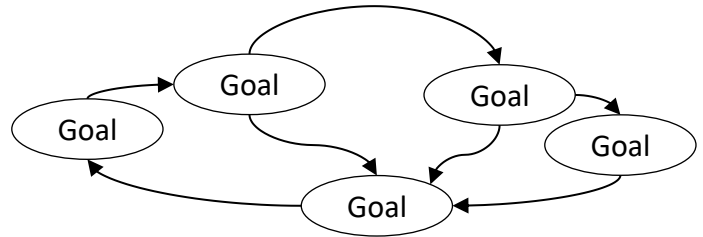
No effect doesn't bring about a longer chain of effects.

We have to look further than the result and its obvious effects.

- Critical / Key goal:** must be reached.
- Optional goal:** supports higher goals, facilitates reaching them, allow to better reaching them or may offer additional benefits.
- Alternative goals:** many paths possible to reach a goal; may require a decision or choice.
- Stepping stone goal:** When this goal is reach, the organisation is in a state of readiness or more comfortable state from which it can start working on next goals.
- Soft goal:** It is impossible to determine measurable criteria for soft goals. They can only be described and estimated.
- Synergic goal:** Achieving both goals reinforce their individual effect.
-

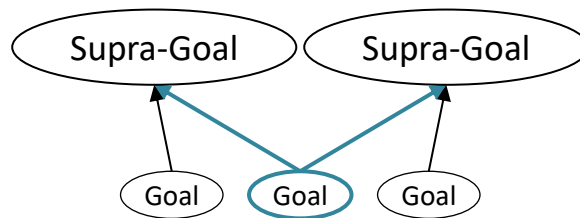


HIERARCHY of goals
Vertical view
(many layers possible)

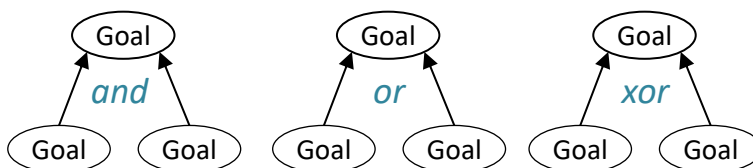


NETWORK of Goals
Horizontal view

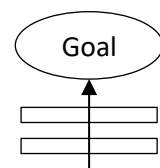
It is not forbidden, in some circumstances, to incorporate higher & supra-goals into a network view of lower goals (or to group child-goals of a parent-goal).



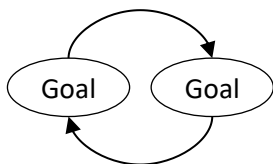
A (sub-)goal may contribute to different supra-goals



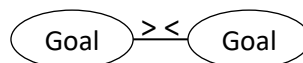
and, or, xor



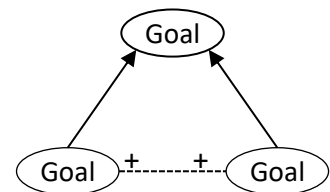
Obstacles



Reinforcing goals

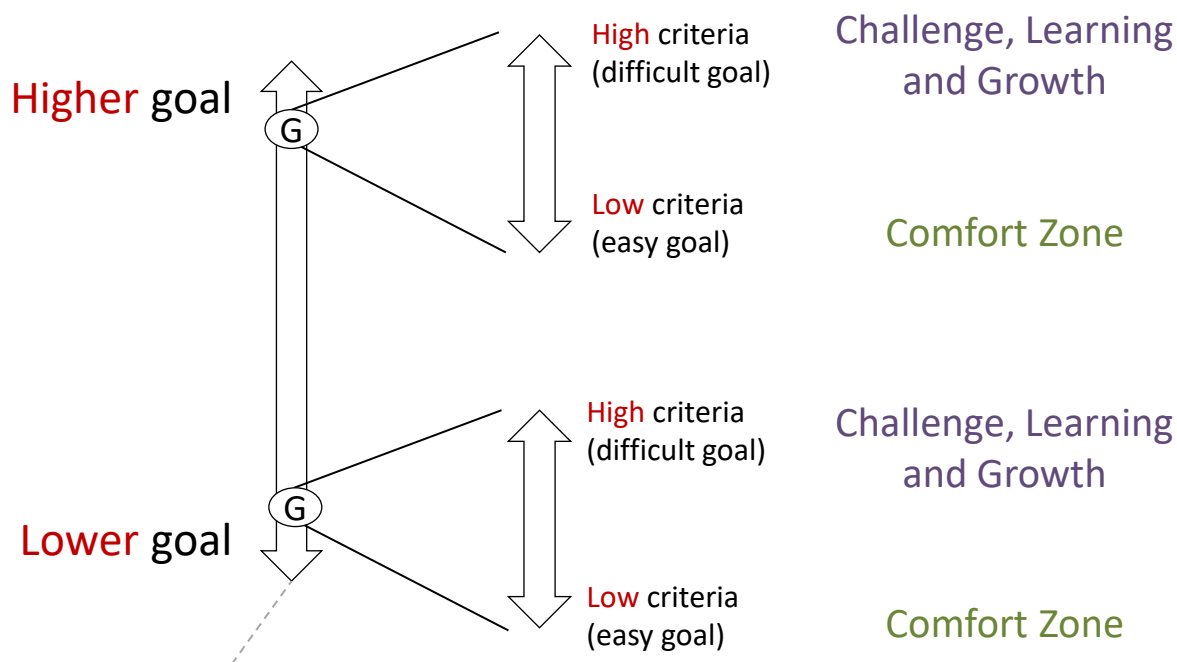


Conflicting goals

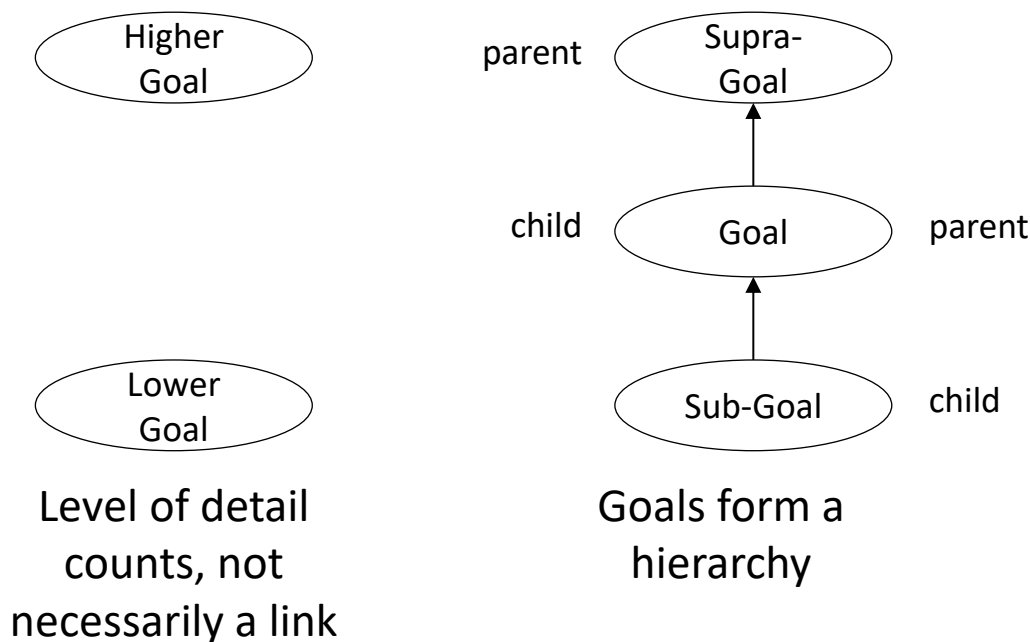


A goal may facilitate (positive influence) or hinder (negative influence) reaching another goal.

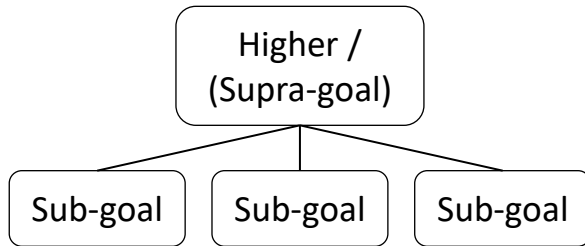
Goals



Overall/local goal, Broadness of scope,
Level of detail & timespan (long versus short)



FOCUSSING ON HIGHER GOALS



**Aiming for higher goals
leads to
greater achievements**

More difficult to achieve – more demanding

- Requires different roles, focus, approach, methods, techniques and tools
- Requires **higher norms**
- Requires **stronger competencies**
- Requires **different beliefs (or world view)**
- Requires a more **global and holistic view**
- Much **more thinking**
- Tests the competencies
- Probably more
 - Aspects to take into account
 - Contradictions
 - Uncertainty
 - Assumptions
 - Complexity, difficulty

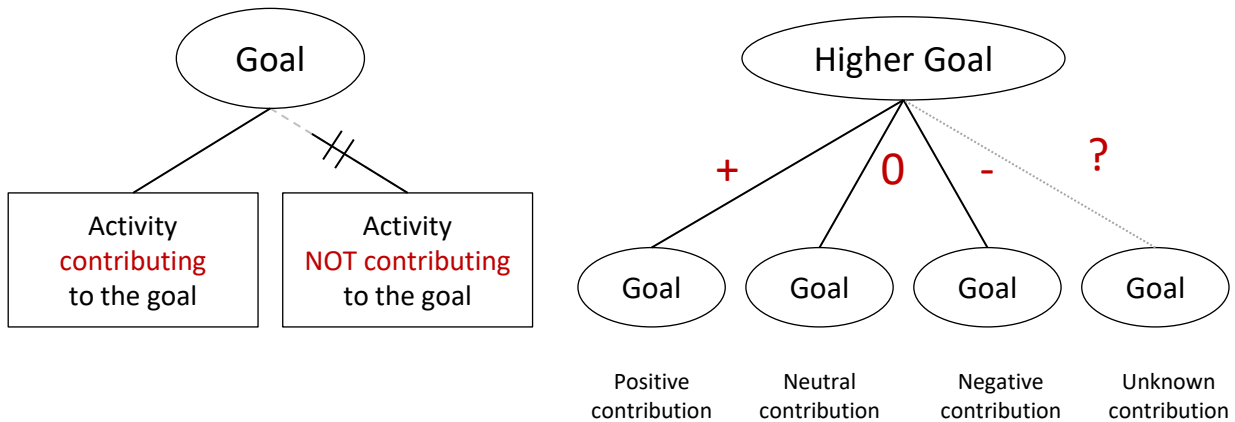
FOCUSSING ON LOWER (LOCAL/EASIER) GOALS ONLY

Assumptions:

- **Reaching lower (local) goals**
 - **is beneficial for the company.**
 - **equals progress.**
 - **helps getting ahead of the competition.**
- **Reaching successfully a goal, makes us more successful.**

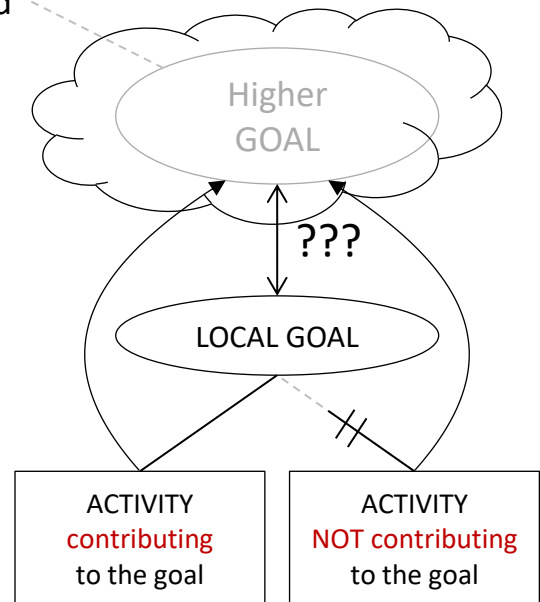
Are we more successful by setting lower goals?

- Easier, quicker, lesser risky, faster 'successes'
- More local thinking
- Lower norms may be used
- Little pressure to improve our skills
- Higher risk for internal hidden issues (under the surface)
- Unexpected effects, inefficiencies, ineffective, lack of direction, ...



If we act and take decisions with or without taking into account a sub-goal (local/lower goal), but without knowing the higher goals (or while ignoring them), we can't know if activities and decisions will contribute to them or not.

Unknown or ignored



Work driven by lower local, goals is unlikely to be coherent, efficient, effective and to reach any significant goal.

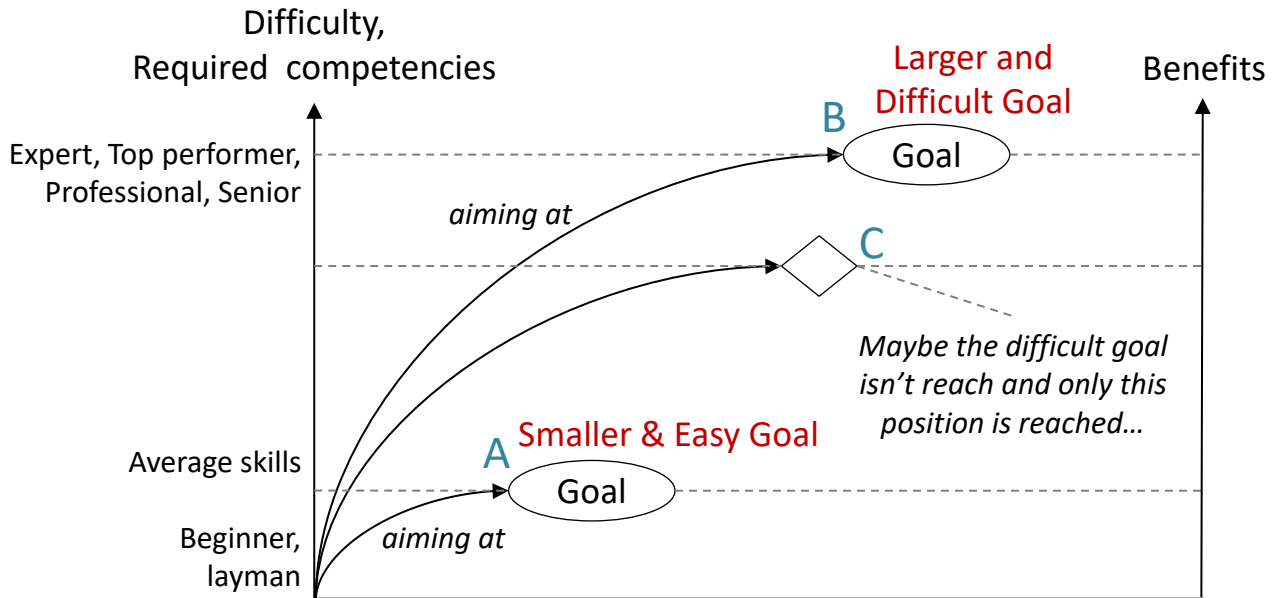
Activities contributing to the lower goal (sub-goal), ignoring the higher goals, may unwittingly work against the higher goal(s).

Assumption:

If all sub-goals are met, then the higher goal is achieved.

Aiming for Difficult Goals

17/09/2019

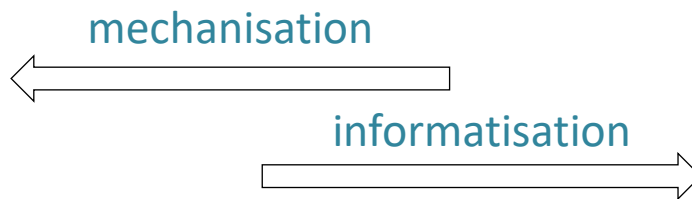


- If we aim at a smaller and easy goal (A) than that goal can and will be reached. The benefits of reaching that goal are rather low. Basic skills may suffice. Anyone can do this. If the endeavour is too easy, not much will be learned.
- Aiming at a larger and more difficult goal (B) and reaching the goal successfully will create yield greater benefits. This requires professional skills, expert's skills, skills of a senior profile.
- If we fail to reach the goal (B) and reach only point (C) then the obtained benefits will be somewhat lesser. But these benefits will still be larger than the benefits obtained by reaching goal the lower goal (A). And at least progress has been made and more interesting lessons can be learned.

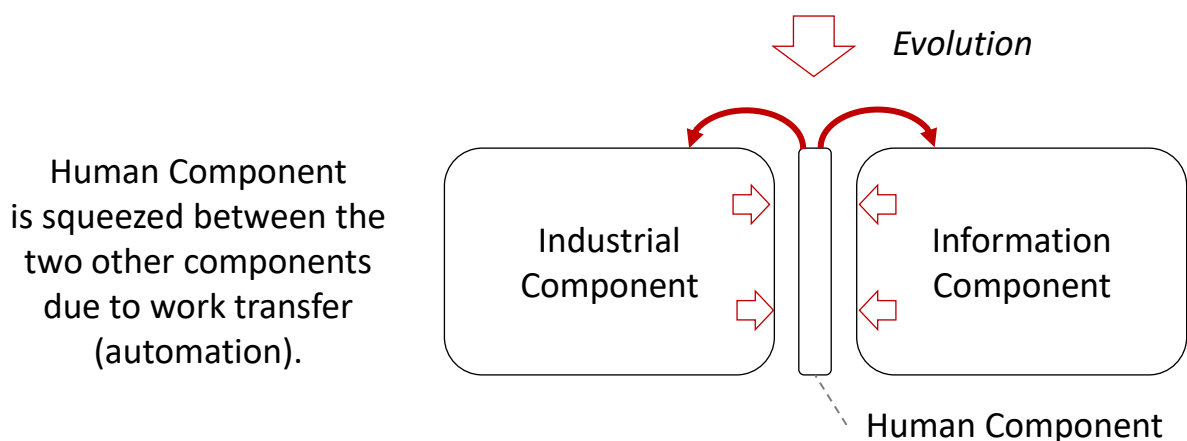
Notes:

In some cases failing to reach an objective won't provide any benefit at all. It may create a waste, a cost, a loss or damage.

	COMPANY		
Executor	INDUSTRIAL COMPONENT	HUMAN COMPONENT	INFORMATION COMPONENT
Way of execution	Automated	Manual	Automated
Storing / Processing / Transporting	Material, Energy	Material & Information	Information
Nature of processes / processing	Pre-defined processes	Interpretation, variable logic, ad hoc decisions, explicit and implicit knowledge	Pre-defined processes



Robotisation = mechanisation + informatisation

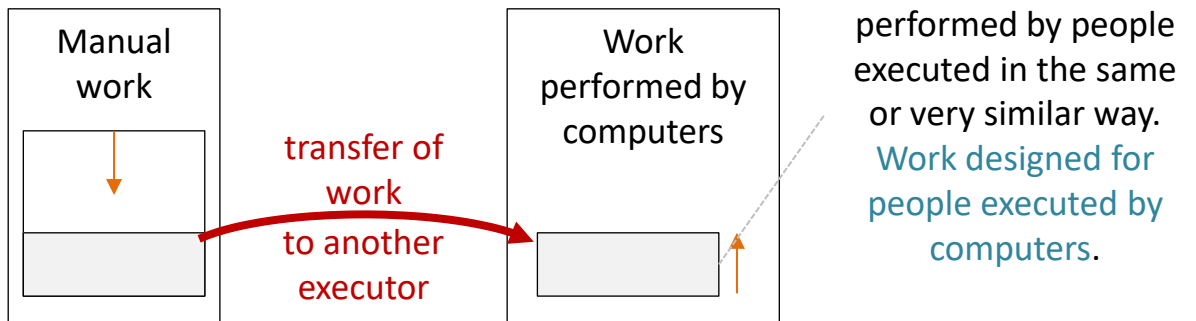


Reflections

Over time, what human activities and business community will be left over after large scale mechanisation and informatisation ?

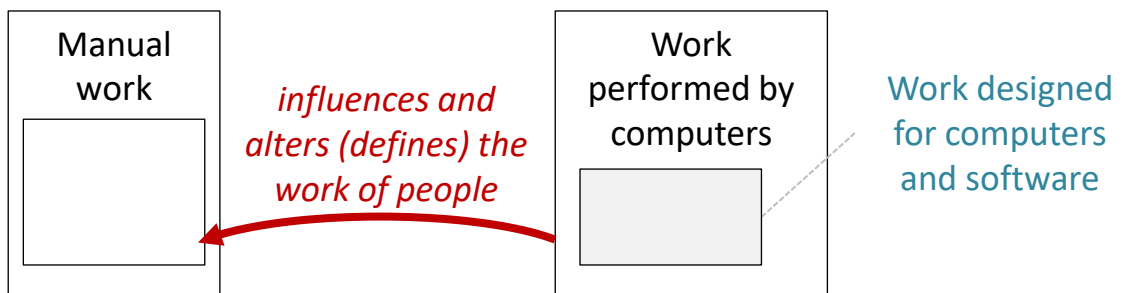
Note: Industrial component is optional. Not all organisations are industrial.

Automating the “As Is”



“Copy-paste” of work (or work is slightly adapted) from people environment to computer environments. Basically, a same way of thinking is applied. Mainly repetitive work is transferred. It values the work of the people. With this transfer people can do more and other work. Productivity increases.

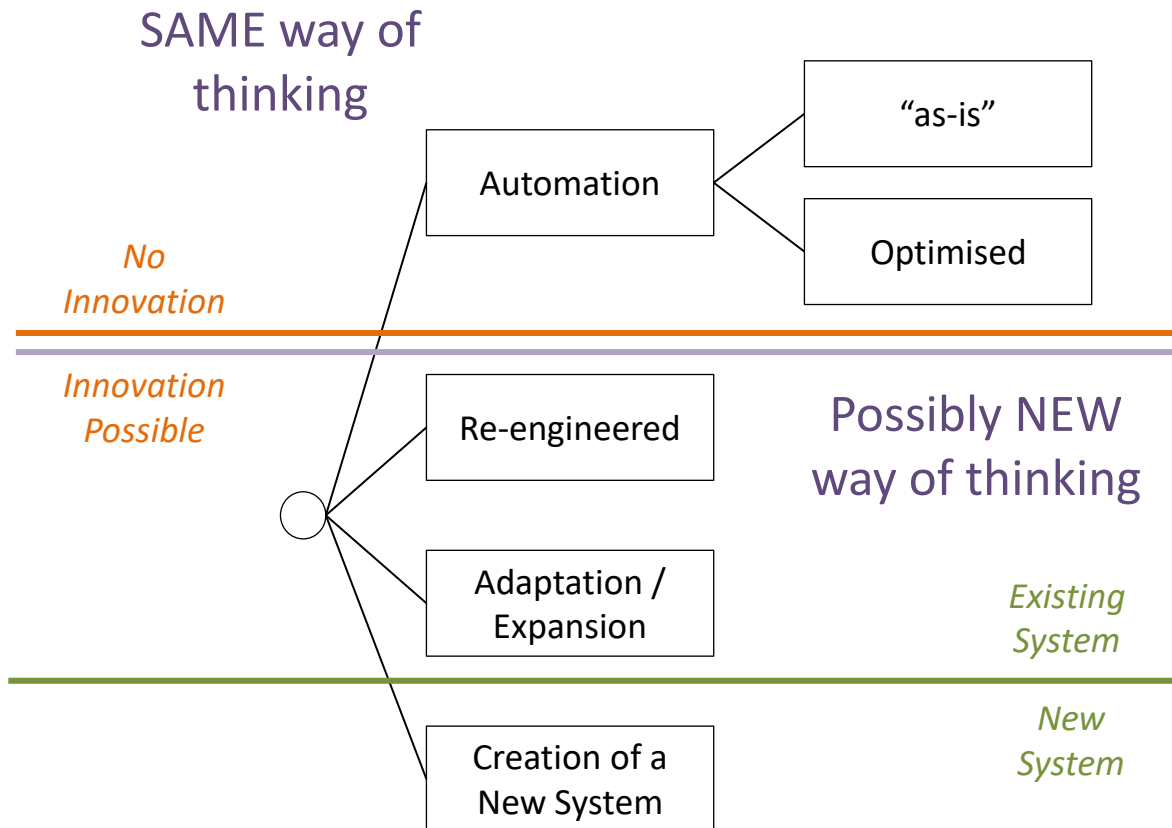
Exploitation of the ability of computers and software



Maximally exploiting computers and software is achieved by designing systems, not based on human abilities, but based on the specific abilities of computers and software. Work of computers is valued. People do work computers can't (yet) do. Requires a way of thinking different than thinking in terms of people and work performed by people.

Notes:

-



Remark: Not all new systems are innovative.

1. Innovative software features and interface
 2. Innovative and more powerful algorithms
 3. Innovative concepts
 4. Innovative system architecture
 5. Innovative processes
 6. Innovative information usages
 7. Innovative information products & services
 8. Innovation supporting new forms of organisation
 9. Innovation in learning and capability development
 10. Innovation contributing to business models
 11. Innovation allowing new or better strategies
-

Reflection:

Why is innovation in 1 and 2 easier?

How often does the IT department come up with innovation on points 3 or higher? If the answer is “rarely” or “never”, then why doesn’t it happen frequently ?

Should this happen frequently? How can this be changed? What conditions are missing?

Innovation doesn't start with a vision or good idea. It starts with an attitude, curiosity and learning, an open mind and an appropriate environment.

Innovation needs ...

- Goal, direction, a problem
- Context
- Constraints
- The right mind-set (curious, investigative, what-if, ...)

Innovative Environment

- Structure, guidelines (but not strict structures, unbreakable laws, ...)
- No pressure or stress, peaceful environment
- No fear
- Failure is allowed. Failure is learning.
- Some freedom
- Some autonomy
- Time, Resources
- No strict plan, deadline (pressure)
- Innovation often happens at the crossroads of two or more domains (or concepts) (overlap of domains)

Revolutionary Idea – Three Phases

1. Ridicule
2. Dangerous
3. Obvious

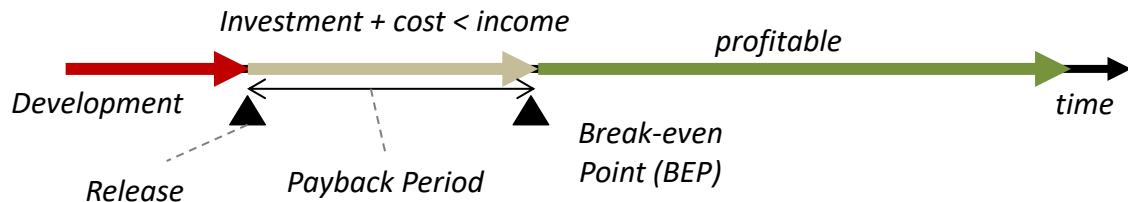
- Authoritarian hierarchical structure (**STRICT** rules and structures)
- Lack of support
- Fearing risks and failure
 - Control (or “transparency”) linked to judgment, evaluation and potential punishments, fear of job loss (restructuring, rightsizing, outsourcing, ...)
- Pressure
- Internal politics, personal agenda’s, ego
- Obligation, procedures to follow, rules
- Chaos, firefighting
- Frequent interruptions, one-hour time sliced day
- Divide and conquer as strategy to manage the company
- Organisational silo’s
- Job silo’s
 - the job description, the function, the authority and responsibility define what activities one may execute, what the individual may think about or to what activities (s)he may participate.
- Knowledge silo’s, job specialisation
- Superficial knowledge, lack of insight
- Short-term thinking, cost-thinking
- Low norms
- Attitude of maintaining and protecting; a defensive attitude
- Dependent thinking, group thinking, single thought, mainstream thinking
- Rejecting and banning disagreements, different thoughts, ...

"If I had asked people what they wanted, they would have said faster horses."

- Henry Ford

A Change is an Investment

First the effort, cost, issues, ... the benefits follows.



Downsides

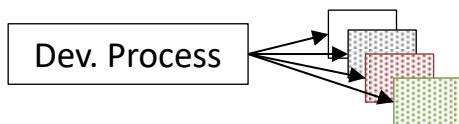
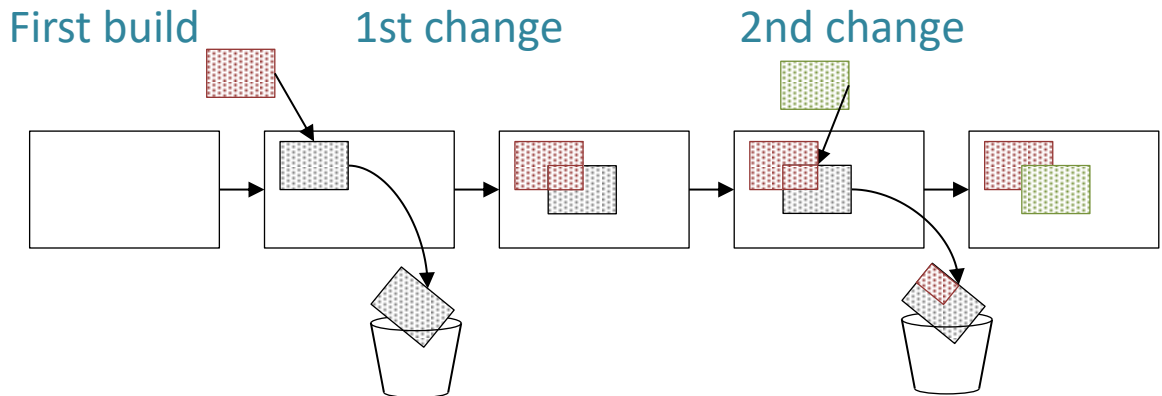
- Effort
- While experts implement one change, they don't work on other developments/changes
- Risks of introducing incoherencies, conflicts, inefficiencies, ...
- Additional stress
- Requires (re-)training
- Risk of confusions between old and new
- Cost
- ...

Benefits

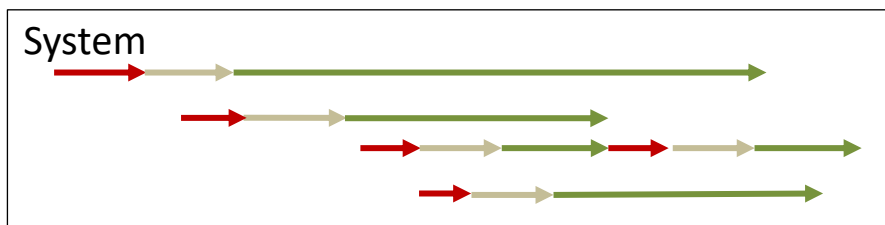
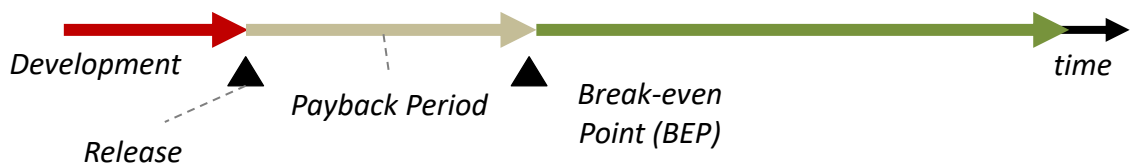
- Learning, Improved skills, new experience
- Improved system, situation, ...
- Create additional value
- Retain or bring customers
- ...

Some negative implications (such as bugs, new limitations, bad quality, ...) may appear much later, when the change is implemented and operational, even after being operational for a longer period of time.

How successive changes destroy the profitability



Every piece of source code has passed through a development process

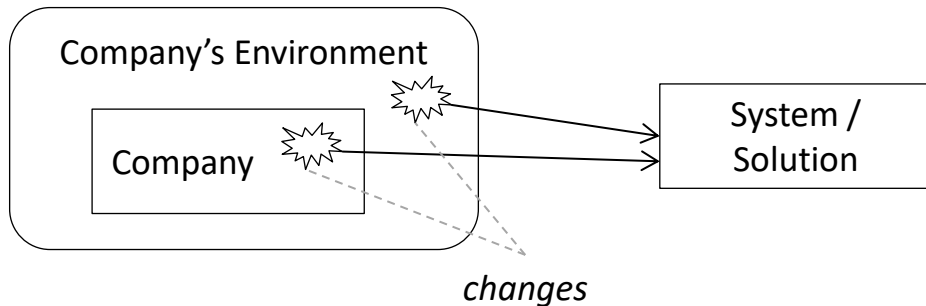


$$\Sigma \text{Cost(Change)} < \Sigma \text{Benefit(Change, time period)} \quad (\text{simplified !!})$$

Some Principle

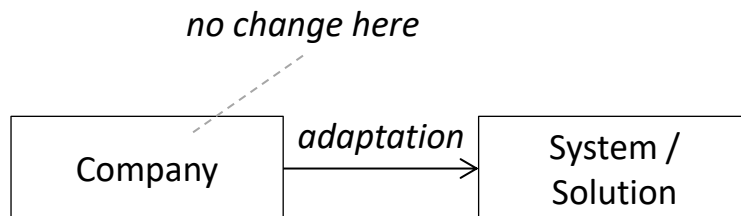
- Every change is work, an effort.
- Every change implies waste (old work thrown away).
- Every change uses resources (analysts, engineers, developers) which can 't be allocated to other work
- Every change is a cost.
- Every change has a payback time.
- Every change influences the profitability of the entire system, first negatively (increased cost), later maybe positively (increased profit). A change is not necessarily beneficial simply because it is a change.

Real Changes



SOMETHING changed in the “Company’s Environment” or in the Business part of the company.

“Fake Changes” – Corrections – ANALYSIS BUGS



NOTHING changed in the “Company’s Environment” or in the Business part of the company, yet the system has to be adapted. They result from **bad decisions and choices**. Some main **causes** are a lack of insight, thinking or skills.

SYSTEMS ANALYSIS PREVENTS FAKE CHANGES

Some fake changes are acceptable (adaptation of strategy, new idea, investigation led to new insight). Even a decent Analysis can not prevent all fake changes. It limits them, particularly the most impacting fake changes.

TIP:

If too many changes are due to lack of information, lack of insight, lack of thinking, lack of skills or other similar causes, then solve the causes.

Changes

External Change	<ul style="list-style-type: none"> • Change is not under control of the company. • Sometimes hard to predict • Can be imposed on the company <p>Examples: Legal changes, market decline, technological progress, partnerships, ...</p>
Internal & Controlled Change	<ul style="list-style-type: none"> • Management decision • Triggered internally • Under control, Rather predictable <p>Examples: Strategy, plans, goals, intentions, policies, reorganisations, R&D, ...</p> <p>Job of the Analyst. Analysts must be informed of discussions or coming decisions.</p>
False Change	<ul style="list-style-type: none"> • Causes: Lack of insight, changing minds • Not caused by a real change • Root cause<u>S</u>? (fill in) • Solutions? (fill in) • Unpredictable, Most are avoidable

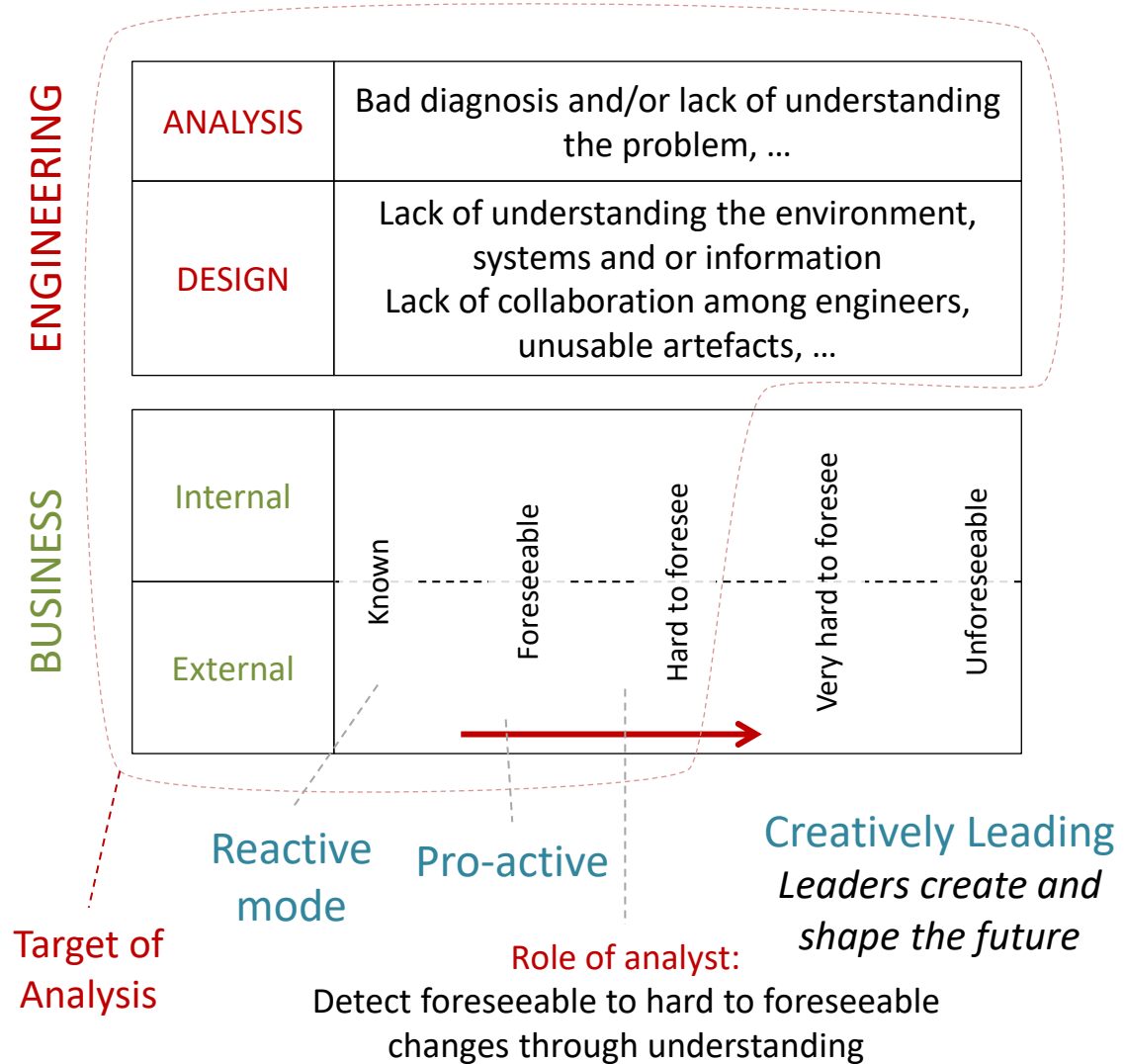


Changes in Systems

Analysis Bugs	<ul style="list-style-type: none"> • The worse the analysis, the more analysis bugs. <ul style="list-style-type: none"> • Causes: Lack of skills, information retention, pressure, lack of collaboration, .. other organisational factors or factors in the work environment. • Often hard to avoid all of them. • Can be similar to false changes.
Design Bugs	<ul style="list-style-type: none"> • Similar to analysis bugs, but now instead of related to the understanding, the bugs concern the design.
Programming Bug	<ul style="list-style-type: none"> • Causes: Lack of insight, changing minds



Changes in Systems

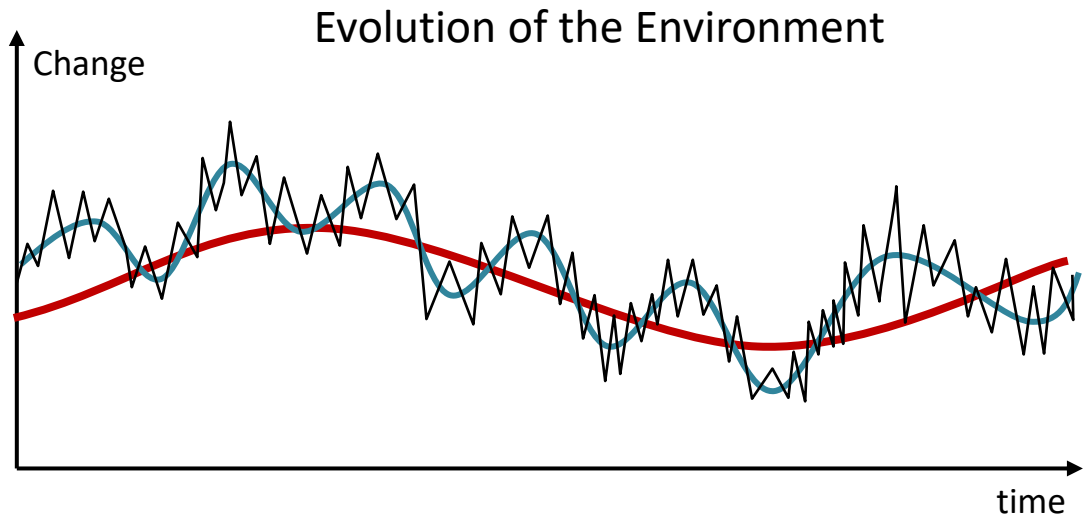


To be able to foresee:

- Understand systems (general systems theory)
- Understand evolution of company's systems
- Understand evolution of business
- Consult plans
- Look at objectives, usage of systems, complaints, intentions

Some internal changes are easier to foresee or are under control of the company itself.

If the management functions in an ad hoc and fire fighting mode, then they are taking a lot of decisions that truly couldn't be foreseen.



Superficial Environmental Changes

- Times scale: a day, a week, a few months, a year
- appear and disappear quickly
- lack of direction
- easily noticeable (newspapers, magazines, advertisements, ...)
- superficial, minor, small opportunities
- responds doesn't take a great effort
- pick the battles

Medium Environmental Changes

- Times scale: several months to several years
- Harder to notice (visible tendencies in society)
- Some tendency and direction can be observed
- Can be worth setting a project or programme for it
- Harder to escape from it

Profound Environmental Changes

- Time scale: many years to several decades or more
- Hard to notice, requires to consider the long term, knowledge of the past (history)
- Clear direction
- Obligatory to follow it

Notes:

What type of changes should shape the architectures? What should not be used to shape the architecture?

Resistance of change is bad because change is inevitable. The resistance must be broken. Those who resist fear the future. They don't understand or they can't adapt. They are nostalgic and can't let go the past. We live in modern times and need to progress. Such unfortunate clichés are well-known.

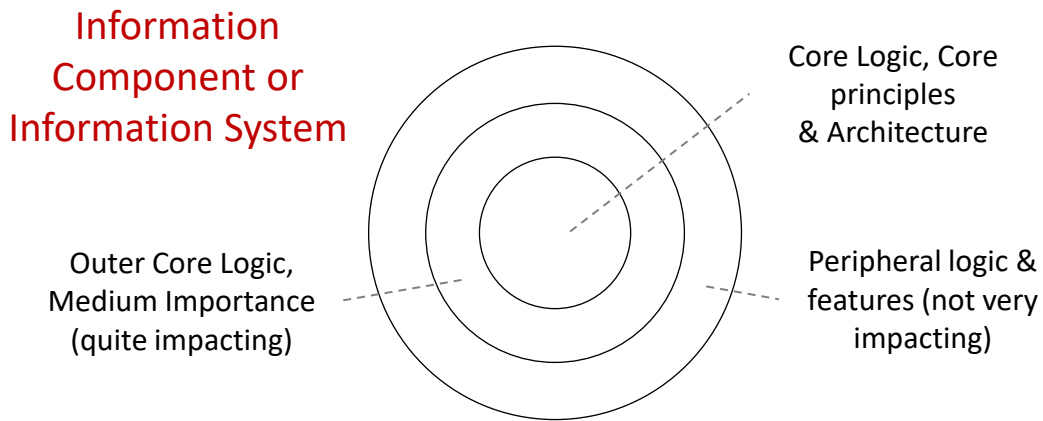
Resistance is natural and is inherent to change. It is healthy to have resistance. Resistance is there for a reason. It is a signal that must be captured. Its message has to be understood. First, resistance prevents us from accepting and responding to every opportunity of change. It tells us something about what and how we do things. If we would do things correctly, then why would there be a resistance?

Resistance exists

- Because of a **loss** for a concerned party
Is the loss justified? Can this loss be compensated?
- Because of **fear**
Fear is or should be abnormal. What is that fear? What created it? How do we deal with people? Can we reassure people? Or take measures to resolve the fear?
- Because of **uncertainty**
Change means uncertainty. And uncertainty is close to fear. But it is important to listen to what these uncertainties are. Maybe we created them and maybe they can be resolved by listening, by providing more and clearer information and by taking specific measures.
- Because there is **disagreement** with the decision
A disagreement can be justified. If there is a disagreement, then maybe something has been overlooked.
- Because a **lack of confidence** and belief
This should also be investigated and measures can be taken.
- and indeed because of change requires an effort and not all persons are ready to do the effort.

In some cases, resistance is created by lack of communication, by a lack of information: what are the intentions? What do we know and what don't we know yet? How much do we care about the people? How will we deal with people during the transition?

Key concepts: right intentions - information and communication – empathy and caring – respect – support



The design of what layer of logic should be based on what evolutionary wave? (superficial, medium or profound)

Type of Environmental Change / Evolution / Opportunity

	Profound, Long Term	Medium	Small, Superficial
Peripheral logic and features	👍👍	👍	👍?
Outer-core Layer of Logic	👍👍	👍	👎
Core Logic & Architecture	👍👍	👍?	💀

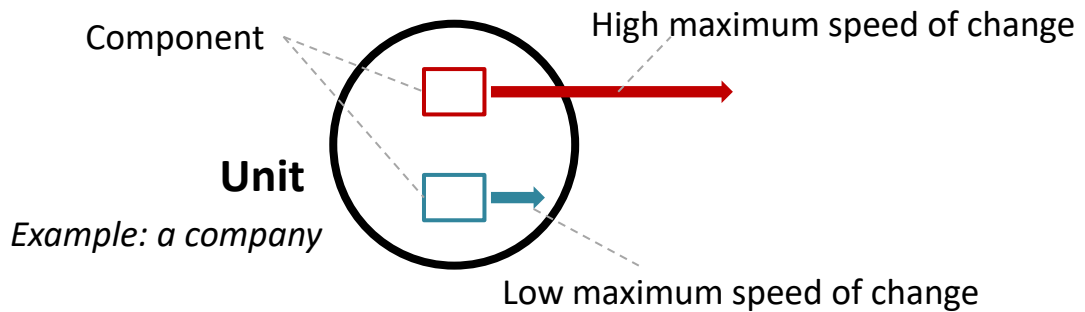
The design of the core, shouldn't be based on superficial changes. It should be based on the more stable deeper, long term evolution.

It is important to understand if a piece of logic is fundamental (long term) or if it is likely to change every few months, so to speak.

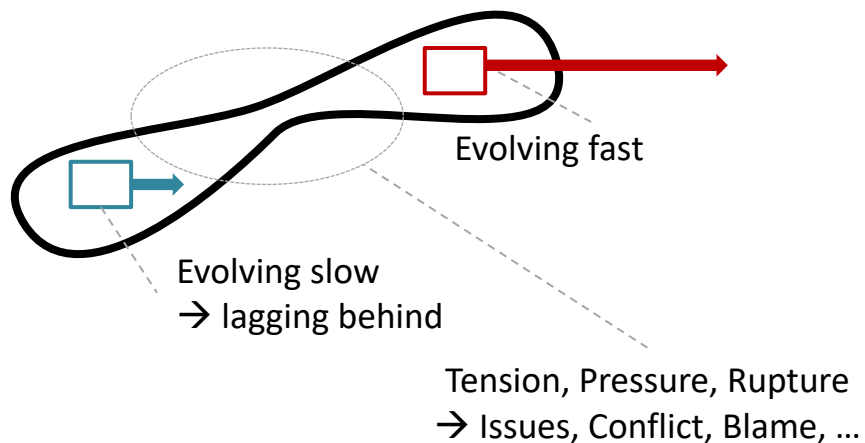
Size of change, its criticality and its profitability have also to be considered in the light of the type of environmental change (evolution).

Change : Speed

10/01/2020



Let all the system's components change at their maximum speed

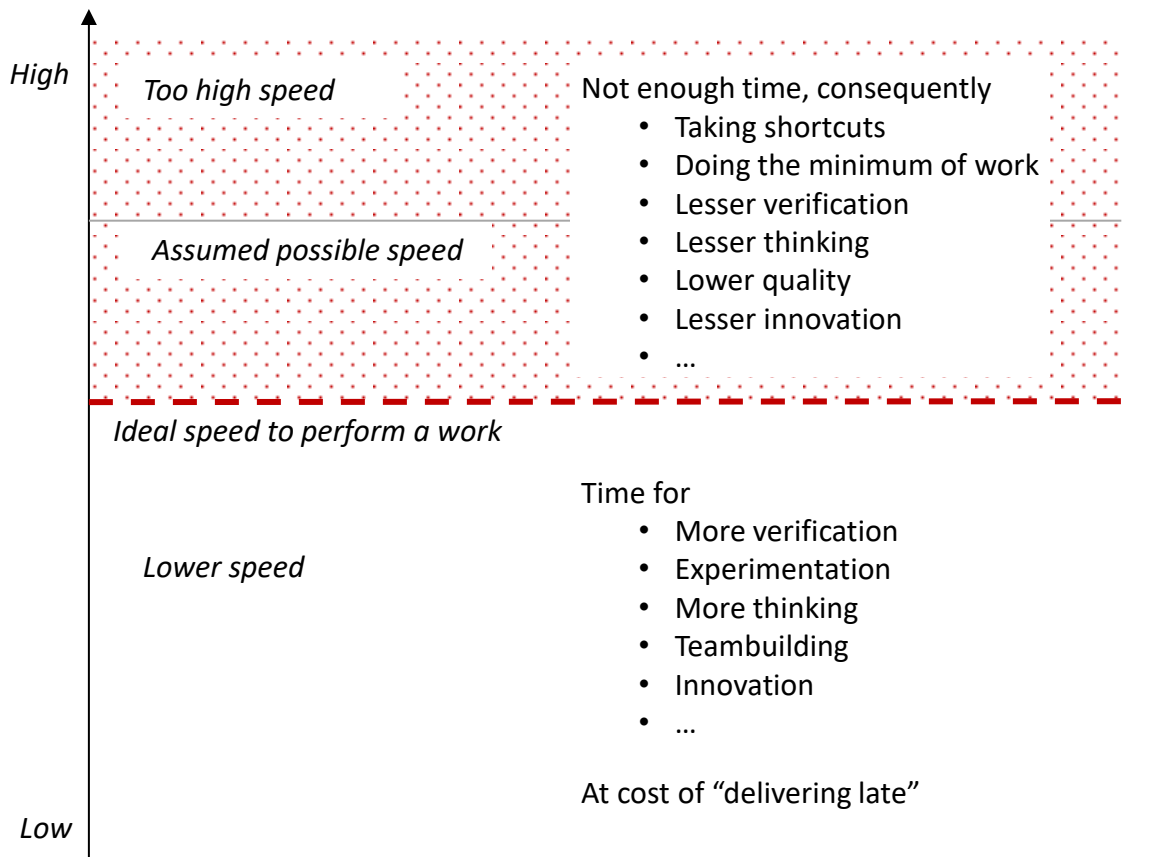


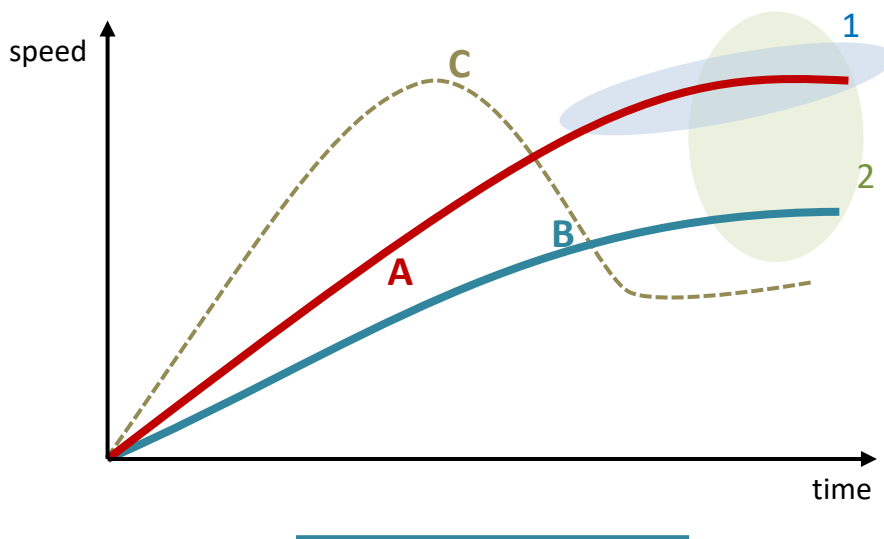
The element slowest to change should define the pace of global change.

Make the slowest components faster
However ...

Change : Speed

10/01/2020





1) Evolution of speed

Line A: Increase of speed is not limitless. Likely companies may meet a plateau, a maximal speed (**blue area 1**). The increase of speed of change creates pressure.

Line C: Fast-Cheap-Local-Tangible (short-cuts, quick wins, technical debts, reactive initiatives under pressure). Wanting to go even faster may lead to an overshoot. Pressure and speed may lead lack of quality and various kinds of problems. It's a kind of "speeding ticket". Understanding and awareness of the consequences of speed may lead to a downwards regulation of the speed and pressure and an increment of cost and risks in the medium or long term. We may not be aware of these cost and risk increment or fail to link them to the speed and pressure. They are considered as normal, while they are and should be avoided.

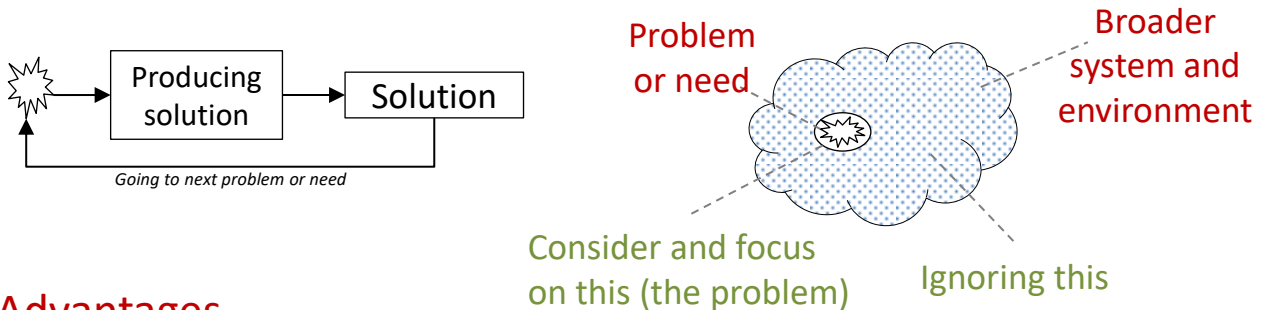
2) Different pace of change

A change in a company often requires many components and aspects to be changed. Not all components in a company can change at the same speed (**Line A** versus **Line B**). Often, this difference in pace of change puts a pressure on the components with a slower pace of change (**green area 2**). They slow down the whole change. This may also put too excessive pace of change and pressure in some areas which lead to lack of quality and other kinds of problems.

Bottom-Up - Approach

14/02/2020

As an opportunity presents itself, appropriate decisions are taken to progress. Principle: perform a step, contemplate the situation, decide of next step, perform the next step.



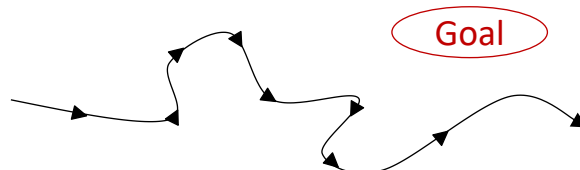
Advantages

- Resolving existing needs (or needs that are about to materialise)
- Concrete cases or demand
- Tackling only someone's known problems. These problem are the most annoying and limiting ones at the present moment. No need to search for unknown problems or demands.
- Better fit with reality. Possibility to provide an answer that suits the local situation
- Effective on local level and in the short term. Some end-users can work better sooner.
- Possibility to have quick-wins

Disadvantages

- Directionless, unknown destination, has no direction, no predefined goal or risk for losing higher goals out of sight
- Reactive, opportunistic, follows
- Risk: Constant pressure (when problem or need already exists)
- Lack of consideration for the effects and implications on the future
- Clueless about what will be the end result over longer period of time
- Risk: Not recognising or considering the overall system, the supra-system the overall environment
- Not scalable
- Risk: One change may create issues else where due to limited scope
- Risk of oversimplification because of reduced scope and thus bad solutions
- Greater risk for symptomatic problem solving
- Ineffective at global level: risk of sub-optimisation, incoherencies, following wrong priorities, unnecessary work, no value creation, ...
- Ineffective in the long term: over time is likely to lead to fragmentation and chaos
- Unpredictable

*Continuously seeking a solution,
aimlessly applying corrections*



Bottom-Up Approach / Demand-driven



*Electricity &
telephone cable
network*

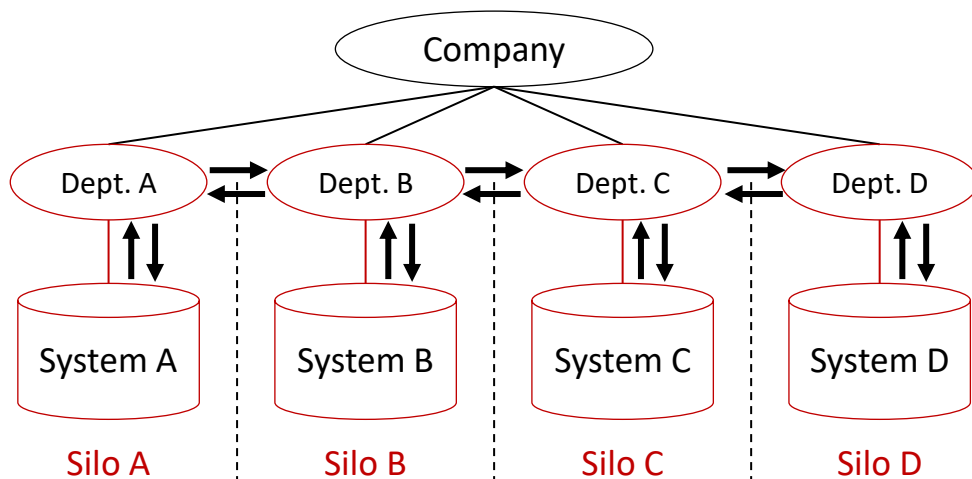
Principle: A wire (feature, process, ...) is added when needed, when demanded. And this is repeated, and repeated, and repeated, ... endlessly.

- We are only interested in the results ... the concrete, tangible & visible results.
- What works is good enough. Really?
- This is what source code looks like (the picture represents metaphor; the source code is actually significantly more complex). The source code is not visible to many. So it is ignored.
- Efficiency? Adaptability? Evolvability? Finding bugs? Fragility of the system? These are also results, time (delays), cost & risks
- Yes, but ... no technical debt is allowed and we do refactoring, ...
- Really?
- Demand-driven, by definition, creates lag. Lagging means pressure. No time for rethinking, re-engineering, refactoring or even cleaning up source code.
- Imagine this after 100, 1000, 1000000 repetition of new demands and changes ... without clean-up

Same is true in many domains, such as rules, policies, processes, data, etc.. It's all hidden under the surface.
We don't see it or/and we don't want to see it.

Lesson from Silo Development

14/02/2020



Historical:

Each department, service or business function was automated. It received its own system. Scope was limited to the organisational unit or business function. Systems were built sequentially over several years. They were built independently from each other. Sometimes different technologies were used.

Main Problems:

1. Communication between systems was troublesome or impossible
2. Incoherencies in logic within the different systems
3. Lack of standardisation, fragmentation
4. Logic is Double logic – No reuse
5. Information sharing is difficult
6. Overall solution is very inefficient

Causes / Lessons to be Learned:

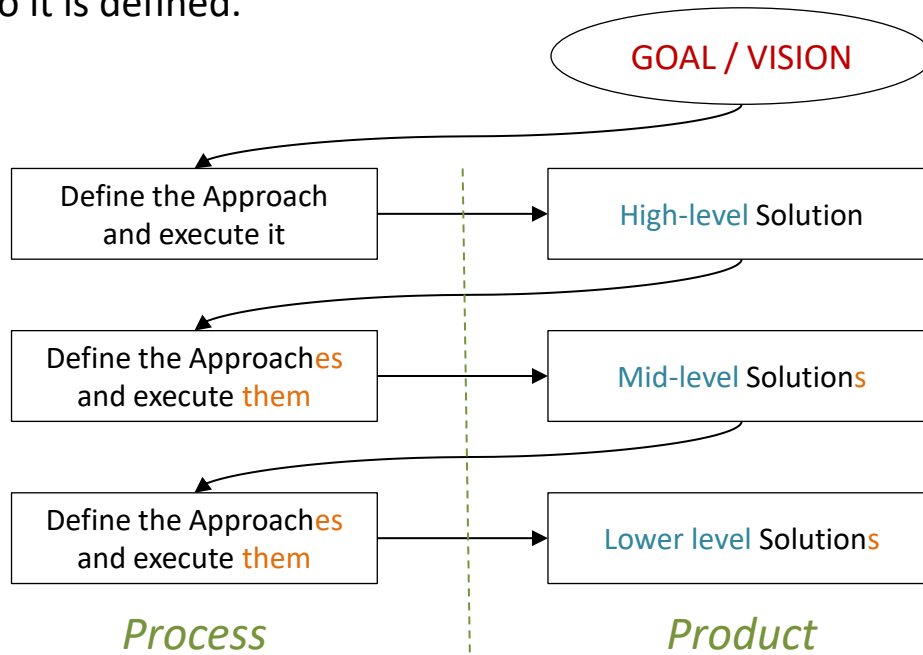
1. The high-level overview with which is started is still too low (in the top-bottom approach, the top must be higher)
2. Problem domain and scope are too limited (limited to the problem). It must be wider.
3. Cross-boundary view is missing

Note :

EAI solved the consequences and made everything more complex. It didn't solve the causes.

Top-Down – Approach

Companies and organisations are driven by a vision, an intended future situation and by goals. They create the future they want. The path to it is defined.



- An approach is defined to conceive a global solution that will meet the goal.
By executing it, it will produce a high-level solution.
This solution can be broken down and approaches can be defined to conceive the mid-level solutions or to detail the global solution more.
This refining process is repeated.
- We may have a layered problem or a layered solution.
- The solution may be expressed as an architecture and many kinds of different models.

Advantages

- Often driven by higher vision, mission, goal, purpose, strategy
- Allows to build a certain future
- Is necessary to undertake larger and more complex initiatives
- Fits most company cultures and core of the companies or organisations
- Allows to conceive solutions at all levels
- Greater visibility over time
- Direction and greater clarity
- Easier to design an architecture and thus to have a clearer view of the product organisation, qualities, functional and logical coherence, security, etc.
- Easier to keep a structure and organisation clean
- Scalability in product
- Scalability in process
- Easier to identify 'all' the business stakeholders
- Dependencies can be more easily identified
- Easier to guide, coordinate and manage
- Allows to be much more multi-disciplinary because the scope is larger.
- Alignment of teams and means (resources)
- Allows to better manage pressure
- Overall optimisation is possible
- Allows to better consider impact in a broader area and effects in the longer term
- Requires advanced skills. Employees are more competent.
- Issues can be tackled at their true level of complexity
- Lesser changes

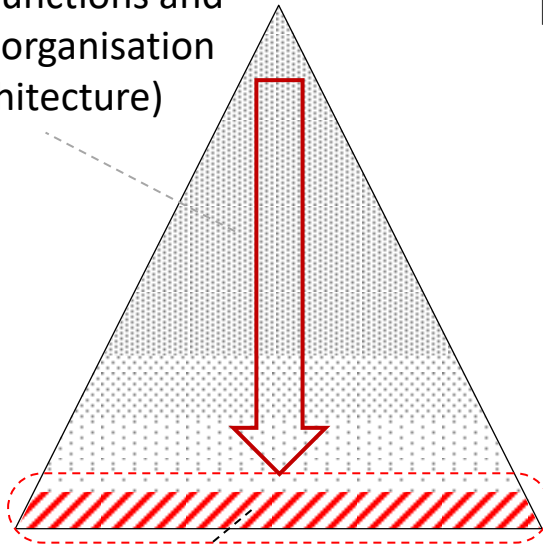
Disadvantages

- Risk of not matching the low level practical reality and smaller needs. But this is easier to solve.
- We may discover skeletons in the closet
- May fit globally, but it may not match the practical issues, difficulties and needs at the operational level. *If the overall system is well-designed, it can then more easily be adjusted to this practical reality.*
- Not suitable to obtain quick-wins

Top-Down / Bottom-Up

TO-DOWN

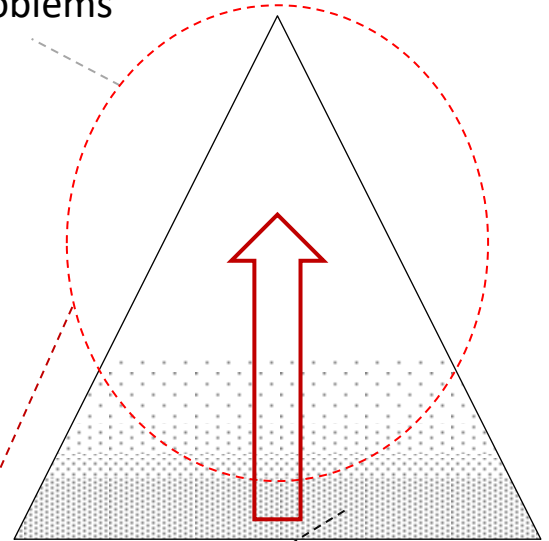
Global functions and overall organisation (architecture)



Risks for practical, local problems and (smaller) unexploited opportunities

BOTTOM-UP

Risks for problems



Chances of lesser problems and opportunities are more likely to be exploited

Risks for problems

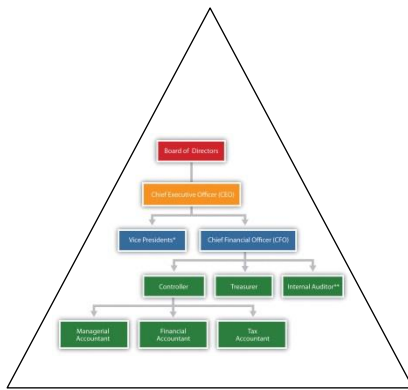
- Working towards and meeting higher goals
- Implementing the intended strategy
- Architectural issues (hairball architecture)
- Fragmentation and overall “fragilisation”
- Long term issues

Question:

Which type of problems (top-down issues or bottom-up issues) are the easiest to be solved ?

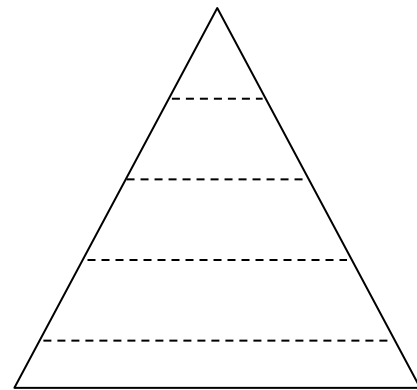
Huge Top-Down - Confusion

14/02/2020



Human Organisation

≠



Logical Levels
(Levels of Detail)

“Top-down” can be approached

from an **ORGANISATIONAL PERSPECTIVE**

(from top management to middle management etc. to the operational level)

OR

from a **LEVEL OF DETAIL**

from global level (main ideas, main lines) down to the details.

But both are NOT the same.

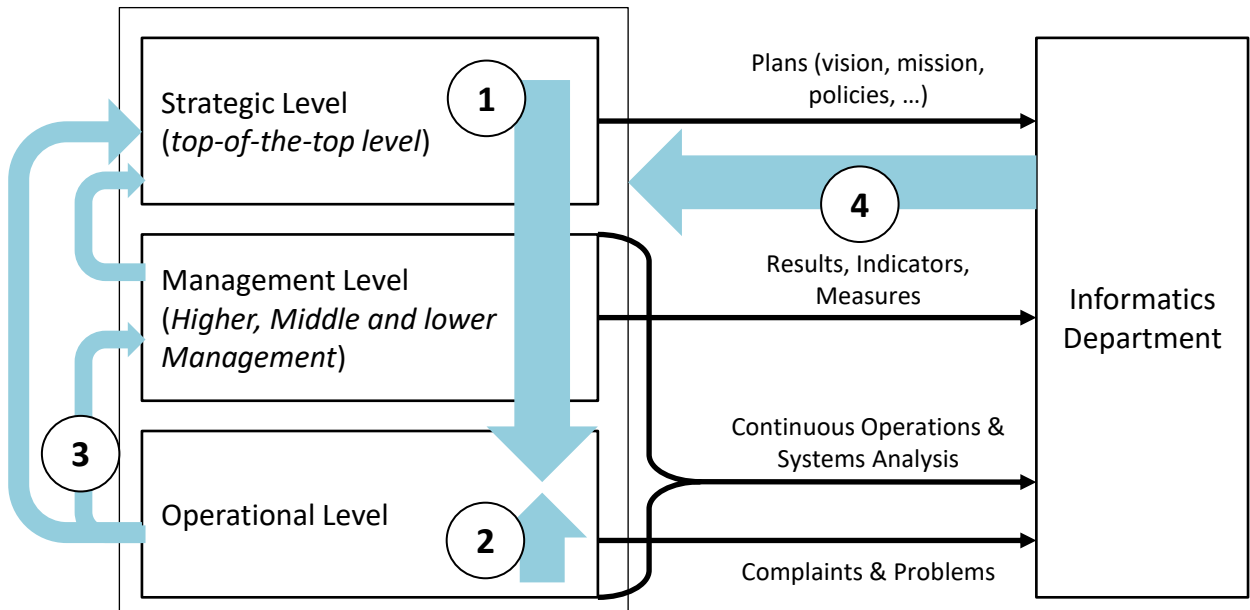
- **Strategy & Management Decisions: Organisational top-down** ✓
Since it is the job of managers to set long term goals, to decompose them and to define the path to achieve those goals, hierarchical levels and logic corresponds for strategies.
- **Solutions & Systems: Logical top-down** ✓

Notes:

Emerging strategies are (at first) spontaneous (unintentional) strategies that are initiated at the operational level.

Induction Flows of Changes

30/10/2018



- 1) Top-Down:** Guidance based on plans. Long Term. Defines the overall structure and main functions. Pro-active. Vision based. Strategy implementation. Business Model implementation.

“THINK BIG, ACT SMALL”

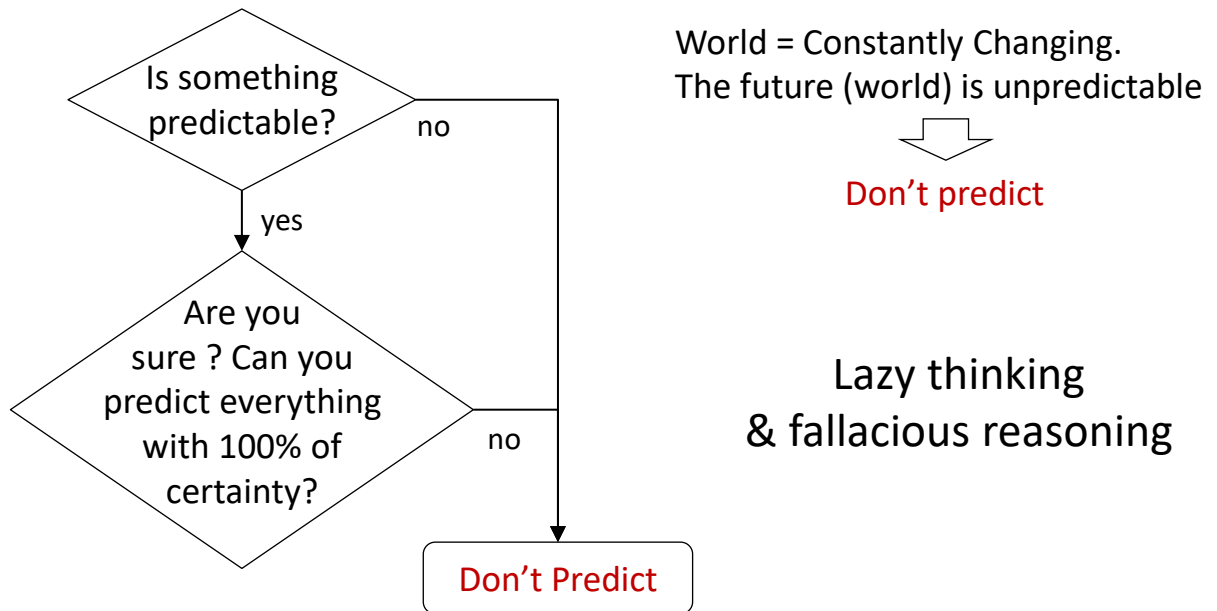
- 2) Bottom-up:** “Think Small – Act Small”: Solving local problems, filling gaps, fine-tuning, removing local obstacles, short term, local needs, reactive
- 3) Bottom-up tackled top-down:** e.g. emerging strategies becoming overall strategy; groups of local issues requiring top-down approach in order to ensure alignment and integration with top-down (avoiding opposite evolutions)
- 4) Engineering-driven and Technology-driven changes:**

Engineering-driven: through continuous Analysis, new concepts, ideas, optimisations, and other improvements and innovations

Notes

Environmental scans may detect changes, obligations and opportunities in the environment. These may trigger one of these flows.

“Think small, act small” = short quick wins; no fundamental changes; creates chaos.



- **Some things are predictable**, others aren't or are lesser pred.
 - Statistical
 - Estimations
 - Cause-consequence relations
- **Degree** of uncertainty (not black or white)
 - "x% chance that ..."
- **Predicting under some assumptions**
 - Expressing assumptions or conditions under which the prediction is made. ("Assuming that ..., then this is likely to happen")
- **Accuracy of the prediction**
 - expressed as a range (margins; between x and y)
 - as a set of possible cases (this, that and/or that)

Making prediction is not guessing or thoughtlessly expressing a possibility

Predicting is a genuine and painstaking effort

Increasing the quality of the prediction

- Having good information
- Understanding the matter
- Understanding the forces at play and the evolution(s)

An alternative to predicting :

BE THE LEADER - CREATE THE FUTURE

It is obvious that without the right information, without understanding and without being able to envision the future or doing the effort of envisioning it (analysis of the future), it is unlikely to make useful predictions.

**“It is not because someone can’t
that it is impossible.”**

Dealing with Unpredictability

18/12/2018

Unpredictability is inherent to software development projects ... but ...

Reducing the Unpredictability

- **Analysis aims to reduce** the unpredictability and limit the issues
 - **Learning and understanding**
 - Business domain knowledge, Business expertise
 - Business environment
 - Right diagnosis done early
- Study of plans, objectives, intentions and expectations
- Verify, don't assume
- Understanding of people
- Understanding the medium and long term evolutions (history, deeper underlying currents)
- Think, imagine, what-if, ...
- Leading, creating / shaping the future, having a vision and goals

Ability to deal with unexpected changes

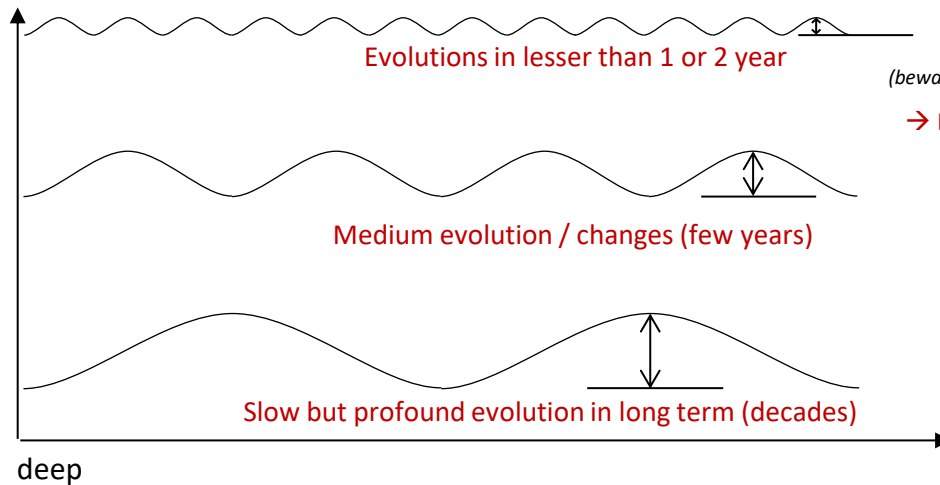
- Vigilance to detect conflicts, issues and opportunities early
- Be flexible and reasonable
- Increase flexibility through
 - A well-designed Architecture
 - Component Based Development
- Adapt plans

Before accepting the 'Change'

- Investigate the cause and nature of the change
- Could the change have been detected through Analysis or through some more thinking?
- Is it worth (cost / benefit) (effort/benefit) ?
- Refused or accepted? Alternative solutions?
- Can different changes be regrouped?
- Do the different changes have something in common?
- Should the change be implemented now or later?
- Can the present solution be prepared for a future implementation? Or is a partial and phased implementation possible?

Environmental Evolutions and Systems 10/01/2020

surface



many – small – quick – little impact –
unpredictable –
short time frame –
short term survival
(beware of effect of high numbers)

→ Deal with it bottom up,
in an adaptive way

few – large – slow –
hugely impacting –
quite predictable –
planned –
guided –

long term survival –
used to shape the future

→ Deal with top-down

A Few Examples



- Advertising, Product promotion
- New products, changes in products or services
- Seasonal changes (predictable)
- Competitor suffering an adverse event
- Small political conflicts, economical glitches
- ...



- New laws, new government and politics
- Changes in distribution network
- Changes in collaborations
- Products and services having a greater impact on society
- ...



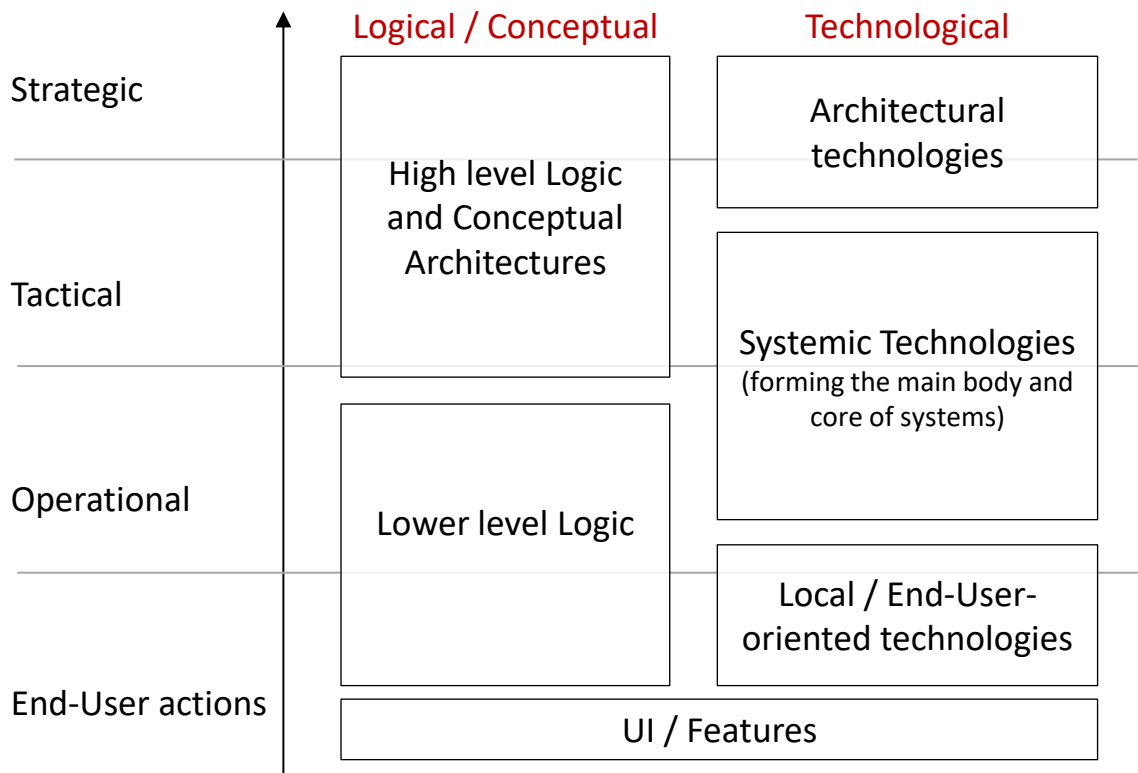
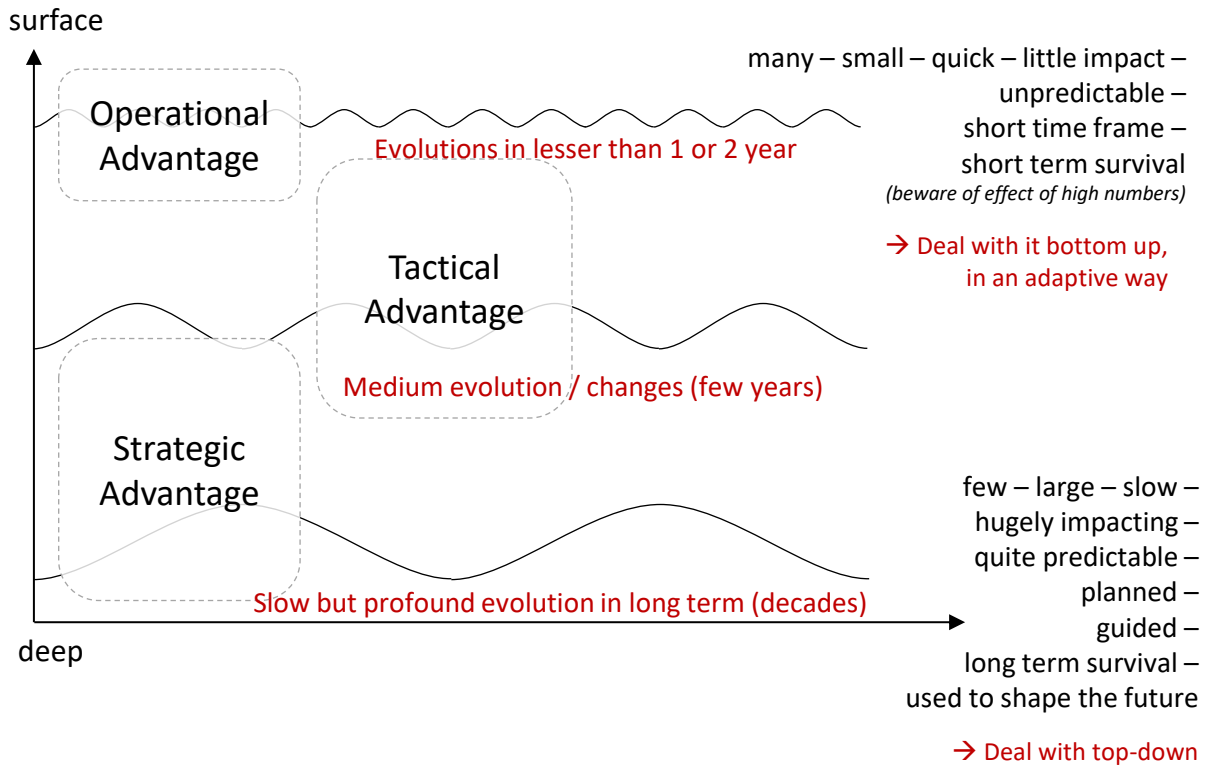
- More diversified, smaller and localised energy production
- Increasing automation
- Artificial Intelligence
- Evolution of education
- Aging population
- Imbalances in society
- Emerging economic powers
- Transition from fossil energy to cleaner forms of energy
- ...

To ponder:

What is the most suitable to base the architecture of a system of systems upon ?
And what is the worst?

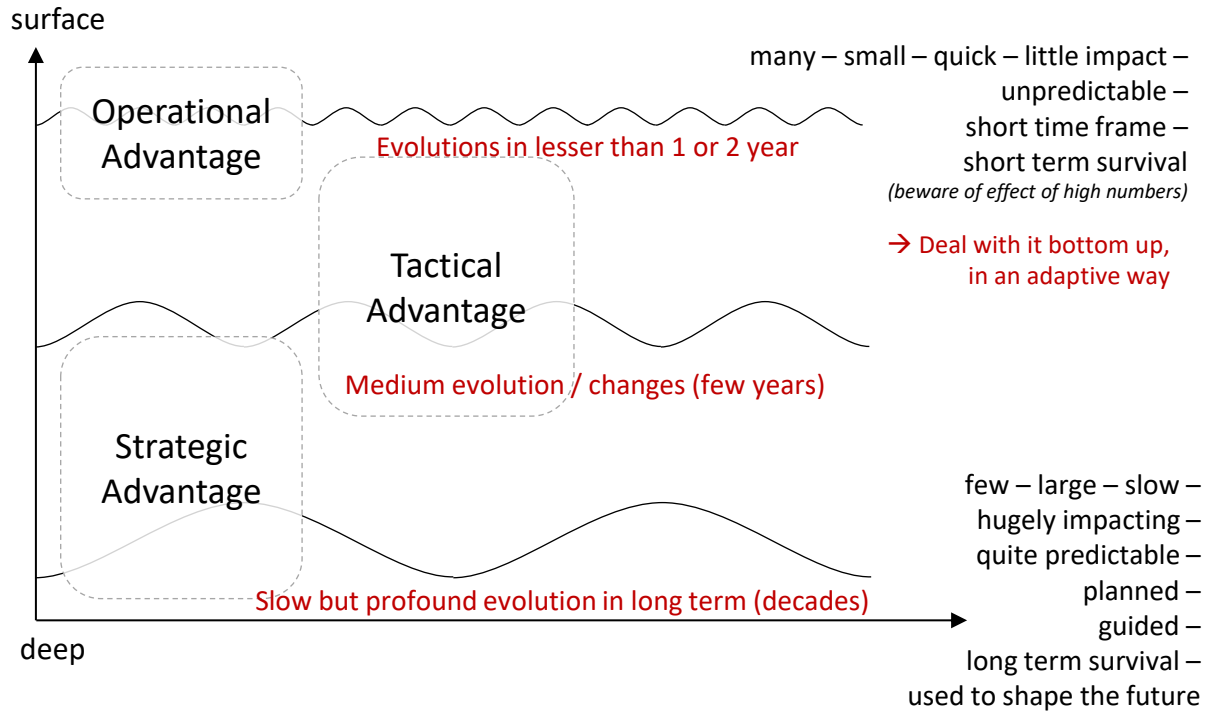
Environmental Evolutions and Systems

10/01/2020

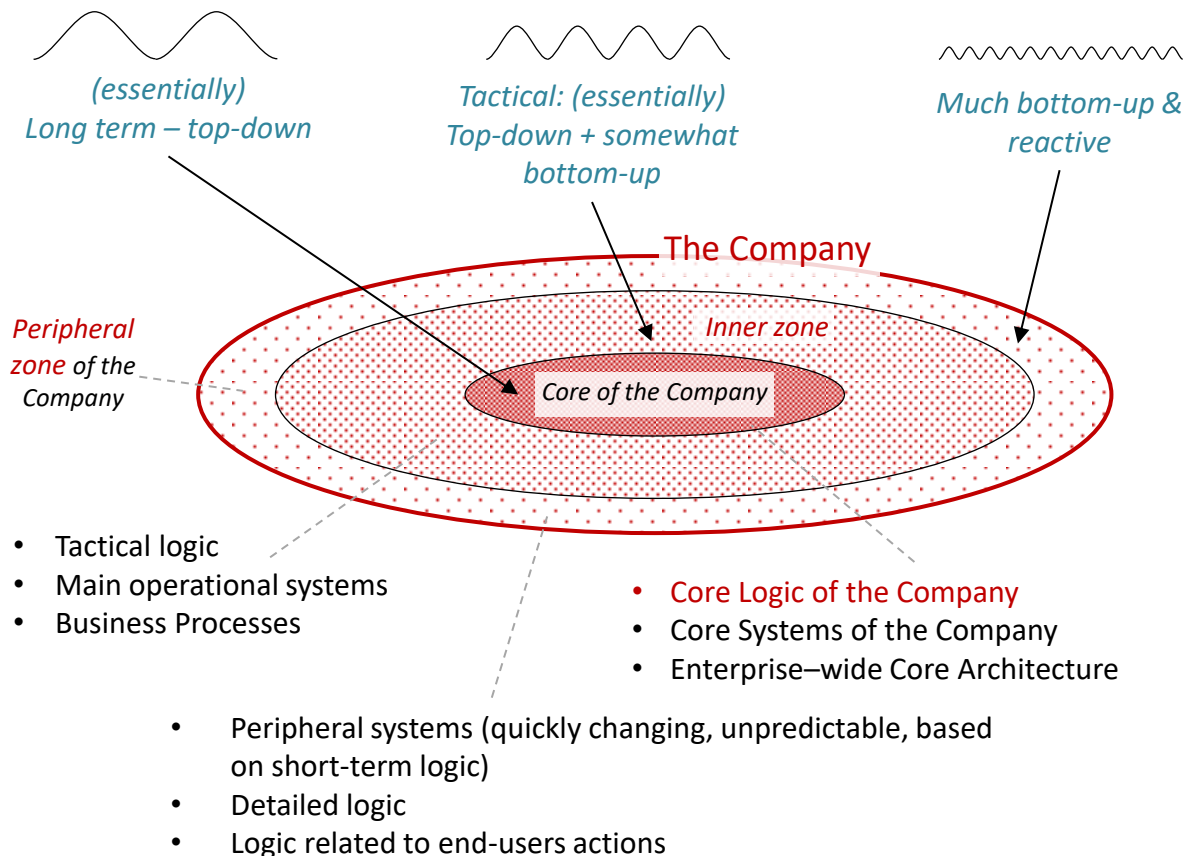


Environmental Evolutions and Systems

10/01/2020



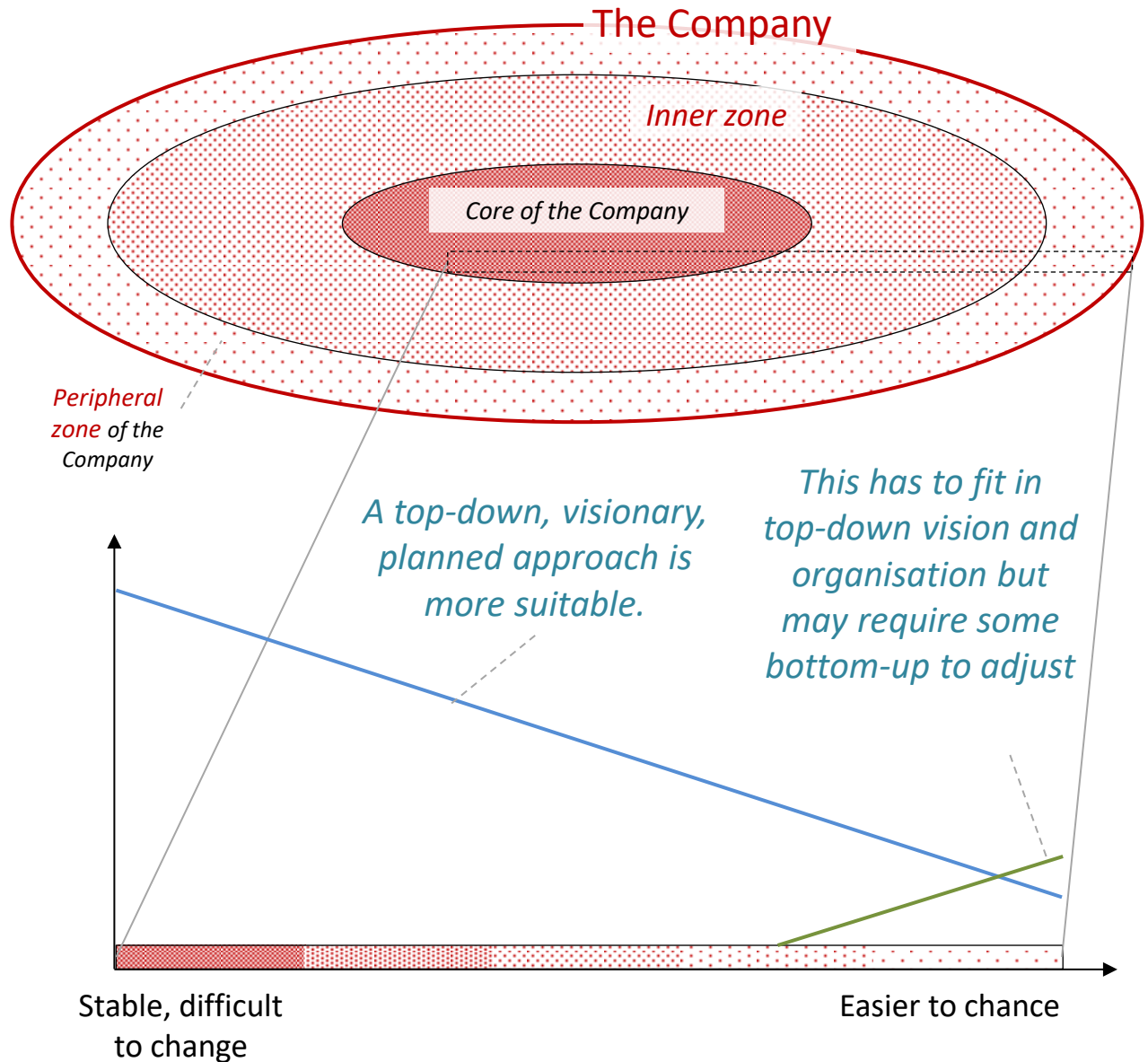
Logical (conceptual, functional), process (activity) and informational perspective on the company



Environmental Evolutions and Systems

10/01/2020

Logical (conceptual, functional), process (activity) and informational perspective on the company



Note:

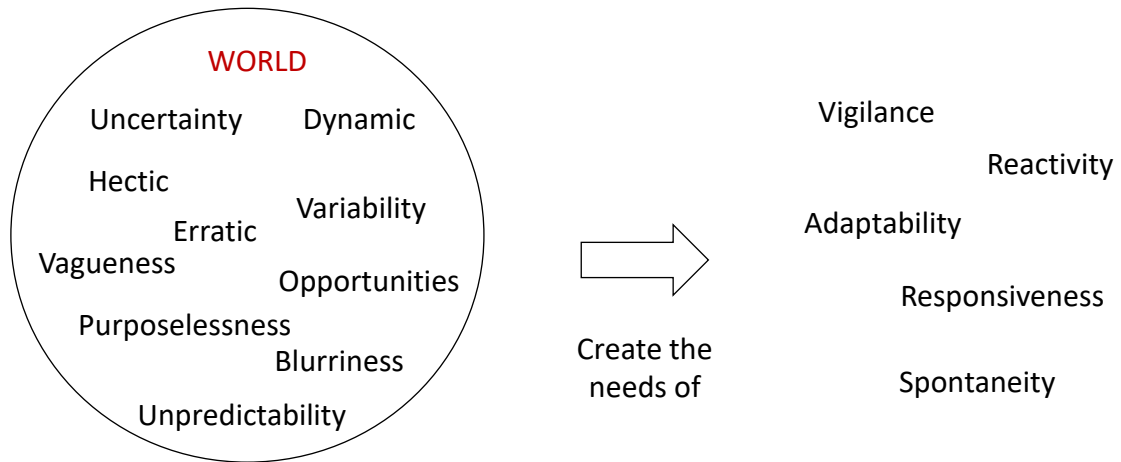
This represents a general idea and has to be understood as “in general, more likely to be more” (or “to be less”). It doesn’t matter whether lines are linear or curved, or whether they cross each other or not.

Every company, situation and even specific issue is different. For each case, it is advised to think about whether the system, the company, the mission, the architecture, the design is more driven by a vision, a strategy, a plan, a global model or by spontaneous opportunities, or what part is most driven by what. This influences the choice for the approach, among others.

Chaos vs Structure (1)

30/10/2018

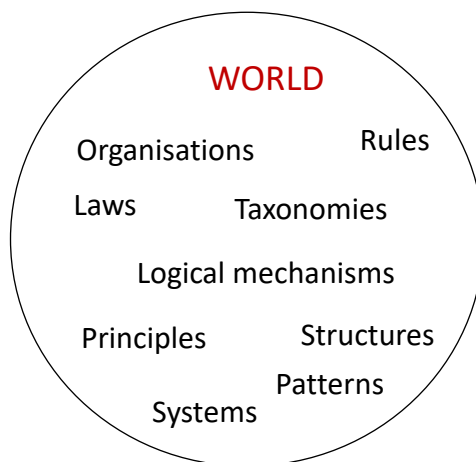
We live in a Chaotic World



The world is full of chaos. **However** ...

There are countries, cities, road infrastructure, energy networks, companies, traffic rules, law systems, laws of physics, laws of biology and chemistry, thinking patterns, languages, planetary systems, ecological systems, economical system, software systems, and so on...

We live in a Structured World



Allows us to

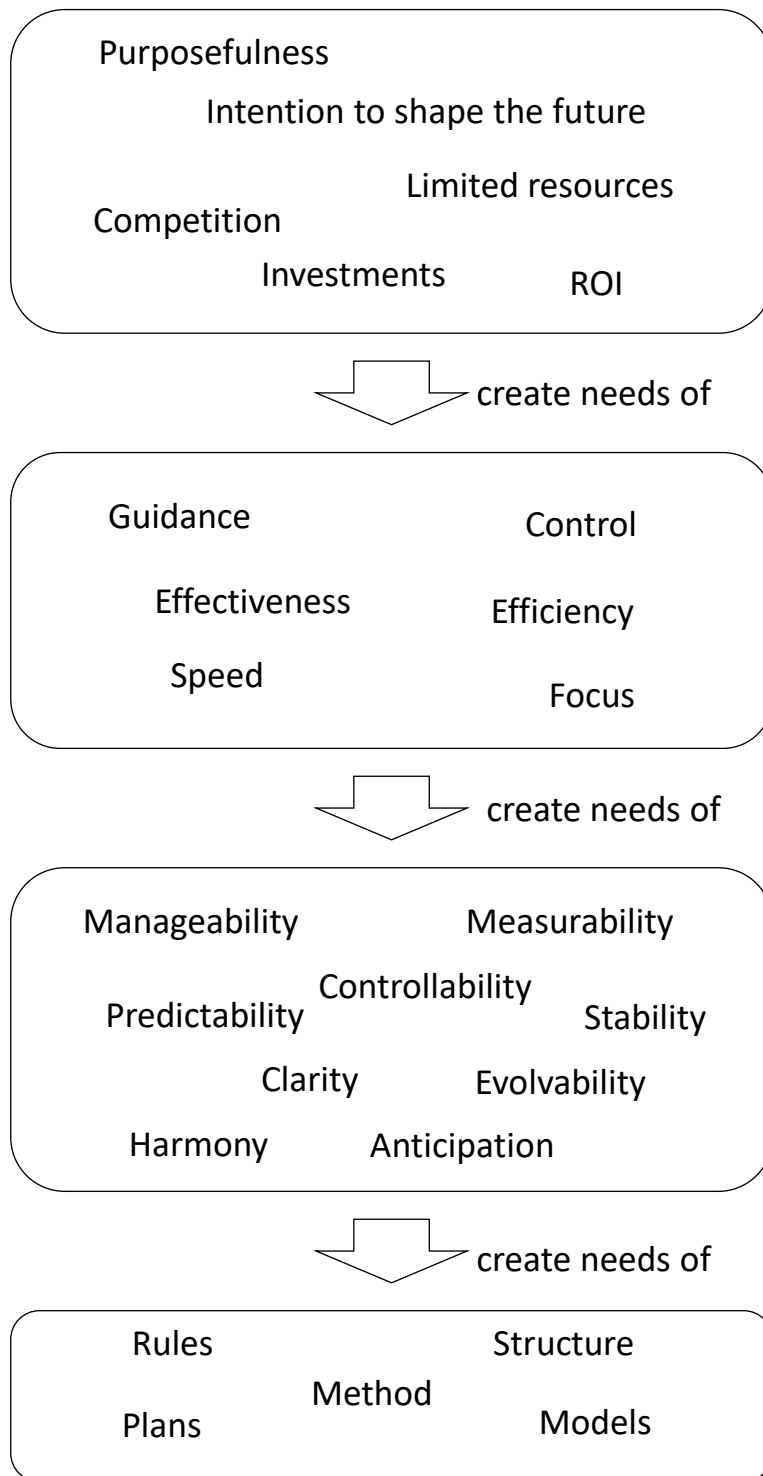
- Understand the world
- Deal with the world
- To anticipate
- Shape the world
- To function more easily
- To collaborate
- To communicate
- To evolve
- To invent and innovate
- ...

Notes:

Note that the product of innovation is always a system, method or technology (something structured), something to get a predictable result.

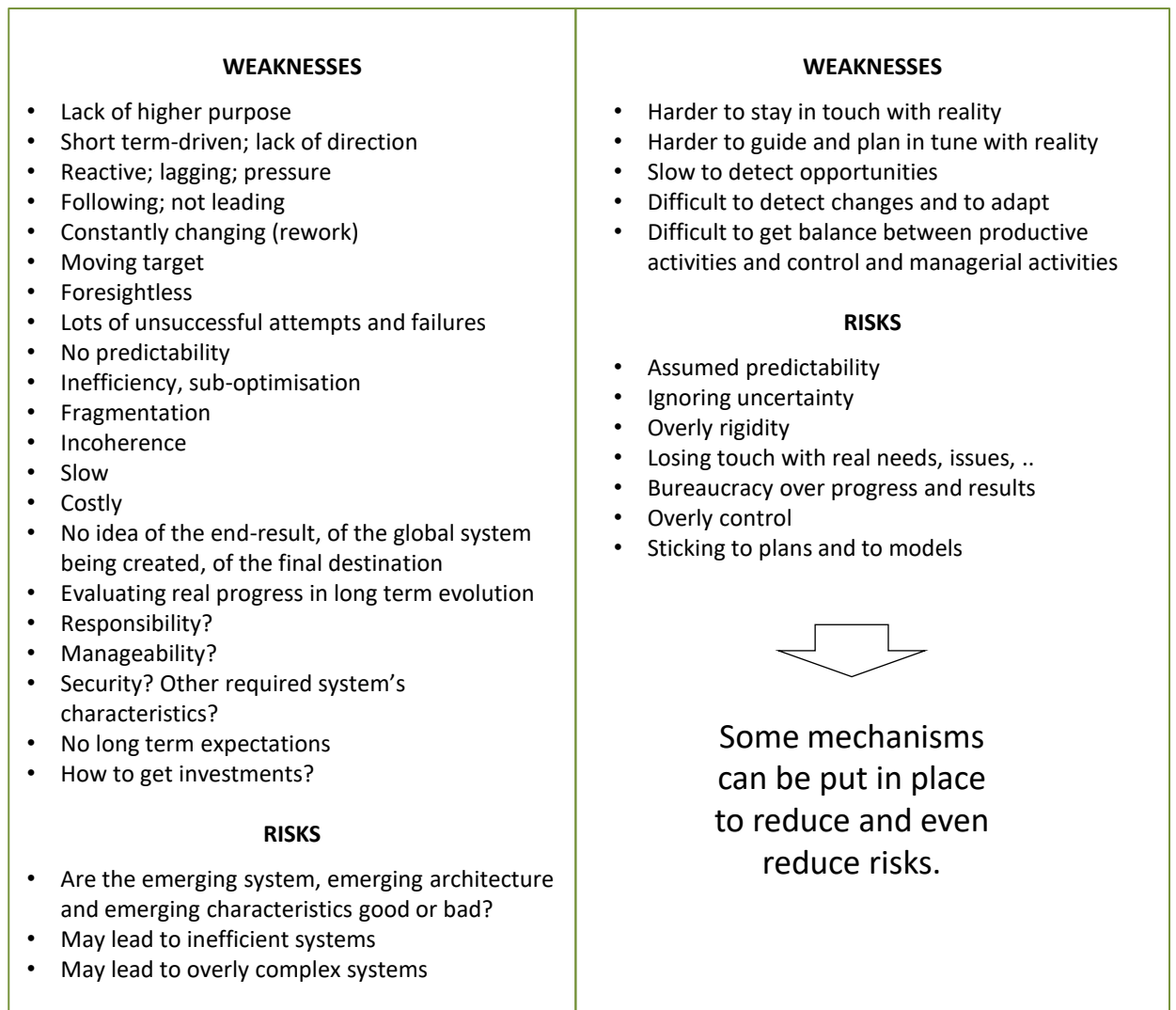
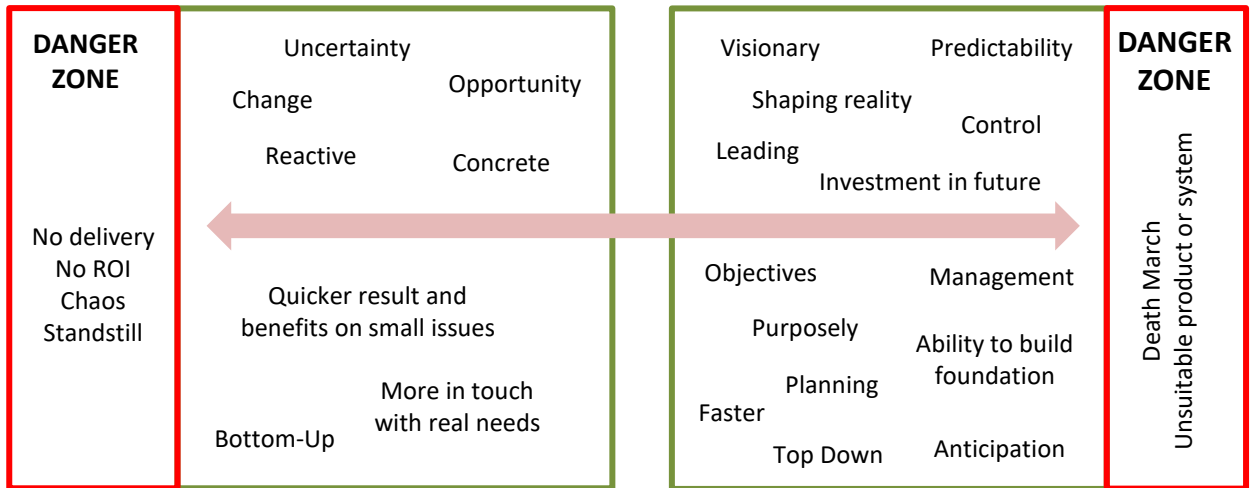
Chaos vs Structure (2)

30/10/2018



Chaos vs Structure (3)

30/10/2018



Chaos vs Structure (4)

30/10/2018

Chaos	Structure
Difficult to grasp	Easier to understand, Clearer
Hard to maintain systems	Easier to maintain
Hard to manage systems	Easier to manage
Hard to measure (except I/O)	Easier to measure
Hard to optimise systems or their environment	Easier to optimise
Difficult to change	Easier to change
Risk of side-effects	Smaller risk for side-effects
Hard to react in case of problems	Easier to react
Lesser likely to be scalable	More likely to be scalable
Confuses the mind	Guides the mind (thinking)
Steep learning path	Smoother learning path

STRUCTURE IS IMPORTANT

A structure means something organised and defined. It doesn't mean "fixed", "unalterable", "set".

A structured and organised environment supports the normal activities. We don't need to think too much about it. The focus can then be directed to creative activities.

Example: We can rely on public transport system (structure) to commute daily to work. If we had to spend two times a day to find out how to get to the job or to home, we would waste a lot of time and energy. This time and energy can now be spent to more creative activities. Similarly, some habits allow us to save time and energy.

We can play follow the structure or (temporarily) stepping out of the structure. A structure can be adapted when needed.

A structure (rules or any other formalised element) doesn't allow to deal with a specific situation. The structure doesn't allow this. We stop following it and do what is necessary, while staying as close to the structure. Once the job is done, once the situation allows it, the structure is followed again.

Chaos vs Structure (5)

30/10/2018

The world is
partly structured.

The world is
partly chaotic.

A **company** needs to be able to deal with both parts

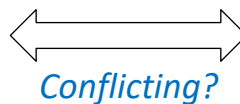
What we don't want: Chaos

→ Confusion, steep learning path, unpredictability, how to produce desired results in a chaotic environment?, difficult to understand and to control, inefficiencies, risks, costly, difficult to adapt, paralysis, ...

How To Solve this? Structure and method

Structure and method organise, bring order, provide clarity, allow efficiency, effectiveness and provide structural flexibility, allows to deal with a higher degree of complexity

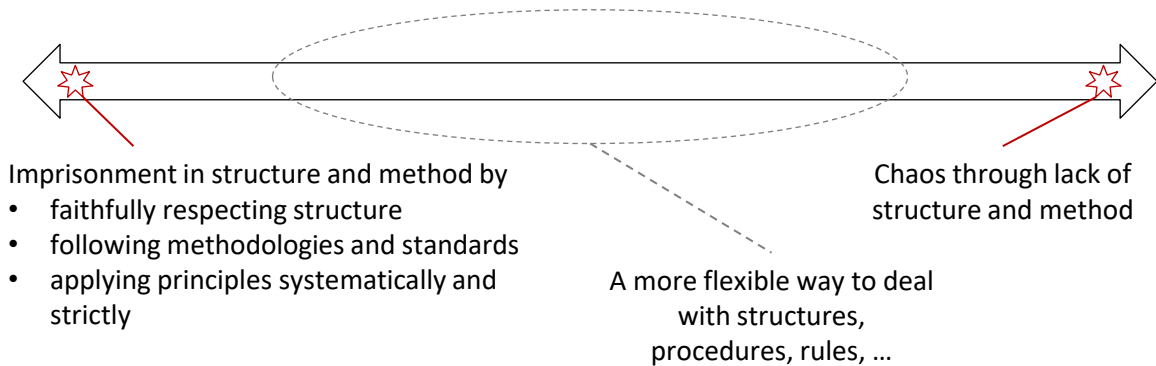
Need for manageability,
efficiency, effectiveness,
control, clarity



Need for ability to
change, flexibility,
adaptability, reactivity

Chaos vs Structure (6)

30/10/2018



Systems, structures, procedures, processes, rules are put in place to increase the efficiency and effectiveness. That's their purpose, their goal.

They should not hinder work !!

- 1) Implement structures (processes, ...) containing **alternatives** and **"free area's"**
The executioners can decide how they work within predefined free area's, boundaries, (example: case based processes, methodologies, ...)
- 2) **Following** the structure, processes, .. in a more **flexible way**
A situation which can't be solved by following the predetermined pattern (system, structure, ...). Either the pattern is adapted, be it temporarily and locally, or the particular situation is dealt with outside the predefined pattern. The executioner stays as close as possible to the predefined pattern and follows it again as soon as possible.
- 3) Implement structures **suitable for adaptation**
The more modular the structures, systems, processes are, the easier it is to adapt them.
- 4) **Capability of Vigilance**
Mechanisms offering Vigilance can be implemented in systems, processes, ... This is the implementation of monitoring mechanisms, controls, functions of environmental scanning, mechanisms to collect complaints, ideas, issues, opportunities, innovation, ...
- 5) **Adapt systems, structures, processes, procedures, plans**
They should improve and evolve over time.

Notes:

Systems, structures, processes and other guiding structures make sense, are valuable and are meant to help. It makes sense to follow them as long as they are suitable and to handle in a different way when necessary. Getting a result is more important than blindly follow predefined patterns.

Predefined structures are often considered as

- to be respected and executed literally all the time
- static
- unalterable
- dictating
- rigid

This assumption is often a matter of perception, interpretation, preconceived ideas and maybe a few other reasons. It is our own mind that deals with it in a rigid way.

They are very useful and offer a great flexibility, freedom and facilitate the functioning of the mind.

Our society is based on systems, structures, processes, and so on.

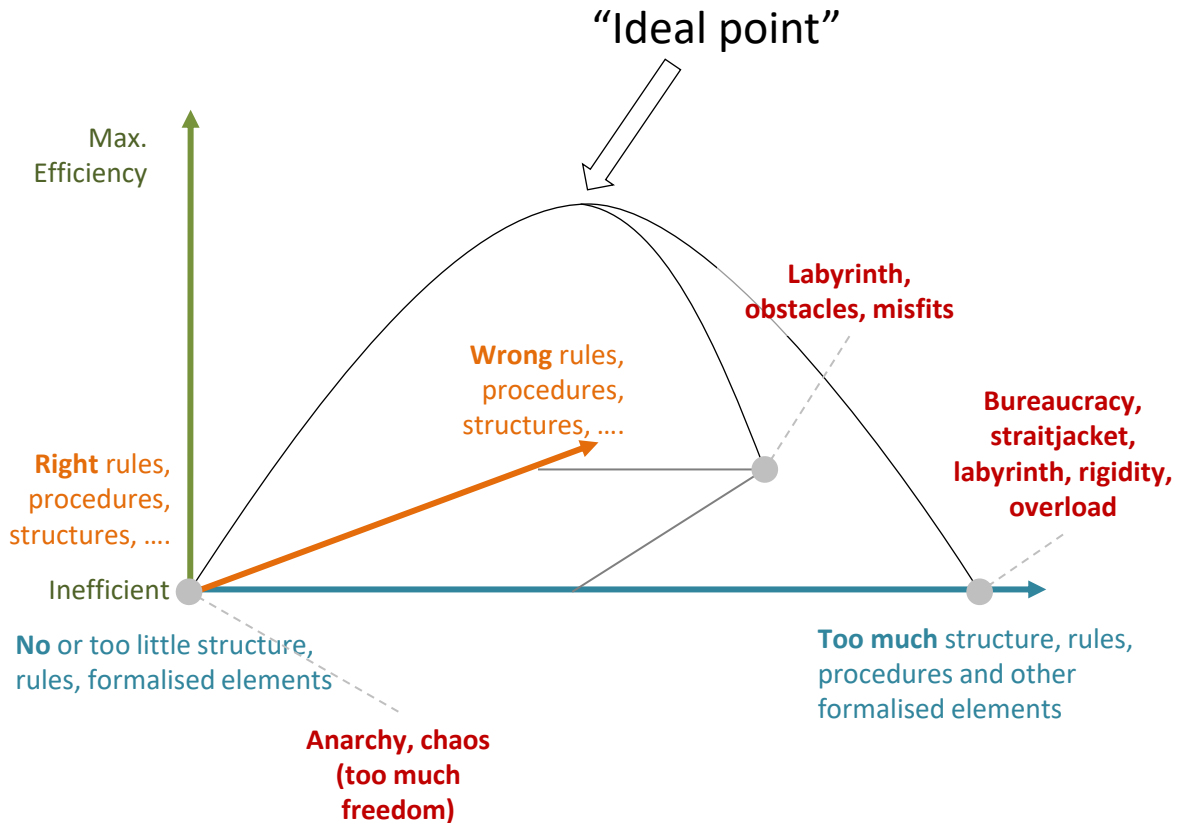
A lot depends on:

- How they are designed
- What higher authorities impose as mandatory
- How the structures are used (or misused)

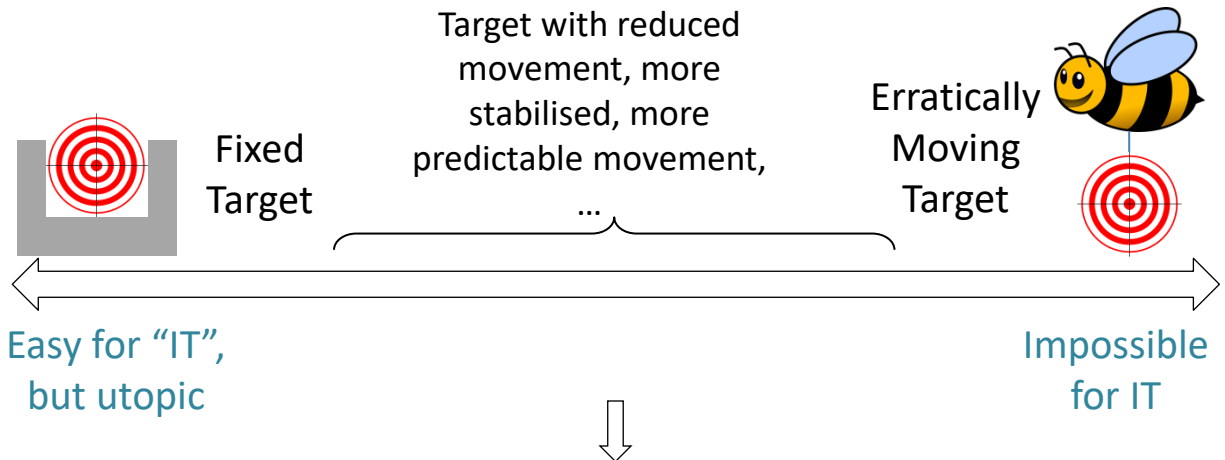
**We shouldn't blame
the idea of "structure"
because of
a wrong design, application or use.**

Chaos – Bureaucracy - Misfit

30/10/2018



- A right structure is needed. It organises and clarifies a way of working. We don't need to worry about solutions already provided by the structure. The structure forms a foundation.
- But the structure shouldn't be strict. We have to be able to decide that, to resolve a specific situation, the best way is to temporarily and to a minimal degree ignore the structure. Once, this necessity is over, we fall back again on the structure.
- A structure may have periods of freedom and 'unstructured' areas, domains, zones, ... of built in.



Stability required
for IS
development

Adapting to
changes in
business

Solution:

- Choosing target that is more/most stable
- Reduce movement of target
- Vigilance – shortening the path between target, movement, changes, issues detection and reaction
- Anticipate movements, changes, needs, ...
- Offering flexibility



IS development is a slow and arduous job. It requires a maximal stability. But business and the business environment is not static. Both needs have to be taken into account.

Variability of the Target

30/10/2018

Project lifespan: Several months or even years

Duration of Customer Satisfaction: can be extremely short
(individual & subjective)

Lifespan of a software system: 10? 15? 20 years?



Value of Customer Satisfaction

vs

Value of a good system

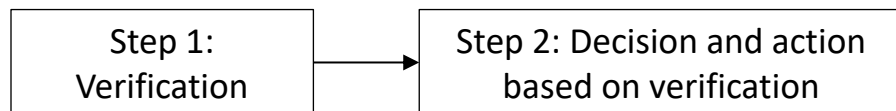
(Appropriate, well-designed, ...)

Control is a VERIFICATION

Purpose (Why do we control?)

- **Guidance**, readiness for further progress, course correction, **adaptation**
- Evaluation, validation and **learning**
- Checking for issues and preventing **issues**.

Control is necessary to increase certainty, to ensure, to learn, to guide, to correct or adapt, to align, to take new decisions and to prevent problems, misuses, aberrations, discrepancies, deviations and regressions. It is necessary for decision making and to keep things on track.



Notes

Control and control mechanisms require effort and time. They cost.

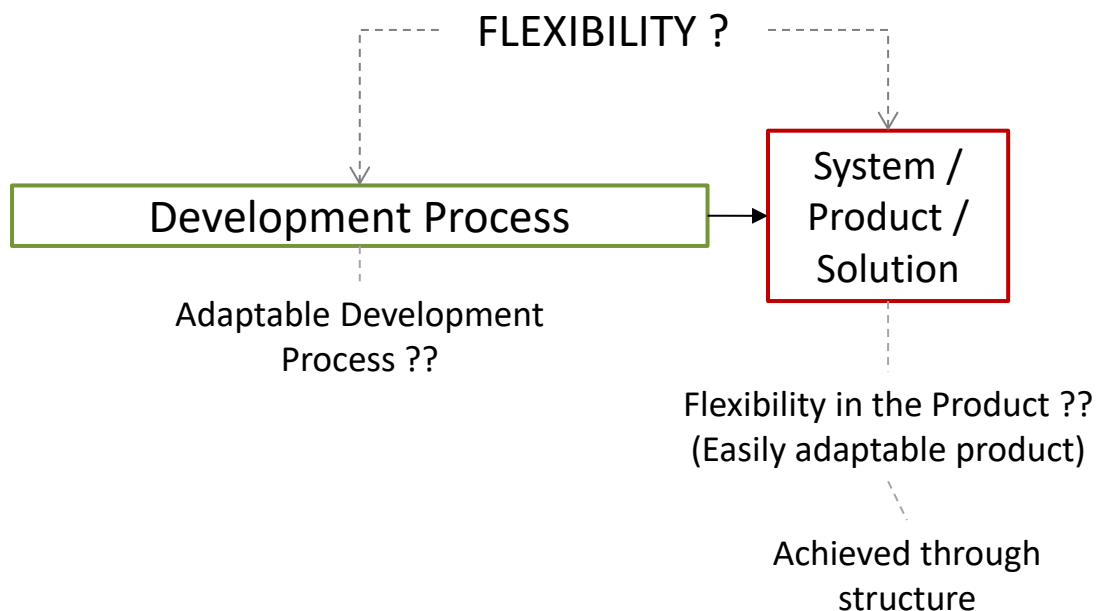
Problematic uses (misuse)

- overly guidance (usually at the cost of other important things); to pilot something that doesn't need or want to be piloted
- Control as verification if we stick to plans, respect rules, as a way to impose rigidity
- tool to exercise and increase power over something)
- freedom limiter (where freedom is required)
- creativity killer (imposing own ideas, decisions, ...)

This might be the consequence of the attitude of the controller and the decisions and actions following the control. Control is then misused in an inappropriately exercised role or in a unhealthy relationship.

“Need for Flexibility” doesn’t mean anything

- Where do you need flexibility ?
- What kind of flexibility is needed ? How much?
- Why do you need flexibility ?



We are interested in flexibility offered by the product’s structure and internal organisation. This allows

- to adapt easier to unforeseen circumstances
- to adapt the course of a strategy when needed
- to implement new ideas
- to more quickly seize opportunities

Flexibility in the Development Process



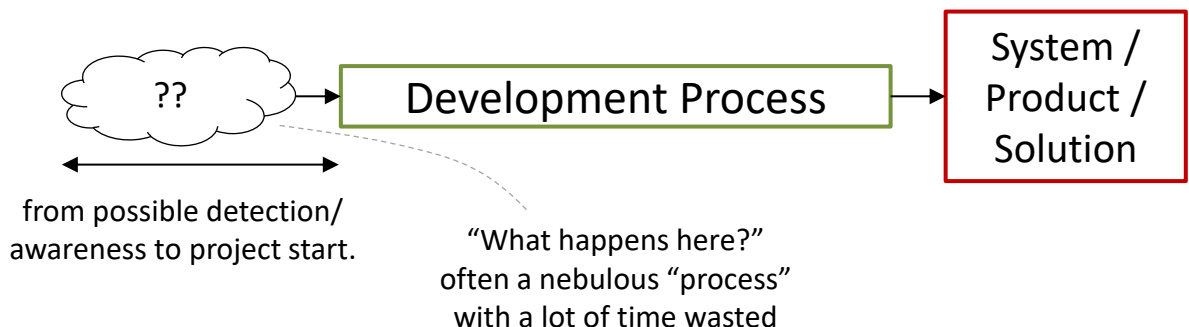
Inevitably, there will always be some need for changes in the development processes.

The necessity of flexibility in the process is a symptomatic solution to solve issues due to hasty and wrong decisions caused by a lack of understanding, which is caused by lack of real analysis, sloppy thinking or by a lack of good documentation.

It's better to solve the causes instead of the consequences. It's better to do things right.

"We may need flexibility to speed up the development process"

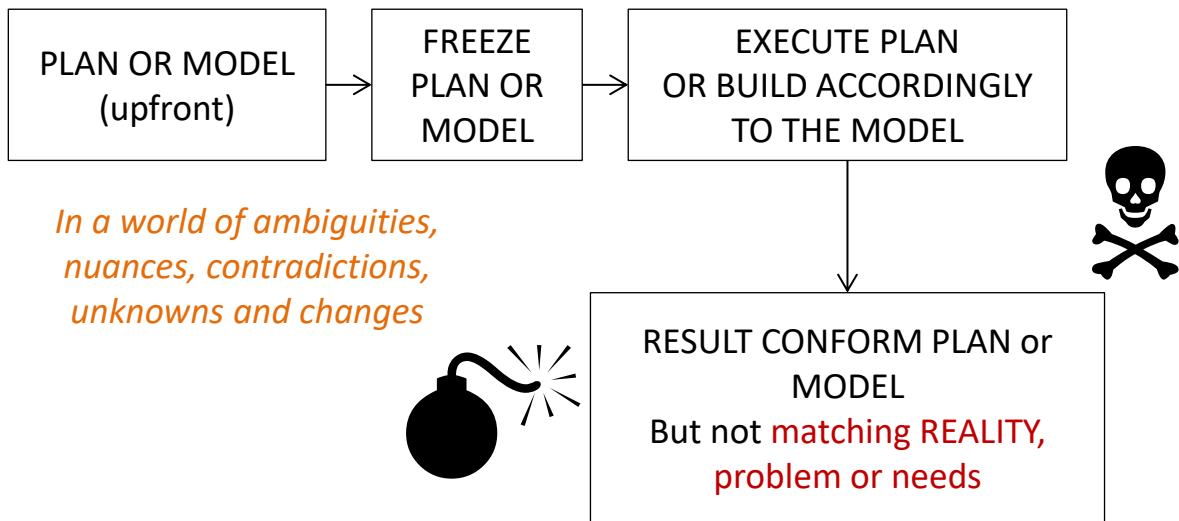
This is an **assumption** that is very **questionable**. Every change implies rework, thus some waste.



Everything that happens before the elaboration and submission of a project proposal is often very confusing and inefficient. How much time has been wasted there?

Notes:

If a lot of flexibility of software development process is demanded, it is a symptom of lack of mastery in the discipline; lack of understanding in the environment, system and mission; bad guiding and so on. Flexibility should, in essence allow to respond to unforeseen issues, not to alleviate lack of competencies, a bad approach or any other similar type of issue.



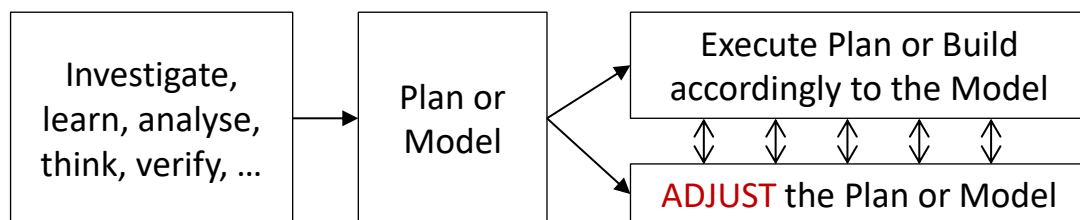
The risk for a failure increases when:

- the upfront plan or model is **detailed**
- the planning or modelling has been done, respectively, by a planner or a modeller **lacking** of **appropriate skills, knowledge** and **insight** and who is **not in touch with the reality**.

The mistake is NOT the upfront planning or modelling itself.

This model is based on the assumption that the plan or model is perfect and the denial of the existence of uncertainties, unknowns and changes. The mistake people made is to refuse the need for adjustments and adaptations later in the process. This is rather an elementary issue.

A Rudimentary Model to Plan or Model “Upfront”



Notes:

The adjective “planned” is often interpreted as “determined upfront, sometimes by ill-informed or unskilled people, frozen(at least for some time) and executed as is”. This interpretation has a very negative connotation.

Validation

Confirmation that something is right, that the project is still on the right track. Validated elements has been investigated, analysed, verified and found correct. They are ready for further use. They should be stable. They are not meant to change anymore. Only, in exceptional circumstances like unforeseen events, they still can change if beneficial to the project. Change means also a delay, a cost and a risk.



Freezing

Putting yourself or the project into a straightjacket until the end of the project.

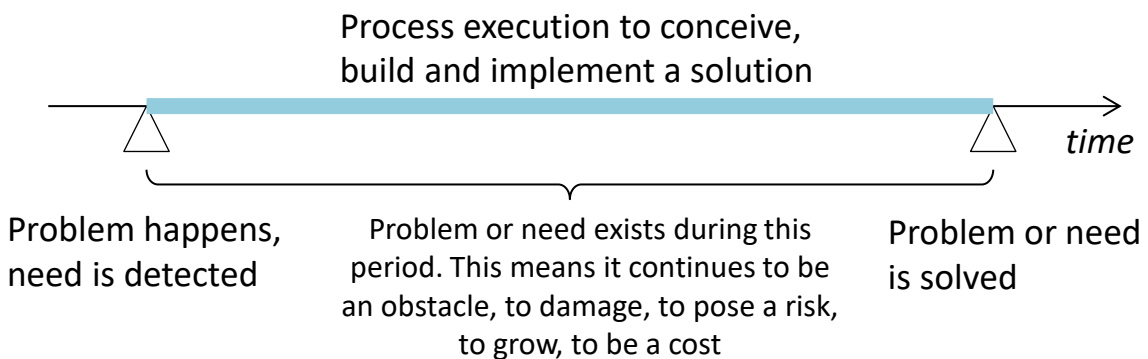


Proactivity (1)

30/10/2018

Proactivity is NOT about building features which are not asked for, which are unnecessary, just in case they might be needed.

Reactive approach is acting in response to an issue that has already become a problem, responding to a demand



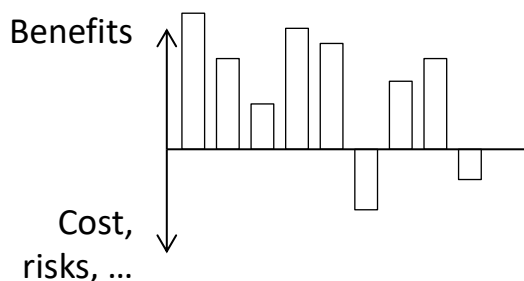
How to limit these costs and risks?

Proactive approach is to respond to potential problems and to future needs before they take place.

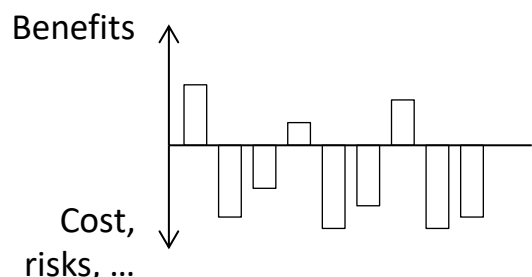
GOAL:

Benefits of profitable measures

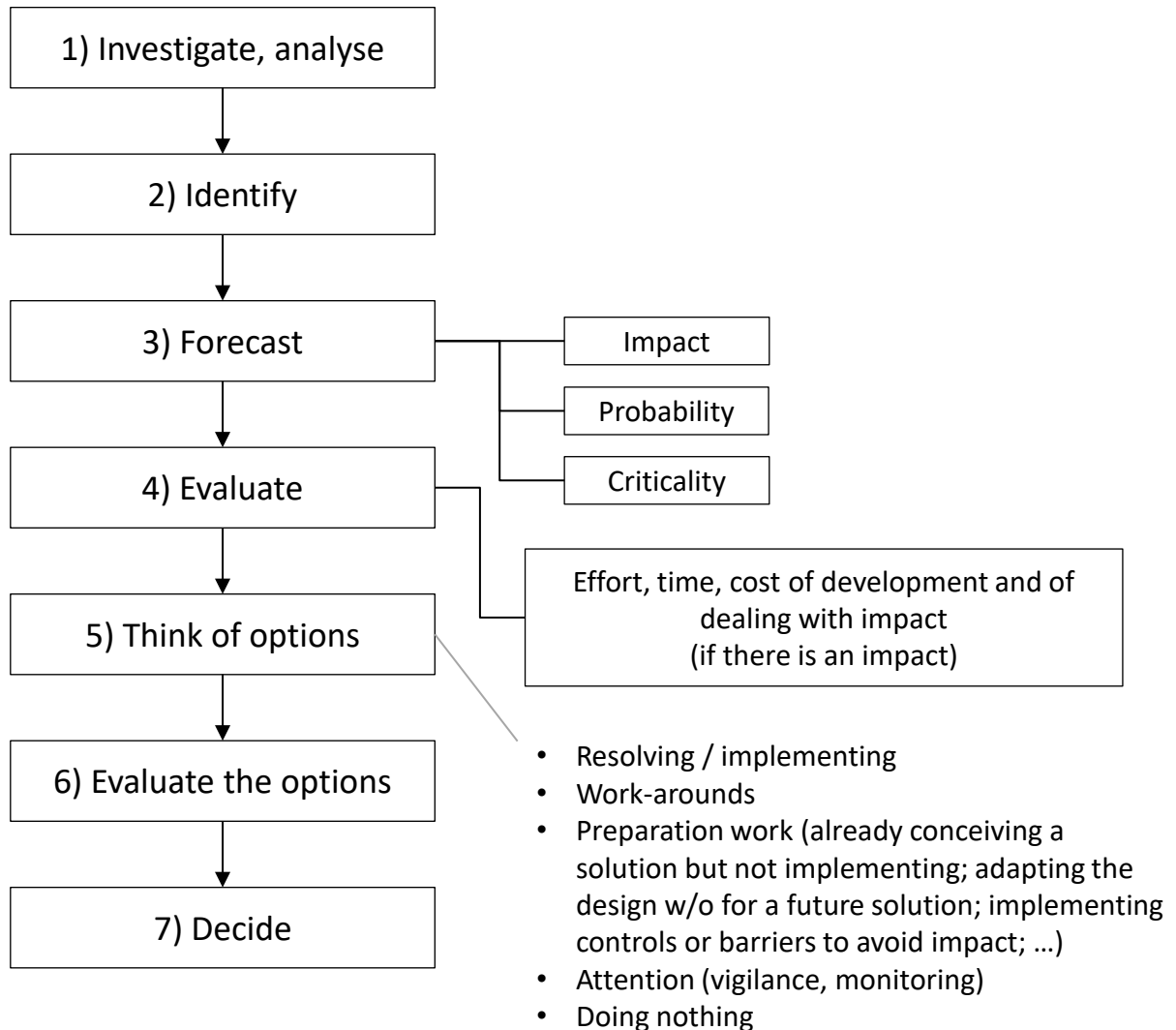
> Loss caused by unfortunate measures



Result of
right proactivity



Result of
failed proactivity



Proactivity is a conscious decision and an intentional action taken for each individual issue or opportunity.

Proactivity **can not be successful without understanding** the systems, the environments, the involved domains, the evolution of the company and without empathy. It requires mastery of the systems analysis discipline as well.

Notes :

This formal flowchart is often reduced to a few questions like : “Will this be useful?”, “Does it cost a lot of effort?”, “Is it good to implement it now?”, “What is the best way to deal with it now and later?”

- We don’t wait to implement breaks in cars until the driver needs it. Proactively, we forecasted that the driver will need breaks.
- We listen to the weather forecast to evaluate if we proactively have to take an umbrella or wear raincoat when going out.

REACTIVE ATTITUDE / APPROACH

- Allowing the environment and events to control the course of actions; following the events as they happen
- Focus is on problems
- Being overwhelmed by problems, large back log.
- Under permanent pressure
- Not feeling in control
- Lacking of foresight
- Being unprepared
- Increased uncertainty
- Forced by circumstances to take decisions and to act
- Dissatisfied customers. Frustration. (experiencing many problems, waiting before they got solved, working under pressure → lower quality, ...)

PROACTIVE ATTITUDE / APPROACH

- Avoids or diminishes future problems and firefighting
- Requires being truly competent
- Focus is on what can be changed and improved.
- Driven by higher values (get the most out of the environment)
- Define the course of actions
- Not just seeking to solve problems presenting themselves, but actively study the situation and look for opportunities.
- Much more planned progress.
- Increased grasp on future and evolution.

From Problematic to Innovating IT

30/10/2018

5. Innovation

- Services
- Working methods
- Organisational structure
- Capability acquisition
- Business Model
- Partnerships

About sizeable innovation in business (not about small innovations with little impact)

4. Driving existing business

Increase business volume, market share, profit margin, quality of service, reliability, reputation, ...

Strong & independent IT department

3. Alignment

Simply responding to present needs. Can be a normal and limited delay. Business is fairly satisfied.

Pulling Upwards

2. Misalignment

Not responding to the present needs. The business community can work. Existing IT solutions support some business activities.

Dragging Downwards

1. Problematic Enterprise Information Solution

IT solutions are more detrimental to the business than it delivers benefits.

Weak & dependent IT department

TCO – Total Cost of Ownership^{30/10/2018}

SET-UP

- Purchase research & purchase process
- Hardware acquisition
- Off-the-shelf software acquisition
- Licenses, warranties
- Customisation
- Software development project
- Data migration
- Testing
- Set-up and deployment
- Training & documentation
- Development or configuration of interfaces to other systems

OPERATIONS

- IT Personnel
- Energy
- Audit & Testing
- Administration & Maintenance
- Support
- Security
- Insurance
- Upgrades and patches
- Training
- Change management cost
- Changes and enhancements
- Downtime, outages and failures
- Disaster recovery
- Depreciation
- Data centre and Infrastructure (acquisition, installation, configuration, management, maintenance, upgrades, replacement, ..)

LONG TERM

- Scalability expenses
- Replacements

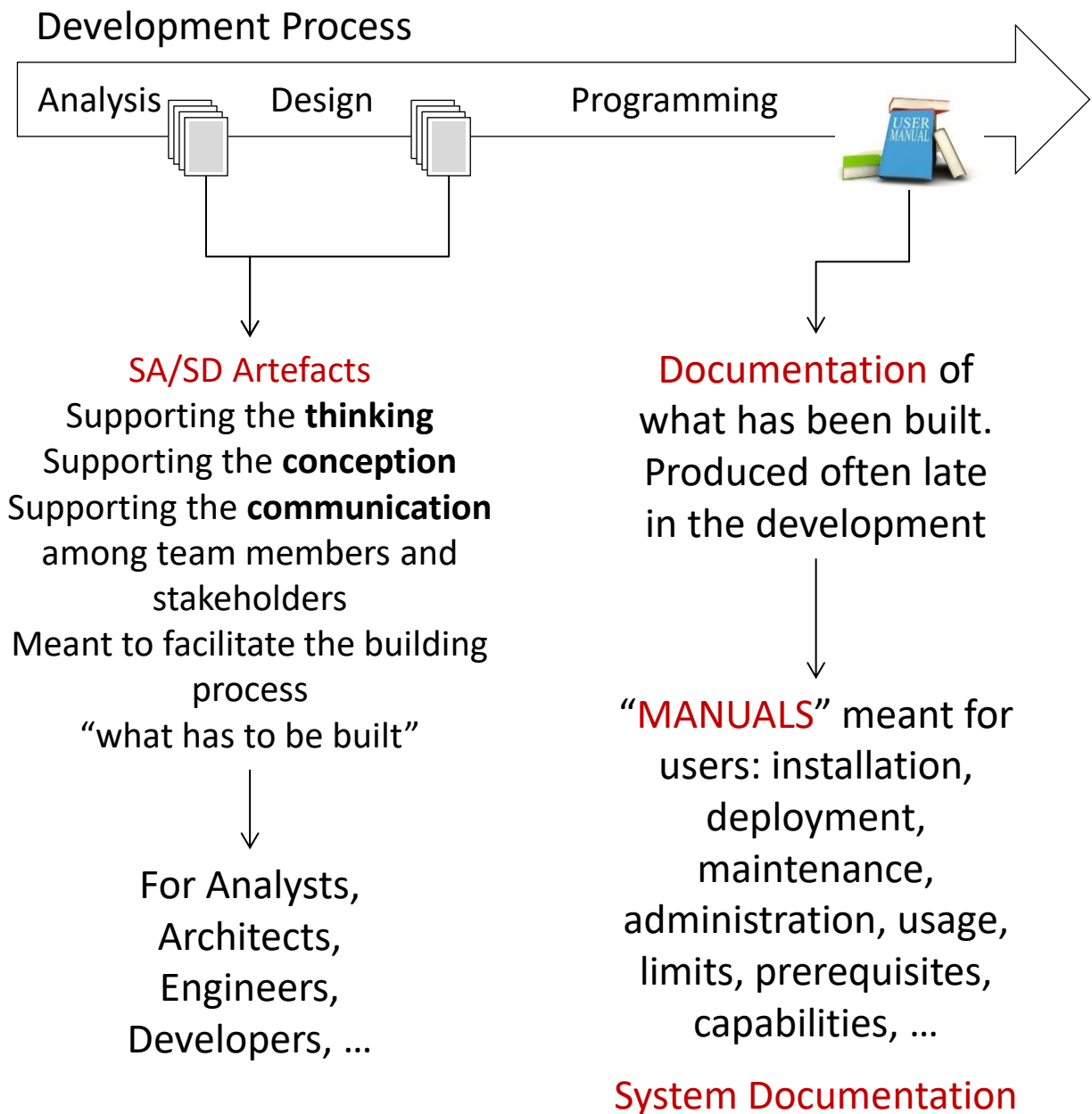
DECOMMISSIONING

- Data export / migration
- Dealing with the archived data

- A 'solution' may not solve the problems or may solve the symptoms and create more chaos.
- A system may create problems. Cost for resolving these new problems.
- A solution may not be optimal. A better solution could have lowered the cost or bring more benefits.
- A sub-optimal solution may put barriers, obstacles, limitations.

SA/SD Artefacts ↔ Documentation

30/10/2018

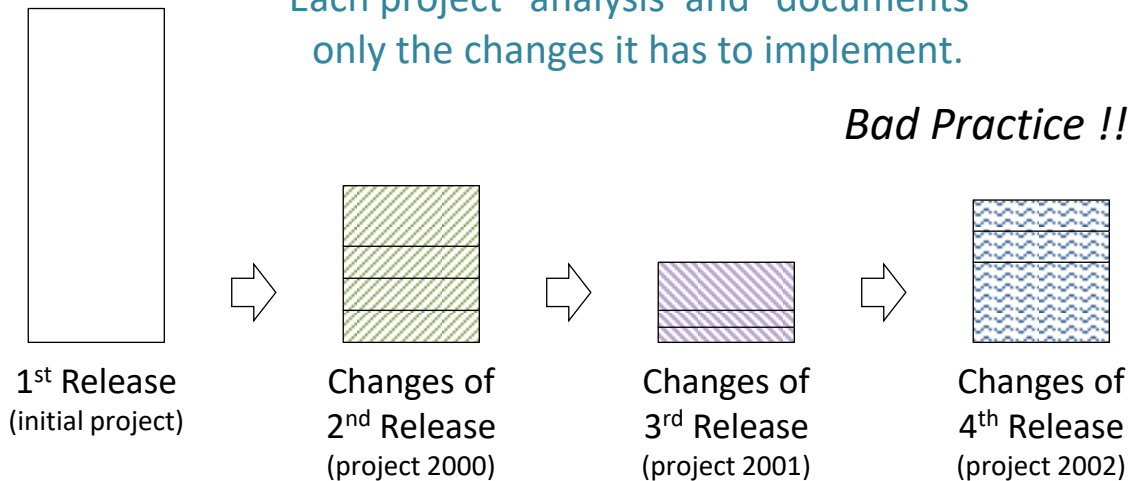


Serves as “input” in next engineering round.
(else you need to re-describe the “as-is” every time again. Former to-be becomes new as-is. No need to re-describe it.)

Analysing / Documenting Only the Differences

Each project “analysis” and “documents”
only the changes it has to implement.

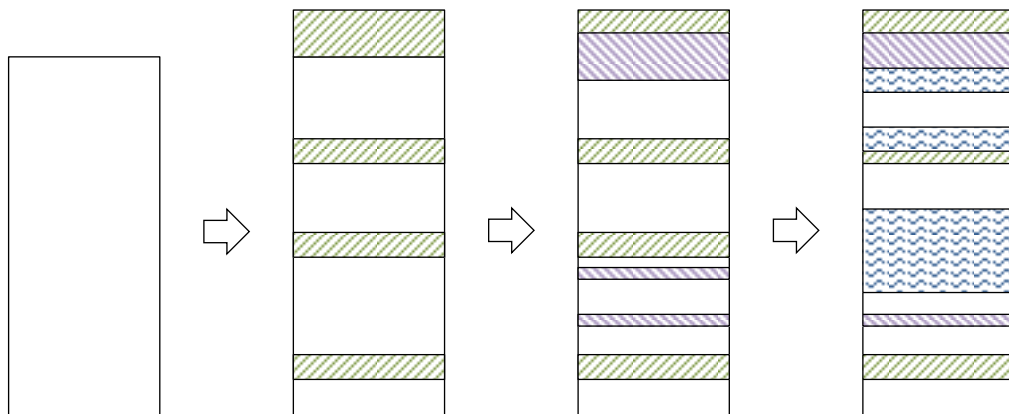
Bad Practice !!

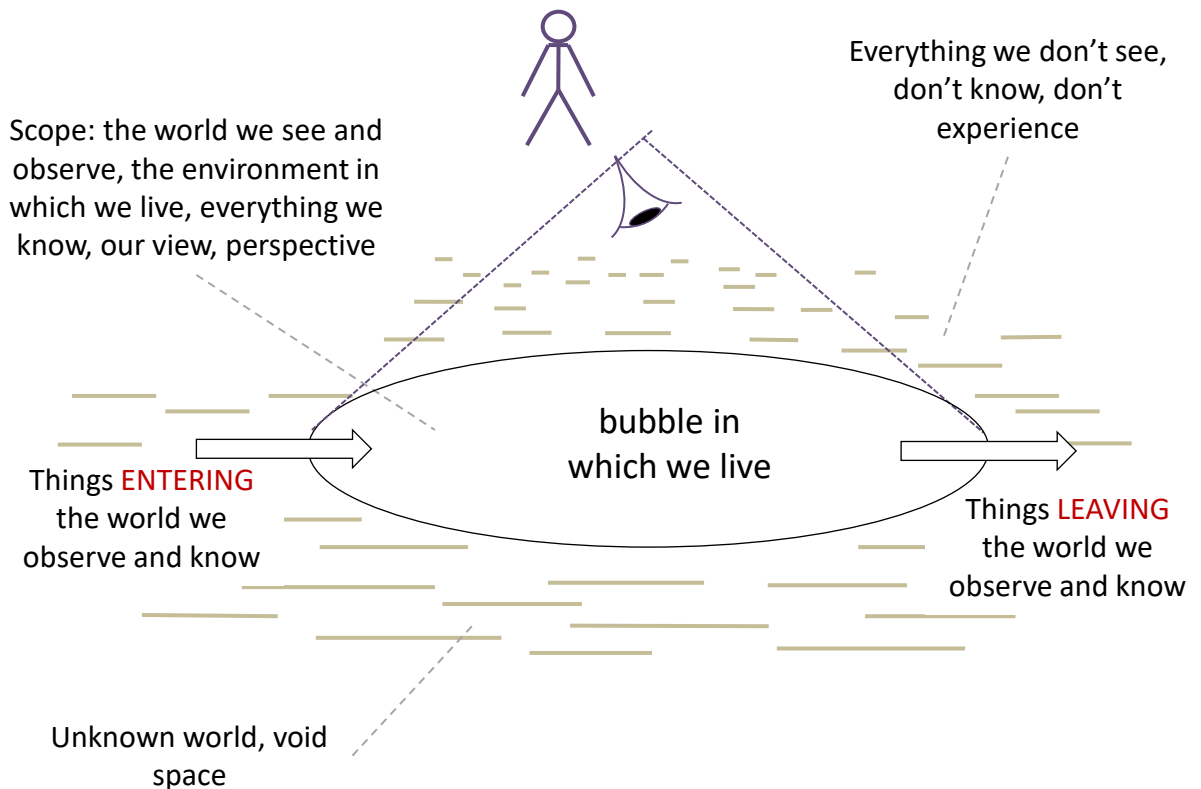


- Advantage: Easier, smaller documents; Changes are easier to locate
- Drawback 1: To understand the present system, one needs to read the artefacts of the original systems, then chronologically all the artefacts describing the successive set of changes. → reading a lot of unnecessary things which aren't true today anymore = waste & confusing
- Drawback 2: No clear view of the entire system! → Hindering the understanding → RISKS!
- Drawback 3: Think of many versions later: Chaos ! This is unsustainable.

GOOD PRACTICE:

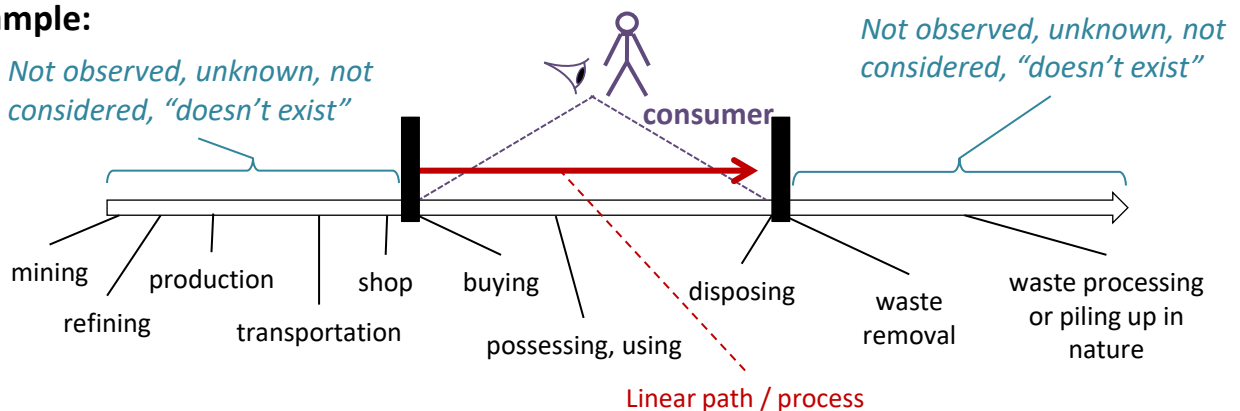
Document changes in the current system documentation/artefacts. This changed document becomes the new current system's documents.





Example:

Not observed, unknown, not considered, "doesn't exist"



Notes:

The world we consider is often constrained within a defined zone. It often stops at some abstract or concrete borders (example SIPOC). We may consider a process until the perimeter of our own department, until the product is delivered to the client, until the information is delivered to the other organisation. Our mind thinks within the perimeter defined by a scope, a mission, a problem statement, a demand, a deadline, a month, a year or other limited period, a concept of action-reaction, our house, our street, our city, ... our live. Our mind often limits our thoughts to a zone or timespan. Our thinking is not integrated into the larger world.

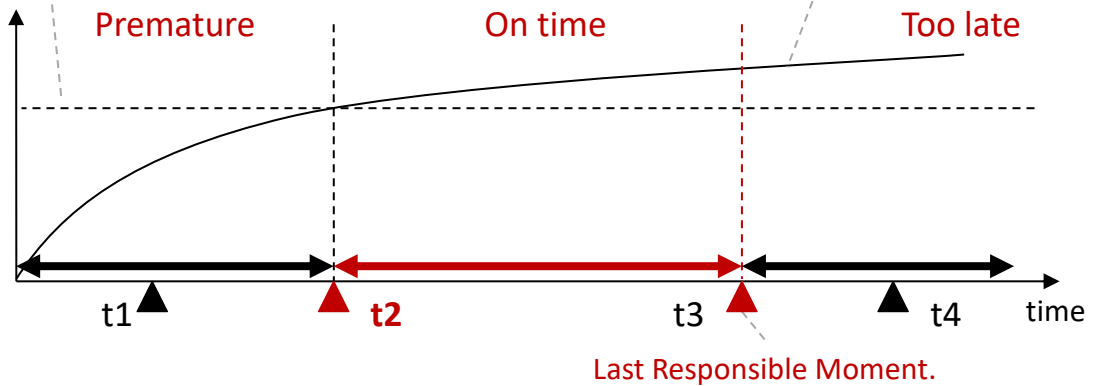
We consider our job and assume what we do integrates well into the larger picture, or, someone else will take care of it. Failing to consider and think outside our perimeter causes a huge amount of issues.

Last Responsible Moment

10/01/2020

Level of confidence of being well-informed (expertise, insight, confirmations, guts feeling, intuition, wisdom, ...)

Quality of conditions to take a decision



t1 : Insufficient information and insight; little certainty; closer to guessing; higher risks, ...

t3 : Last Responsible Moment. A decision taken here is still on time, but it can not be postponed. A decisions taken beyond the LRM is taken too late and will have more negative consequences. Question: How to determine the LRM ?

t2 : Decision is the **ideal moment**. Confidence of being well informed, of agreements and so on is fairly high or at acceptable level.

- Each decision not taken is an uncertainty. It may block work and other decisions.
- Each taken decision removes some uncertainty. It creates clarity and increases the stability. From the moment a decision is taken, it work can be based on it.
- A decision that is not taken can be a blocking factor.
- $t3 - t2 =$ wasted time if decision is taken on t3

t4 : Decision is too late. Sometimes, it isn't even necessary anymore to take a decision.

**Take decisions as early as possible,
but not too early.**

The curve is **illustrative** and depends of the specific situation. Factors for decision making can and have to be improved by actions:

- Information has to be gathered.
- Insight has to be acquired.
- Verifications are also useful.
- Agreements can be reached.
- Certainty can be increased.

Note: We may argue that taking a decision late leaves the options open. There is no guarantee that the decision will be changed. At one point a decision must be taken anyway. A taken decision can still be changed. So, there is no real reason to take decisions late.



BUSINESS-IT ALIGNMENT

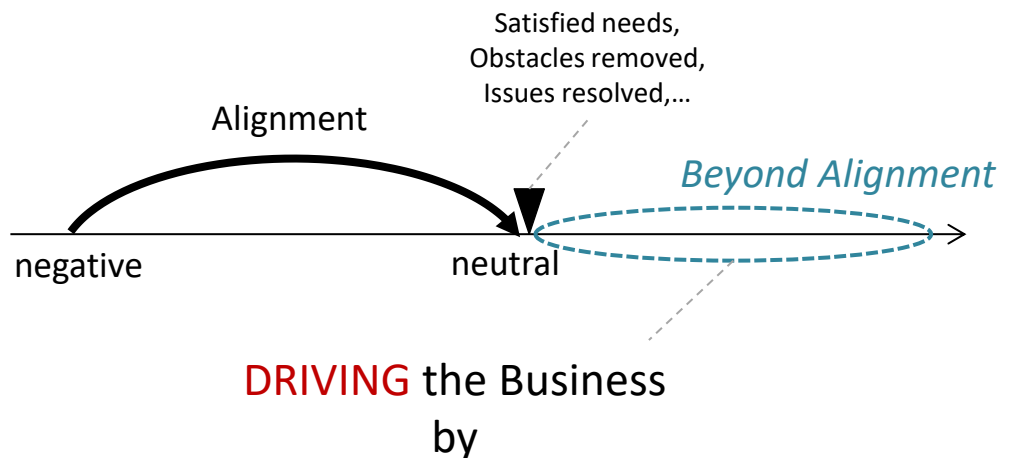
Business-IT Alignment

What do we want or need to Align? What can be Aligned ?

- Strategies, plans and priorities?
- Plans and allocated budgets?
- Overall capabilities with necessity and demand?
- Operational functioning and results?
- IT efforts and investments with business objectives?
- Systems features with the demands?
- Implemented logic with business logic?
- The limited and acceptable amount of submitted demands and a low rate of complaints
- ...

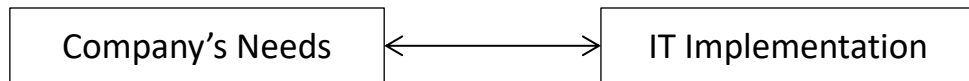
Alignment is about achieving a correspondence, a matching, conformance, an appropriateness, harmony, balance.

How about aiming BEYOND the state of ALIGNMENT?



Achieving perfect Alignment might be unrealistic.
The objective is to achieve the best possible alignment.

1. Alignment in the **PRESENT** – The Operational Level



- Gap – Unsatisfied needs
- Limitations & Obstacles
- Complaints
- Issues
- Disappointing results

1. **Detection** and **Identification** of needs, obstacles, ...
2. **Diagnosis**
3. **Adaptation** of the IT implementation

It is unlikely to achieve good results with demand-based reactive approaches. The lag is too important and too much time and resources are wasted.

It is possible to put in place mechanisms (early detection) and solution to reduce this lag and to facilitate adaptation (flexibility through structure).

With the alignment of the situation as it is today,
companies also want to be aligned in the future.

Note:

Business-IT alignment is not a situation that can be reached and then forgotten. This state has to be maintained in time.

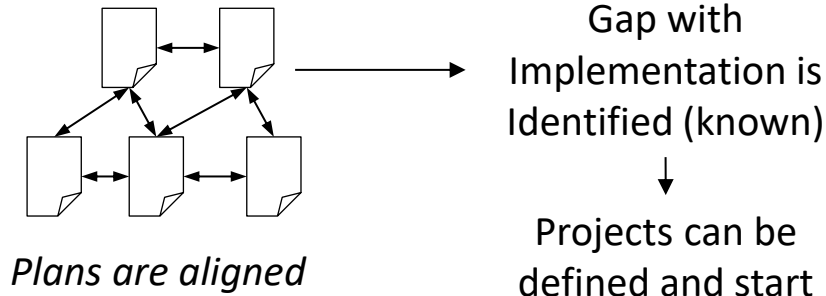
Business-IT Alignment

Maintaining the state of Alignment

With the alignment of the situation as it is today, companies also want to be aligned in the future.

2. Alignment in the **FUTURE**

A. **PLANS**



Alignment of plans with

- Vision, Mission, Values
- Business Goals
- Business Drivers
- Business Priorities
- Business Timeline
- Business Budgets / Investments
- Business Functions
- Business Capabilities
- Business Model
- Business Operation Model

Important Note:

This is only a one-side set of targets to align with (the business perspective). Other targets are those related to the system's health and long term survival (see: System versus system Usage/Deployed Activity)

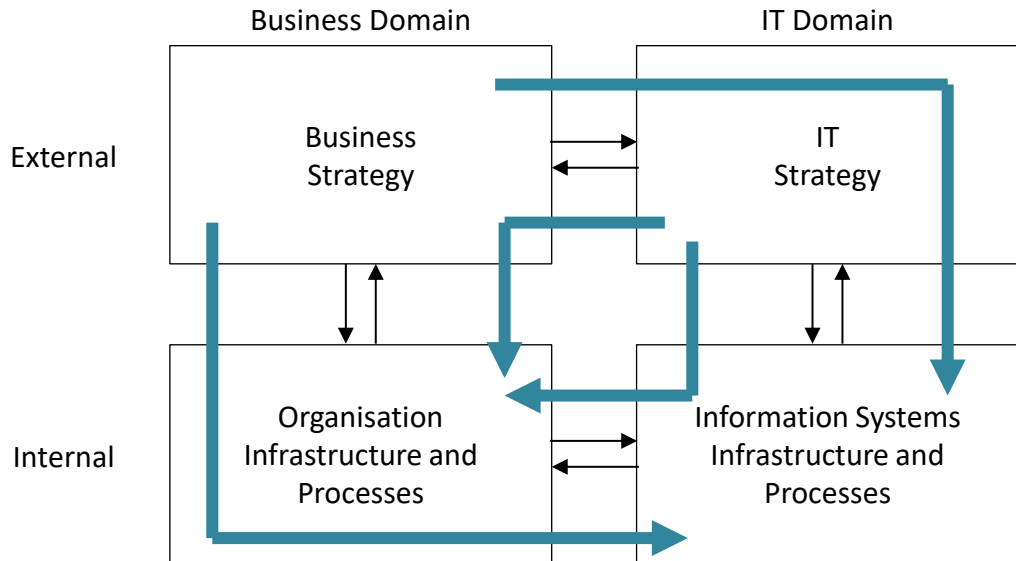
B. **PROACTIVITY**

The understanding of the environment, of people, of forces, of mechanisms, of evolutions help to align proactively (see subject "proactivity").

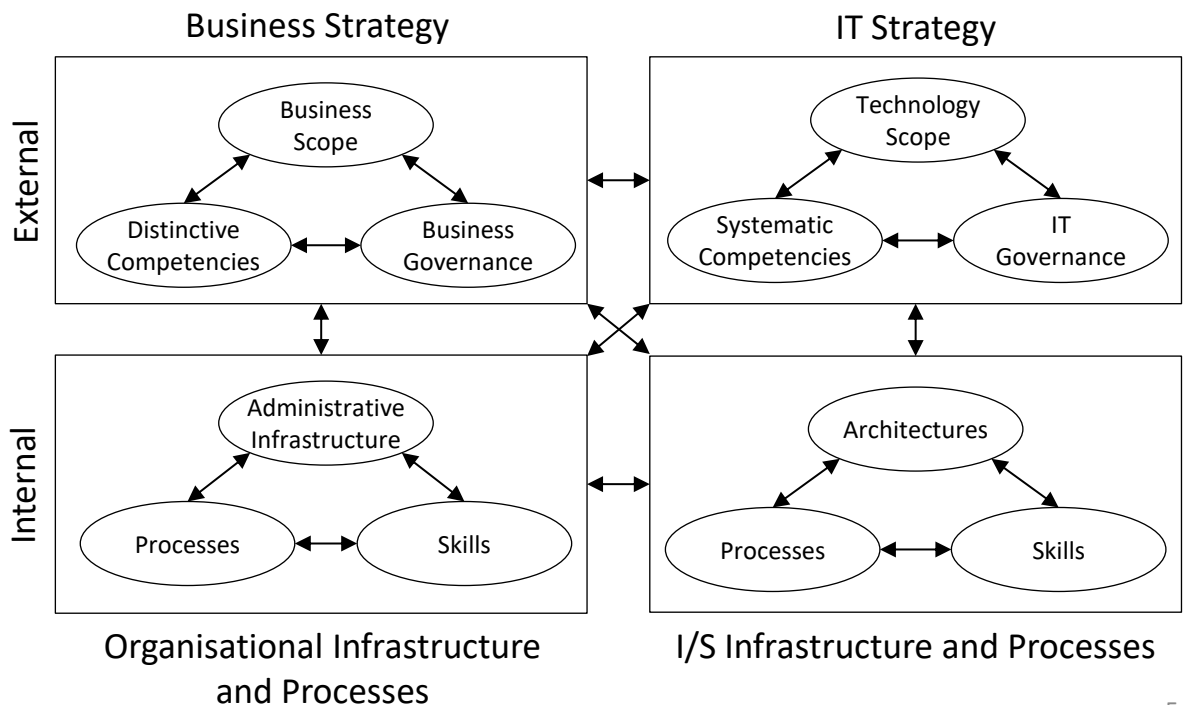
Alignment

Henderson – Venkatraman

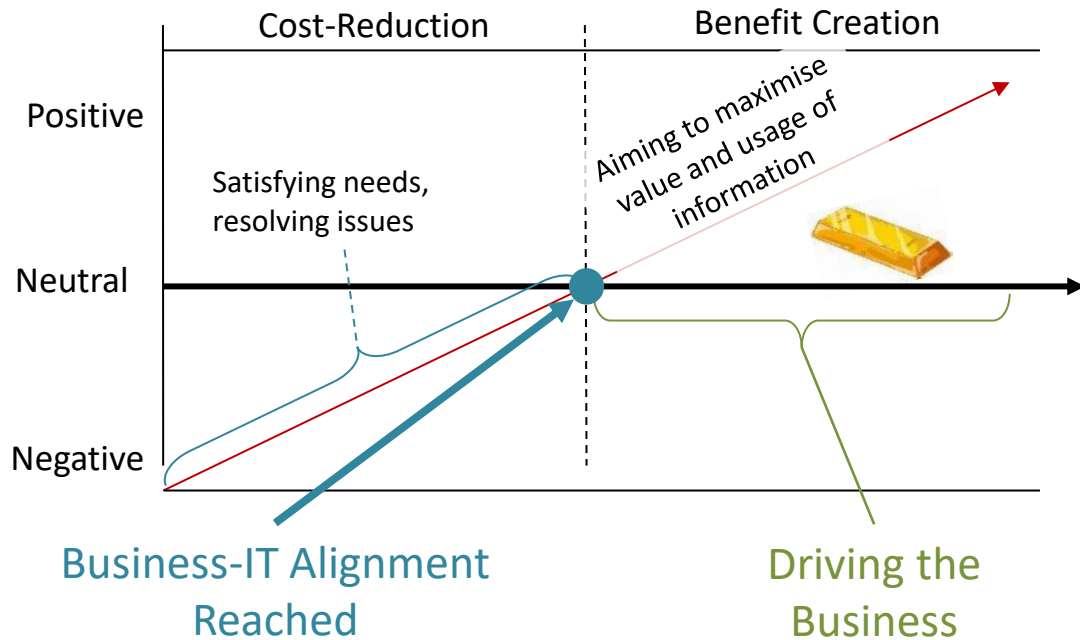
Strategic Alignment Model (SAM)



FUNCTIONAL INTEGRATION



Beyond Business-IT Alignment

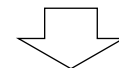


*Business-IT Alignment =
 “Successfully accomplishing what the business
 community asked for and what it needs”
 = a single perspective (business perspective)
 based on limited insight in systems and in
 possibilities of Informatics*

- Increasing business volumes and/or profit
- Supporting the business beyond the needs
- Lower the cost and risk beyond the demand
- Beyond the expectations and demand of the business
- INNOVATION

+

More information for
 strengthening and optimising the enterprise
 & for better management and guidance.



Impossible to be achieved by the business community

Hard to be achieved by 'IT'

Can be achieved by **Business Informatics**

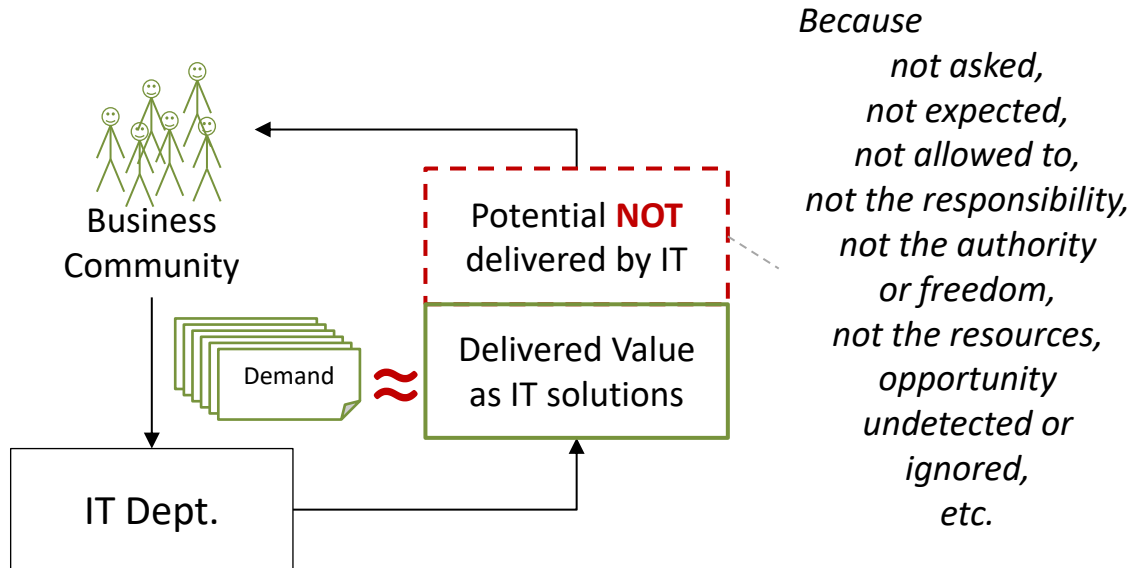
WARNING !

The quality of the enterprise as a system must not be ignored in this picture !!
 (perspective of system vs activity)

Business-IT Alignment

The IT implementation allows the business community to deploy and execute business activities in the way it intend to do it.

The IT implementation correspond to their demand and expectations and delivers the operational and financial expected results. It meets the business objectives.

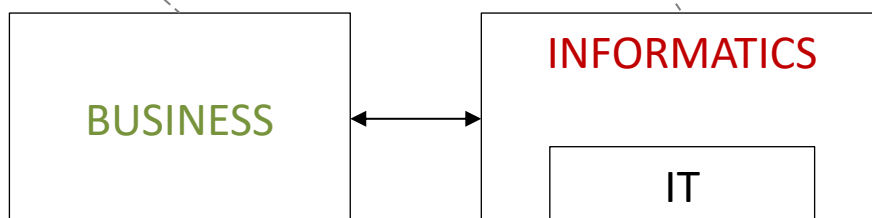


1. Business Community has a high level, general, business perspective and user's perspective, which is very shallow and narrow.
2. Business knowledge, responsibilities and goals can be siloed.
3. Business demands are oriented towards activity and results obtained through activities and don't take the shape and health of the systems into account.
4. This approach can lead to a piece-meal & stove-piped development.
5. What if the demands do not ask for the optimal (much better) solutions or even don't ask for the right problem to be solved ?
6. It is reactive.
7. Mostly operational (practical) or tactical demands (no big picture, no real or direct strategic contribution)
8. IT won't deliver (much) more than what the business asks for.
9. Creates pressure
10. Kills innovation

Untapped Potential in IT and Informatics

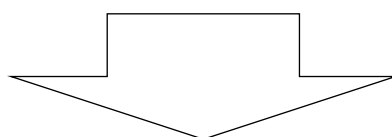
How to do business ?

How to use information ?



How to use information technologies ?

IT as sub-discipline of Informatics



Untapped
potential value

Untapped
potential value



IT

Informatics

Possibilities technologies, features
and technological offer to the
Business Community create value,
which the Business community is
not aware about.

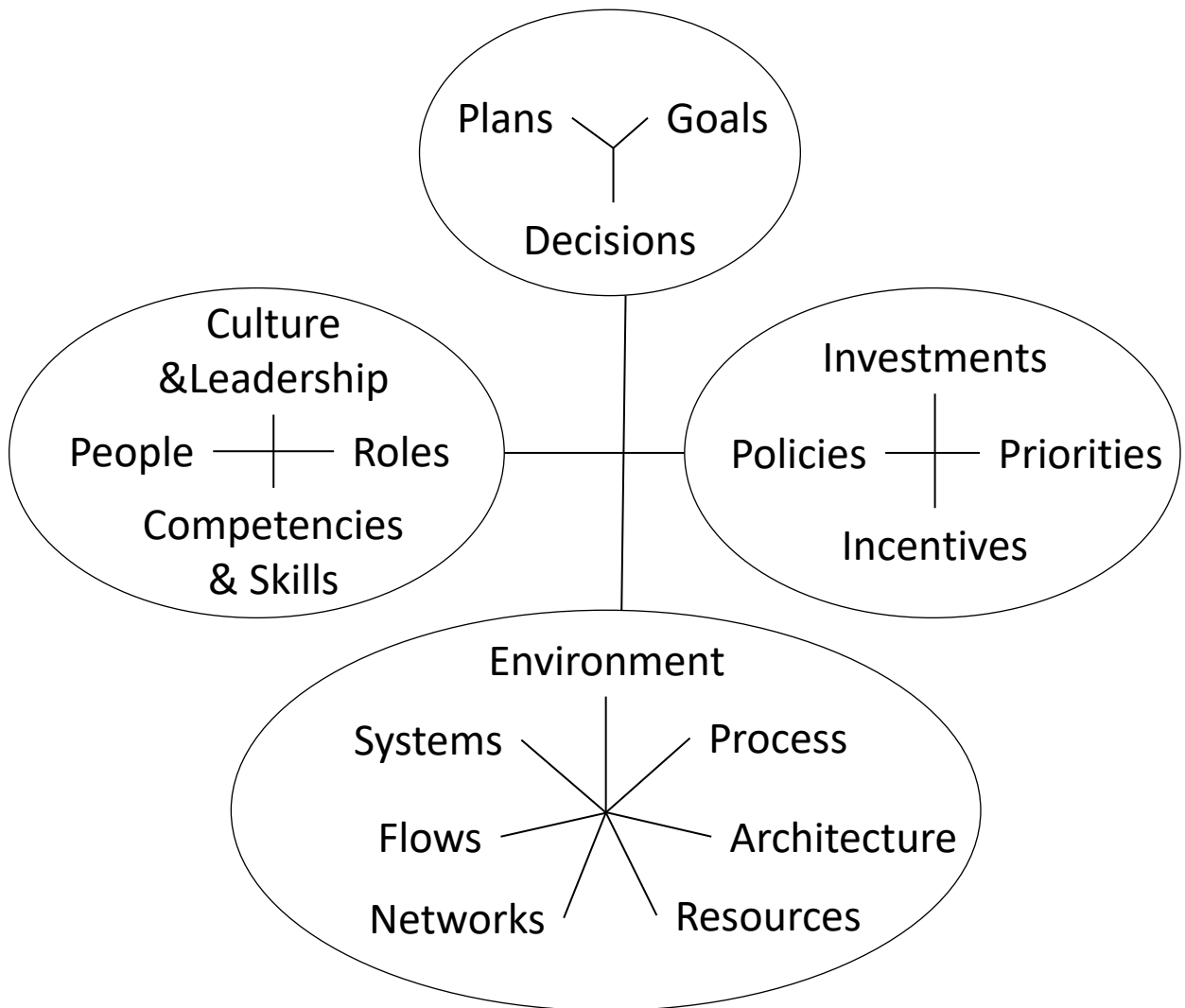
Room for
alignment

Technology = Cost Saving

Informatics = Cost Saving +
Business Driving

Global Internal Alignment

30/10/2018



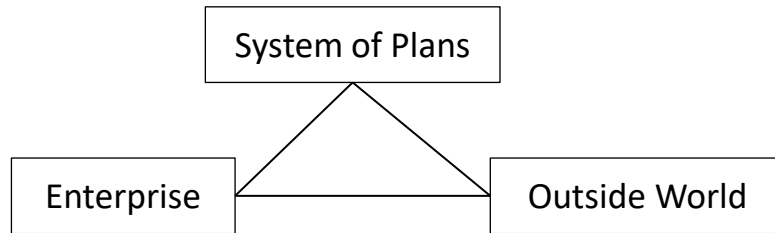
Notes :

Why is this important?

- 1) All these aspects are expressed in information
- 2) They have to be supported with information
- 3) Information systems have to be aligned with this.

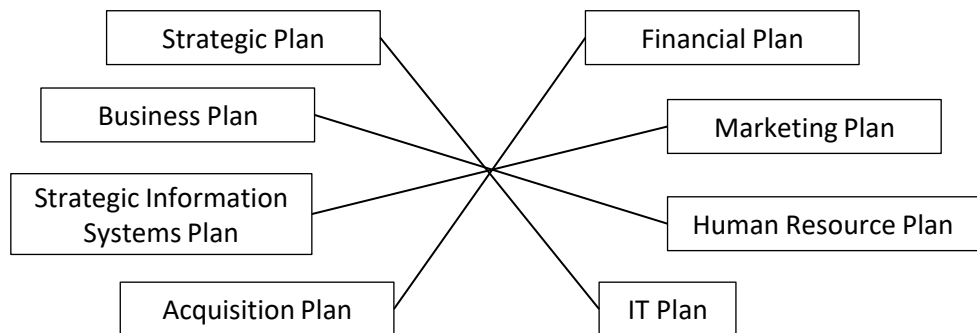
It is the **Analyst's** and architect's **job** to contribute to this alignment by identifying misalignments and conceiving aligned solutions.

Alignment of the Company

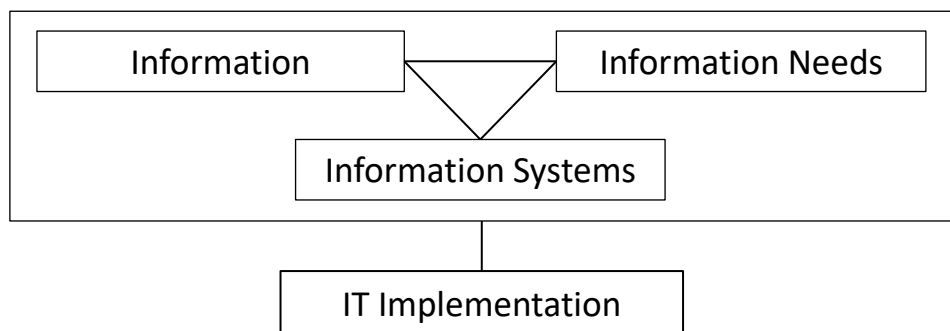


Alignment of Plans

Plan Architecture - System of Plans – Organised and managed Set of Plans – Hierarchical Network of Plans



Alignment of information and Information Needs



Notes:

Alignment is a momentary state. Because of variability, everything has to be kept in sync (aligned). They have to evolve together. (different speeds of adaptation!).

Alignment is a reactive process. It's a minimal state companies need to drive, to lead, to create, to innovate.



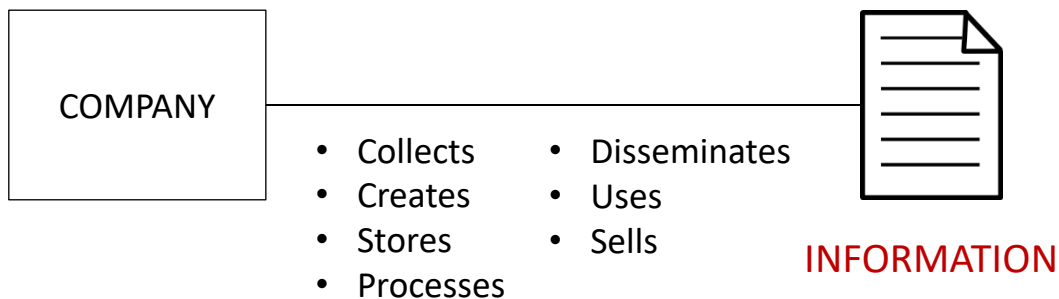
INFORMATION EXPLOITATION

Information is meaningful (valuable) to

- people
- companies and organisations

Information is a **resources**. It must be ...

- Available → captured and stored
- Fit for usage → processable & processed
- Accessible
- Stored
- Organised
- Managed → manageable
- Exploited → exploitable, valuable, reliable
- Secured



Information is captured, created, processed, stored, disseminated and used by

1. **People**
2. **Machines**
3. **Software/computer systems**

Information is a Critical Resource

30/10/2018

Page to be reviewed

Information is **power**. Information has **value**.

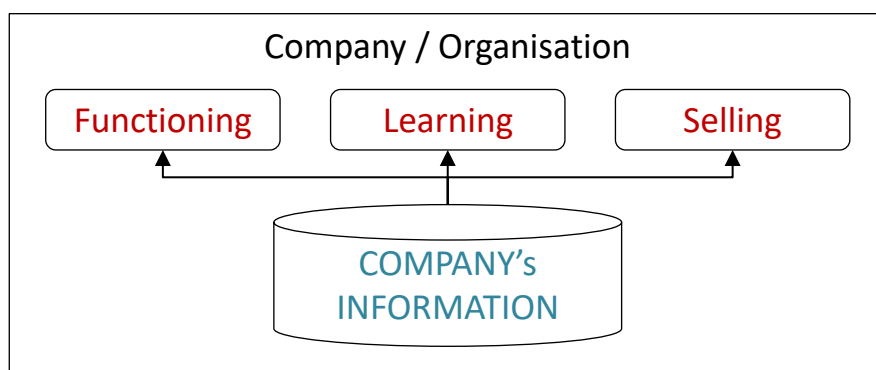
Information is key in decision making, choices, inventions, ...

A company **without** information **instantaneously stops functioning**.

Information

- supports the **functioning** of the company
- supports the **management** of the company
- as **product** or **service**
- is knowledge / **understanding**
- is key to **innovation**

3 Ways to exploit information



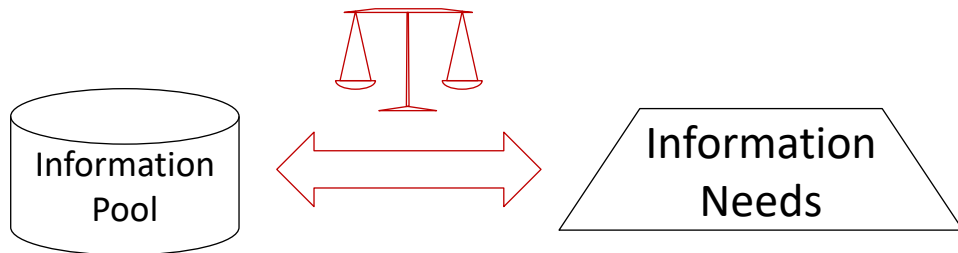
How to maximally exploit Information?

Information must be taken very seriously

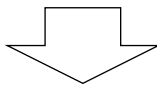
Information Exploitation

Page to be reviewed

Value of Information is linked to the Needs and Usage

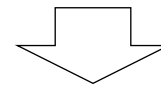


BALANCING INFORMATION WITH INFORMATION "NEEDS"



Maximising

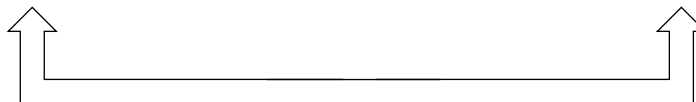
- the **Quantity** (of relevant information)
- the **Value** of Information
- the **Exploitability** of Information



Maximise the Usage

- **Needs**
 - Detected - Hidden ignored needs
 - Satisfied - Unsatisfied
- **Opportunity**

"What can be done with an information that is valuable?"



*Matching
Create correspondence*

Note:

Information has also potential value. It can be needed later. The amount of information present is larger than what is necessary to satisfy the present needs.

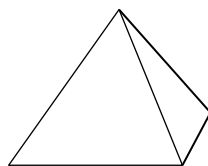
Information Needs

Needs-side of the Balance

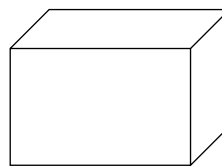
Sources of needs



Environment
with people,
systems, ... like
company, society

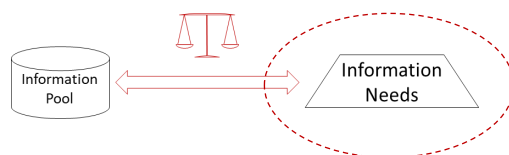


Organisation



System

*These sources
have to be
investigated*



	people are aware of / know	people are not aware of/ ignore / underestimate
Needs		
Improvements		
Opportunities		

Easy, at the surface,
visible, small, local,
obvious, concrete,
experienced, ...

Easy to detect

Much harder to detect ..

- *Can be important*
- *Can make the difference (innovation)*

- Hidden
- **Requires deeper understanding, special skills, attention and effort**

*Often we aware something is wrong,
but we don't really know wat, we can't
pinpoint it or indicate it wrongly.*

for informatics experts

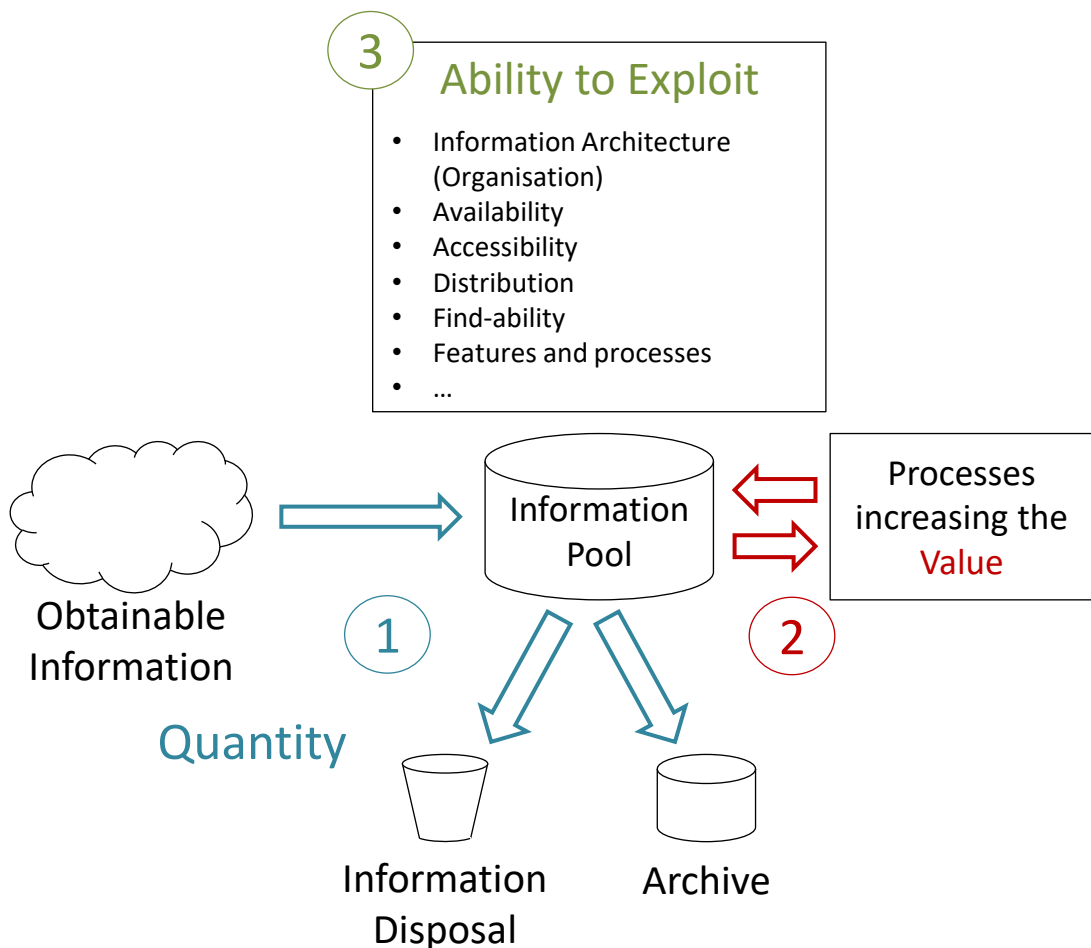
Information Exploitation

Information-side of the Balance



1. Increase **Volume** of useful information
2. Increase **Value** of information
3. Possibility & **Ability** to **Exploit** the Information

HOW?



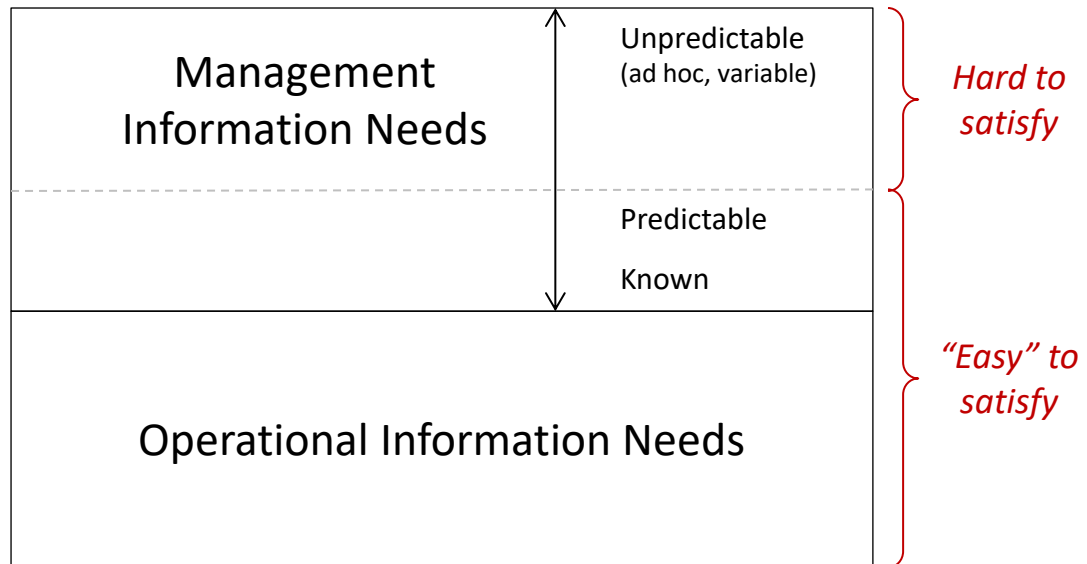
More about this on next pages

A) Need's Perspective

- **Company's** Information Need
 - Required for the company to function
 - Independent of what people wants, desire or decide
 - Ability to respond to usual information needs and to more unpredictable information needs
 - If considered broadly, rather stable, may change over longer period or a change is announced (decisions, plans, ...)
- **People** Information Need
 - Ad hoc needs:
 - required by people to reach one-off goals
 - to perform, unique tasks
 - to increase their insight
 - Lesser predictable

B) Value Perspective

- Some information, through its meaning, has intrinsic value
 - Usage or potential usage
 - Curiosity & Interest
- Value is subjective and has to be estimated: criticality, likelihood, potential purpose, opportunity, value of decisions or actions made possible by the information
- Possibility to learn:
 - about the organisation, its environment, its functioning, its results necessary for management, improvement, innovation, ...
- Possibility to sell as product or service



- If the focus is only on responding to operational needs, it will be hard to respond to management information needs.
- If the focus is expanded to the known, evident or very predictable management information needs, then only these questions can be answered.
- If the information is considered more broadly, more of the unpredictable information needs can be answered. This allows to increase the insight of the management in the company and thus its control over the company.

- Information needs can **only** be **solved by** having the **information** when needed and where needed.
 - Information will only fulfil needs **if it has certain qualities** (identifiable, linked (relations), reliable, complete, precise, consistent, up-to-date, accuracy, accessible, process-able, timely, ...)
- **User dependency** (needs depend of **user and user group** (function, service, department, ...))
- **Situation dependence** (needs may vary according to the situation)
- **Time** dependent (needs may vary over time)
- **Stability**
 - Information needs of the company → more permanent → more stable
 - Information needs of people → more temporary → more changes
- Some needs remain **undetected** or **badly understood** (not always easy to detect and understand real needs – symptoms of information issues)

ANALYST REQUIRED !!

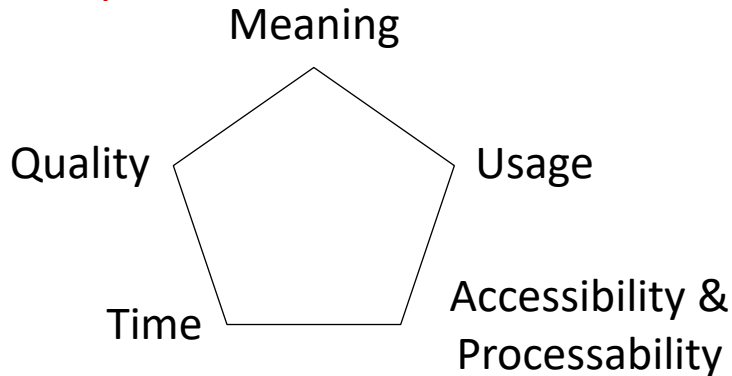
Notes

Important slide! Should actually be one of the first in the course.

It's all about "information needs", not about "technology needs"!!

Technology are only means to be better able to cope with information needs.

Value determined by



Value of information is subjective. It's an estimation. It depends of the individual, circumstances, time (variable in time), place, ...

Meaning

- What does it mean?
- Meaning is increased when the information is **connected** to other information.

Quality

- Reliability, clarity, unambiguity, precision, accuracy, completeness, up-to-date, recent, format, ...

Time

- Information has to be available on time or must pertain to meaningful period

Accessibility & Processability

- Inaccessible information is worthless
- Difficulty to process decreases the benefits of information (processing costs time and resources)
- In time, on the right place, in the right form increases the value (or doesn't decrease it).

Usage & Possible Usage

- What **is** it used for and what **can** it be used for?
- Is its usage essential, critical or optional?

**Consider the maximal (or max. potential)
value of information**

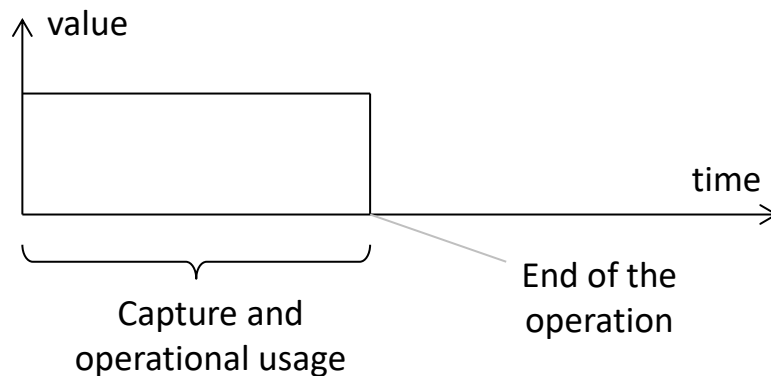
Practical value: Value by practical usage of information. Responding to an existing information need or to a predictable need that will occur in the short term.

Example: The address of the client where the product will be delivered within a week.

Estimated potential value: Subjective value assigned to information based on possible usages in the future.

Example: Names and addresses of former customers. They are still useful. Today, they have no practical value (no ongoing order). But these clients can put a new order in the future and this information can be useful for future mailings or marketing initiatives.

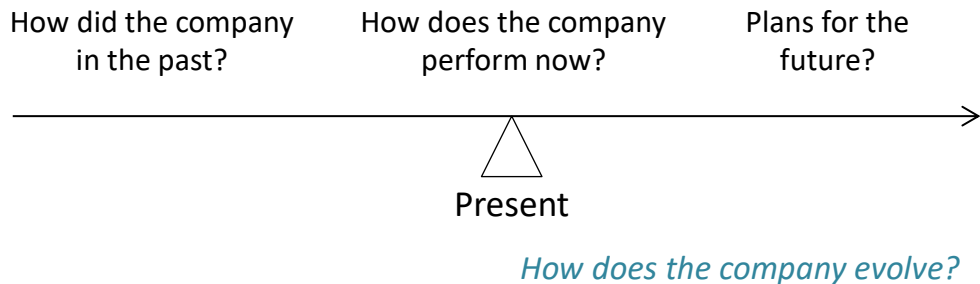
Operational value of an information element



Usage is known. Relatively predictable.

Example: A customer order has operational value from the moment the order is placed until it is completed. Once completed the operational value falls back to nearly zero.

Management value of an information element

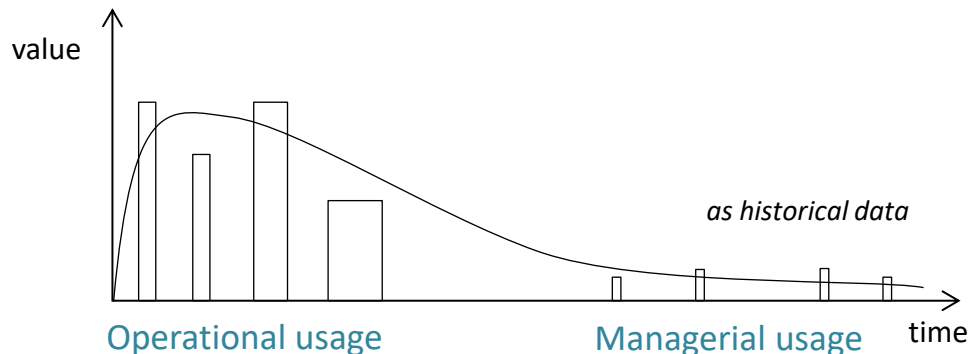


Predictable

Habitual information about the functioning of the company (example: KPI's)

Unpredictable as a response to ad hoc investigations

- Detection of issues or opportunities
- Investigations
- Simulations



Management needs information, among others, to plan future operations, to monitor performances, to look for tendencies.

Operational data is often used as a selection. Aggregated data or lesser detailed data are used for management purposes. The value of historical data tend to decrease over time.

At some moments in time an information element may regain in value. The value curve is not as smooth, but rather irregular.

Information Value is determined by following aspects:

1. Meaning - Relevance - Usefulness - Interest

Information can be interesting but not useful. Or it may be useful for decisions and actions.

2. Truthfulness - Up-to-date

Degree of truthfulness ranging from false to true.

3. Trustworthiness – Reliability

User's of information must be able to rely on information. For example, right information mixed with false information will undermine the trustworthiness of the right information.

4. Accuracy - Vagueness - Ambiguity - Coherency - Precision - Detailed - Clarity

Information has to be accurate enough for its intended and possible usage. Obtaining information that is 100% accurate might be (too) costly. Information that is not accurate enough may decrease its usability and its value. Degree from general or vague to precise.

5. Completeness

Information needs to be complete. Some operations, processes or decisions are more difficult, more risky or can not take place if some information is missing.

6. Context

Information without knowing and understanding the context my not be much worth.

7. Atomicity - Structured

Elementary, atomic pieces of information are generally more worth than aggregated information.

8. Probability - Risk - Likelihood - Certainty

In general, the greater the probability, the more usable the information is. Lesser probable information can be very valuable as well.

9. Variability

Information can be true on one moment and lose its validity later in time.

10. Timeliness

Information is often related to time in 2 ways: It relates to a moment or period of time. It is useable at a certain moment or period of time and thus must then be accessible to the user. The information has to be presented to the employees on time. Information coming too late (too soon) is of no or lesser use.

11. Availability

Information that is inexistent can't be used. It must be recorded. It is worthwhile to look at the clients and to search in the company's environment for interesting information.

12. Accessibility

Information that is inaccessible is useless. Accessibility can be a matter of geographical localisation.

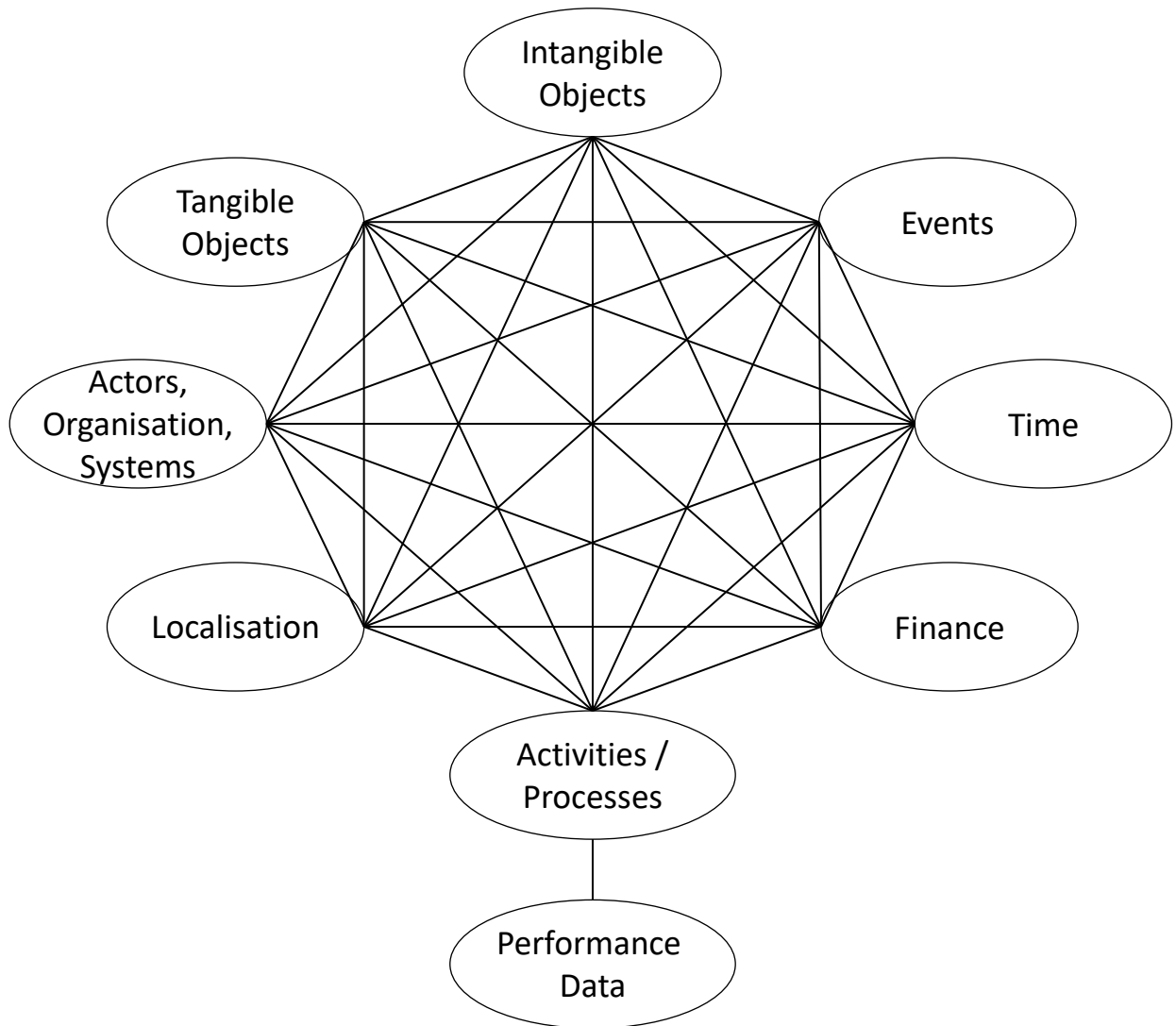
13. Awareness

If users ignore the presence and availability of information, they are not likely to use it. Ensuring the existence of information is known is important.

14. Processability – Appropriateness for Processing

Atomicity, organisation, integration, standardisation, format, encoding and many other factors influence the processability of the information.

Basic model of Operational Information

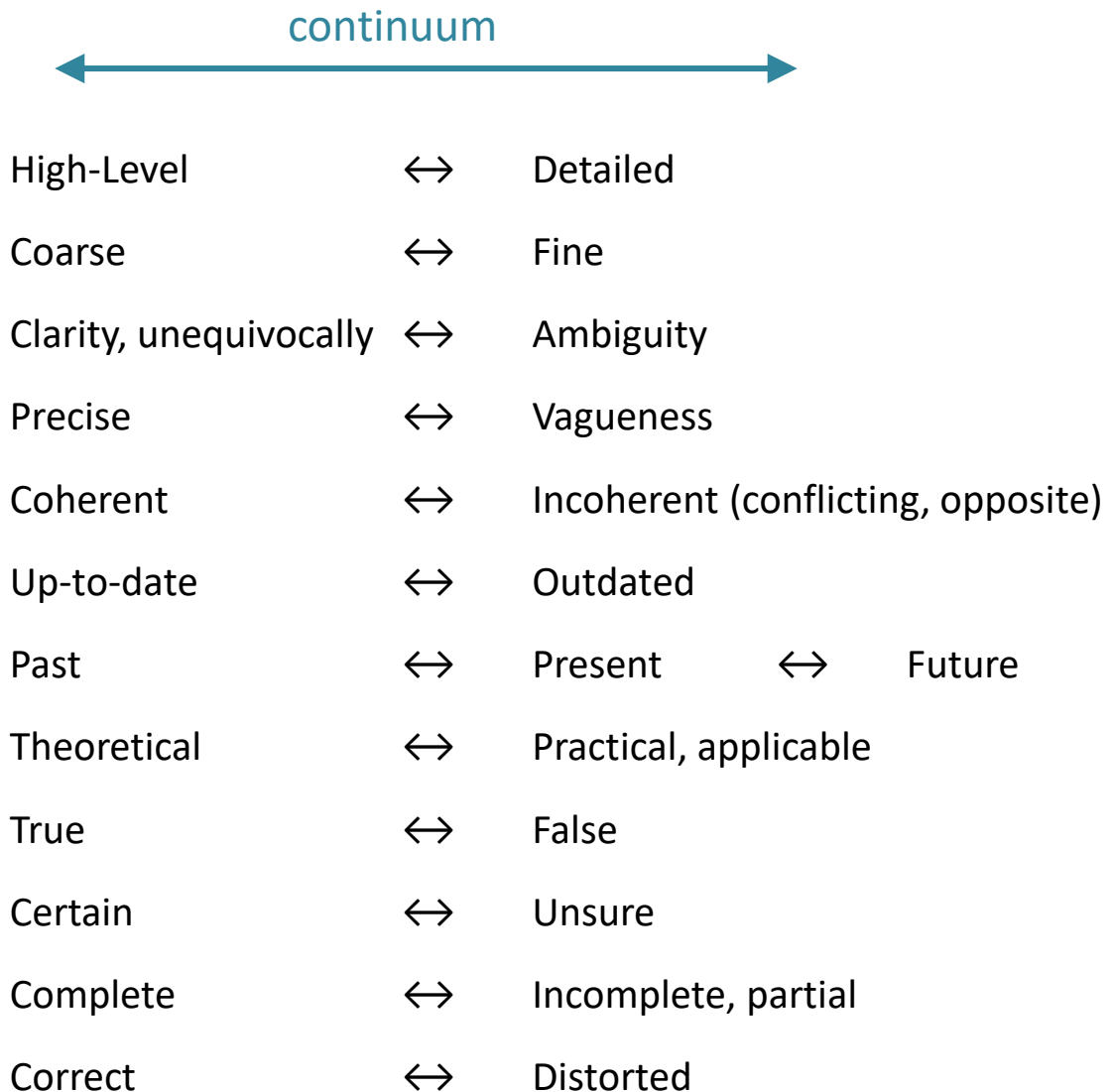


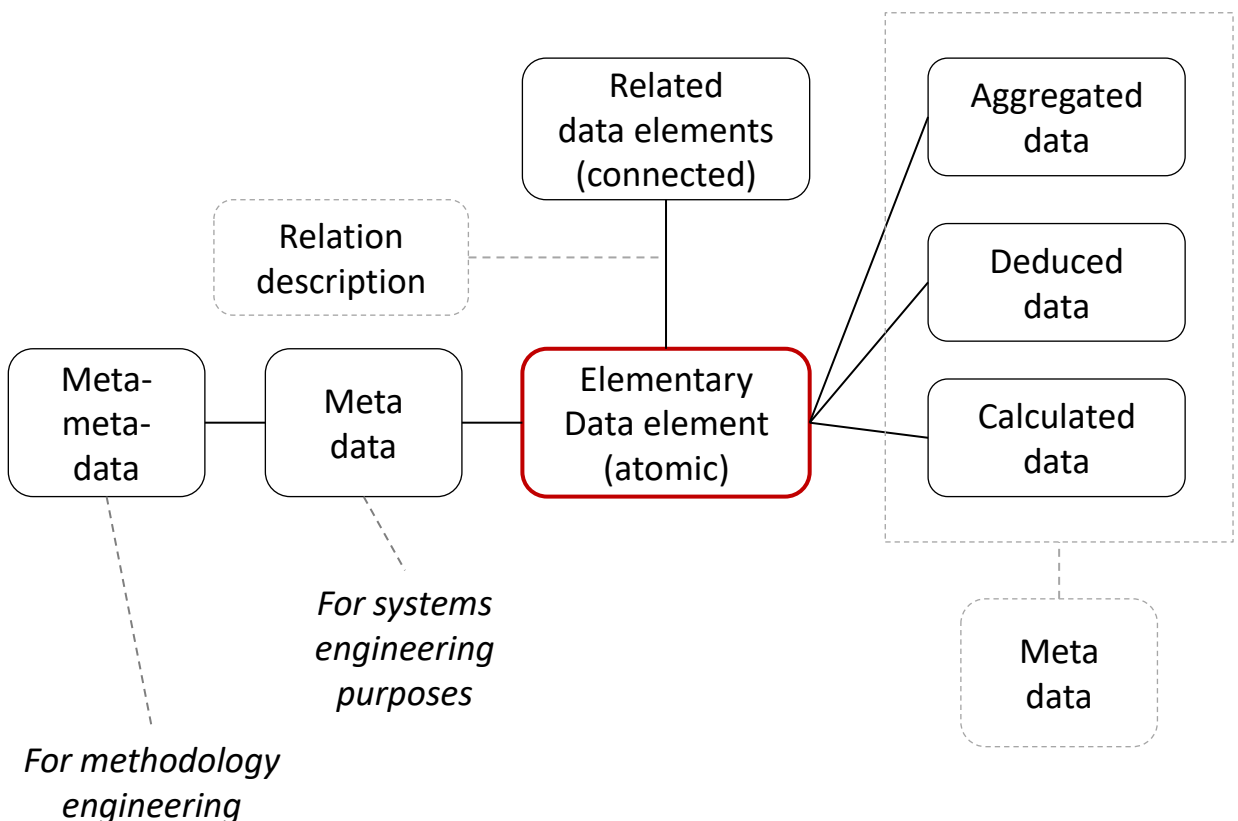
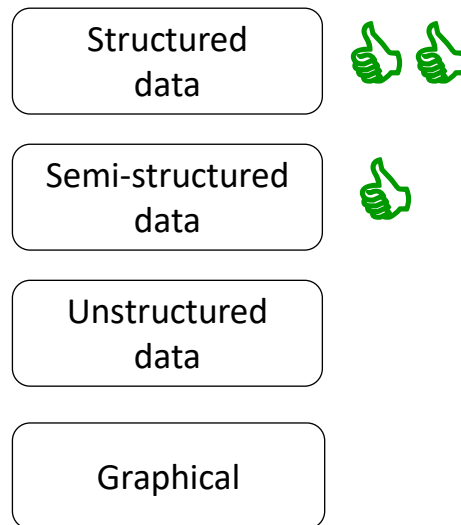
This is a basic model that may help when investigating operational issues. It may help to come up with some questions. There might be lesser or more entities involved.

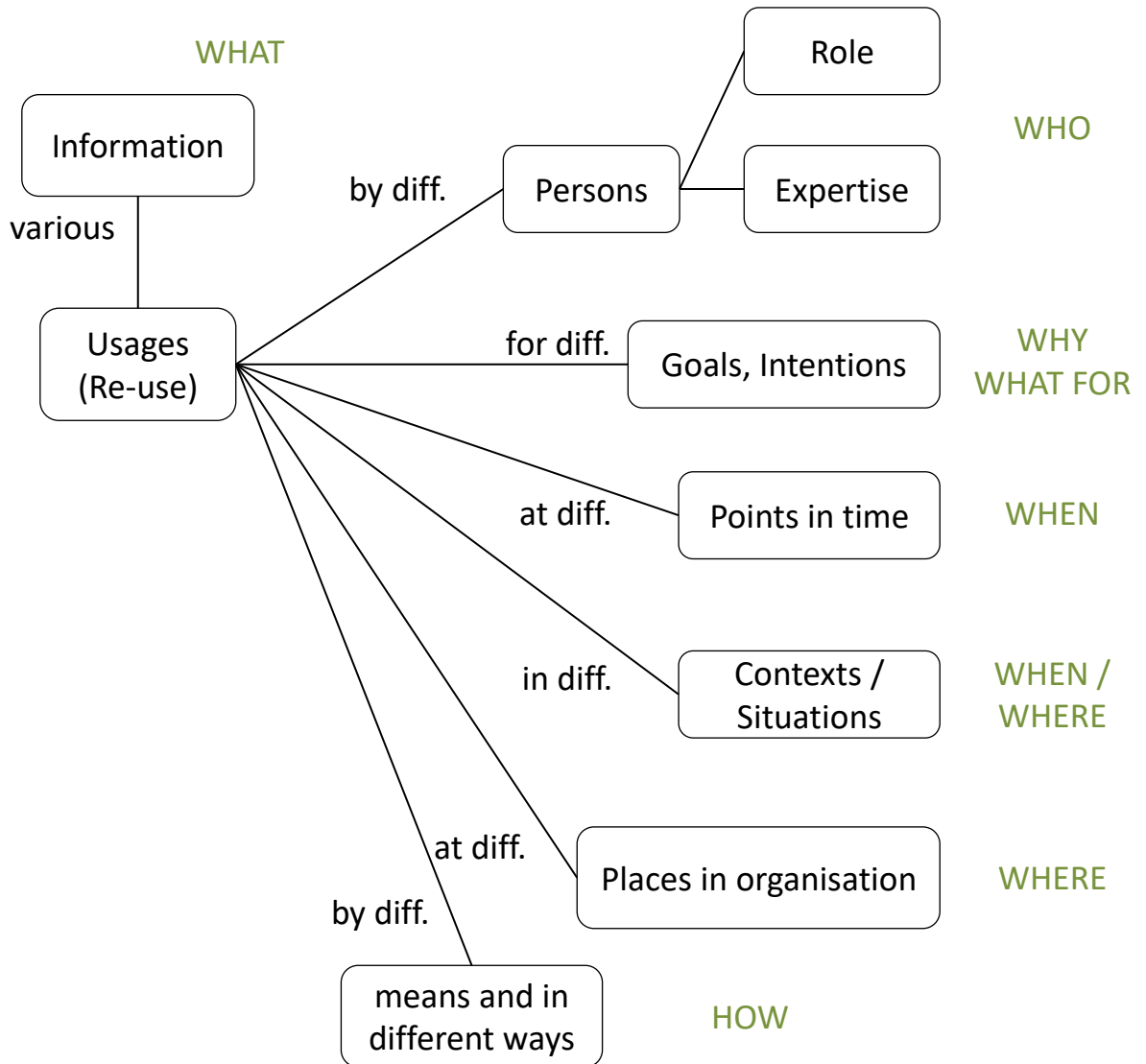
A right classification of these elements is crucial for later use by management.

Degrees of Information Quality

30/10/2018







Prepare information for multiple usages

Design information solution (information architecture/model)
that allow multiple usages of information

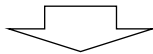
Operations

DESTROYING

the **value** and **exploitability**
of information

- Adding
- Modifying
- Usage of freedom and flexibility to manipulate data
- Leaving it as is
- Aggregating
- Unstructured information
- Disconnecting
- Transferring
- Duplicating
- Unconsidered dissemination

(End-users can do this)



- Inefficiencies
- Loss of meaning
- Loss of value of information
- Loss of reliability
- Difficulty to process
- Chaos
- Information Swamp/ Overload

EASY
Operations

Operations

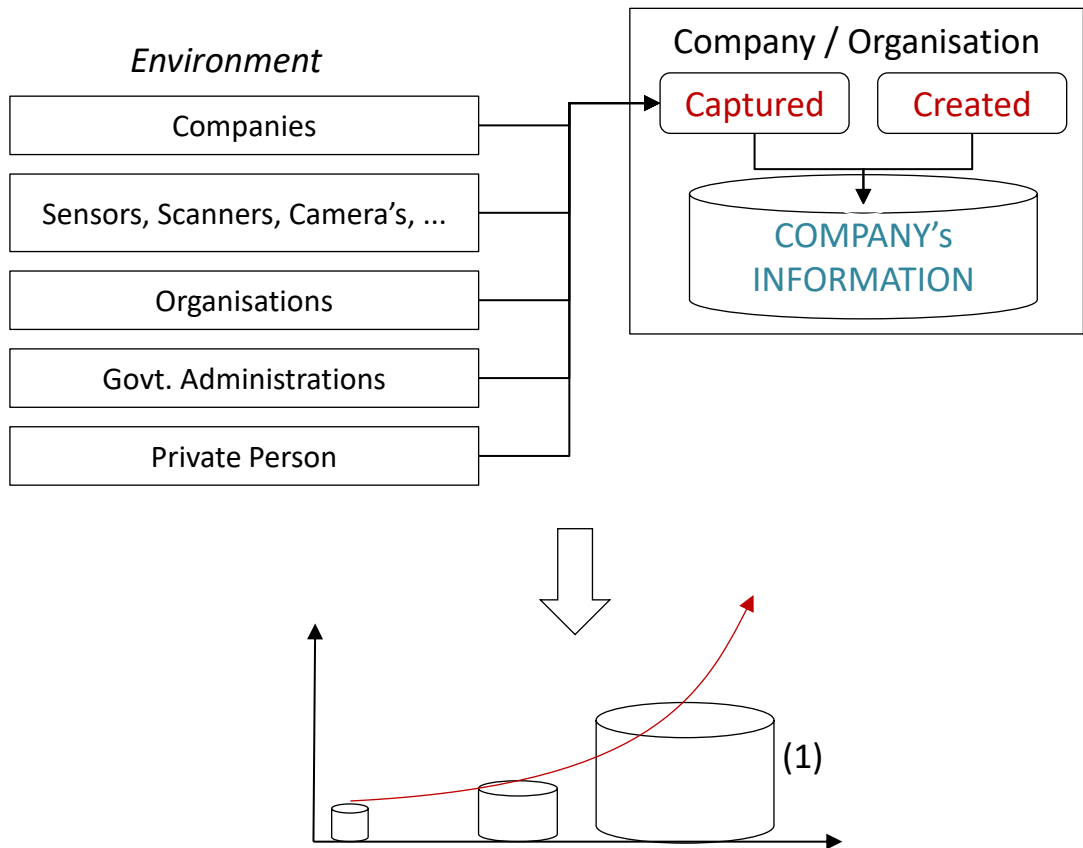
INCREASING

the **value** and **exploitability**

- Identifying
- Meta-data (creating, describing)
- Decomposing
- Analysing
- Selecting
- Standardising
- Cleaning
- Modelling, structuring, organising, formalising
- Connecting
- Keeping up-to-date & reliable
- Maintaining consistency
- Managing information status
- Managing information throughout its life-cycle
- Preserving and maximising the value of information
- Maximising the processability

CHALLENGING
Processes

2 Ways to increase information



Capturing information is crucial

Missing information is a cost, a lost opportunity.
It may have far reaching consequences.

REFLECTIONS

(1) Endless growth of information? Good or bad? To be controlled? To be managed? How to deal with it? How to adapt systems to it?

Information tends to increase over time + **Information Entropy**

IF

- ▶ the nature of information is not understood

AND

- ▶ the principles of information engineering and information management (*including proper information usage*) are not respected

THEN

it's easy to create **INFORMATION OVERLOAD**
and **INFORMATION CHAOS**

Losing control over information

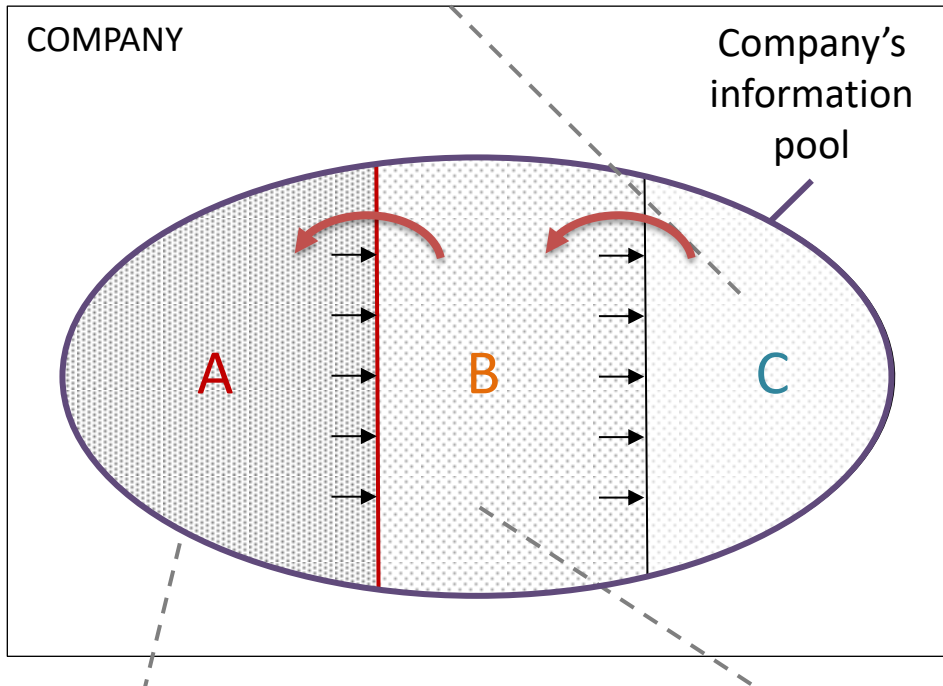
=

Losing control over the company

Over time, quantity tend to increase
whilst quality tend to decrease

UNREGULATABLE INFORMATION AREA

End-users decide freely about what information they capture, organise, process and use and how they do it; information is not formally controlled; not standardised; not inventoried; not quality checked; hard to process; maybe copied but not shared; ... on the edge of chaos.



ENGINEERED INFORMATION AREA

organised, formalised, computerised information and information processing

CANDIDATE INFORMATION FOR ENGINEERING

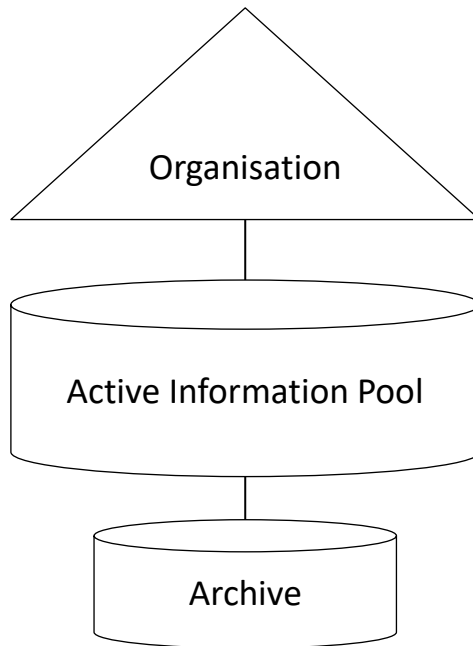
Information and information processing that can be engineered in the future

Notes:

Expand the *Engineered Information Area* (A) by engineering area B.

Expand the *Information Candidate for Engineering Area* (B) by detecting information and processes in area C that might be engineered. Overtime, area C will shrink.

Area C is difficult to formalise, to clean up, to maintain, to manage, to exploit, to engineer, etc., at least, for now. (OAS supports the exploitation of 'B' and 'C'-type information).



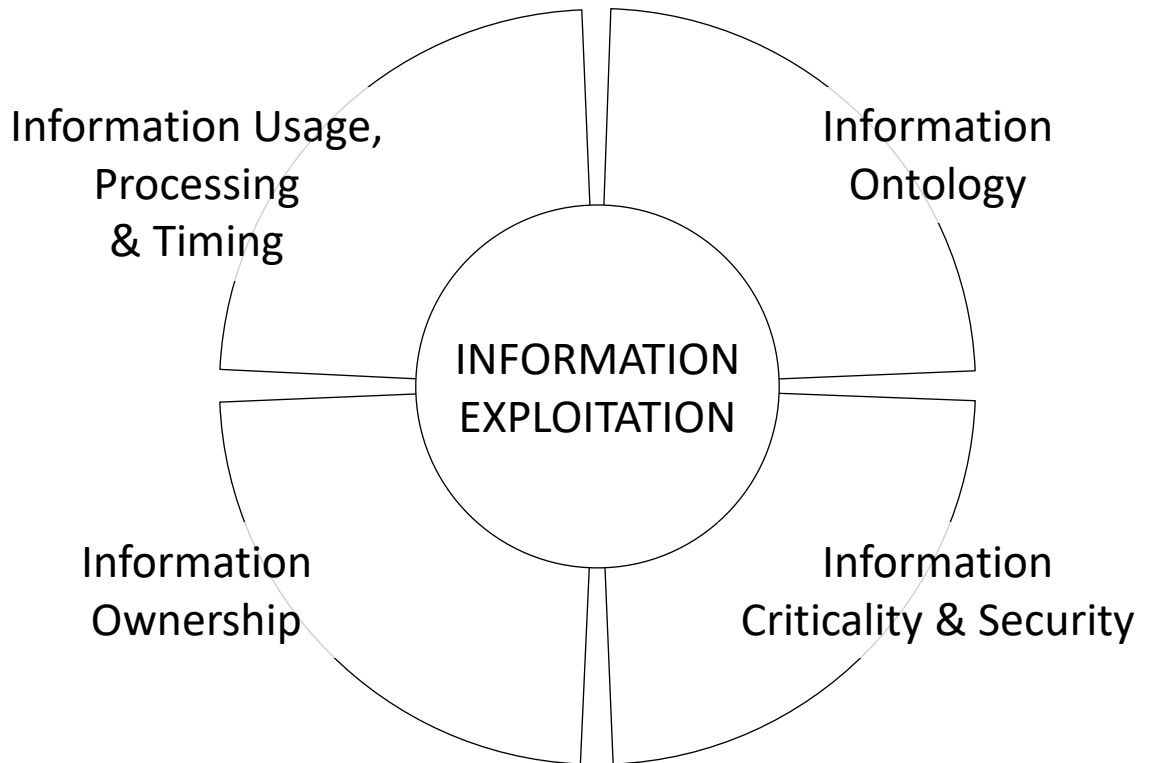
INFORMATION POOL

- Respect essential information qualities (below)
- Only valuable information
- Organise information
- Avoid information duplication
- Maximise information and information quality
- Maximise information value
- No useless information (remove, clean up, archive)

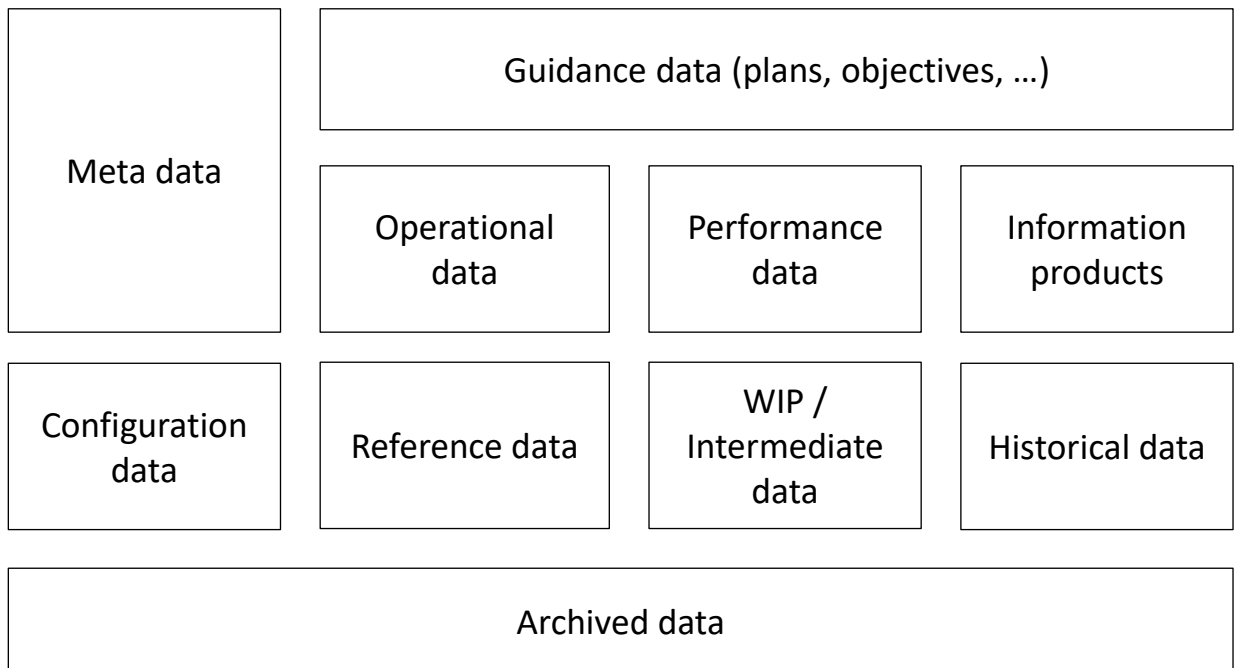
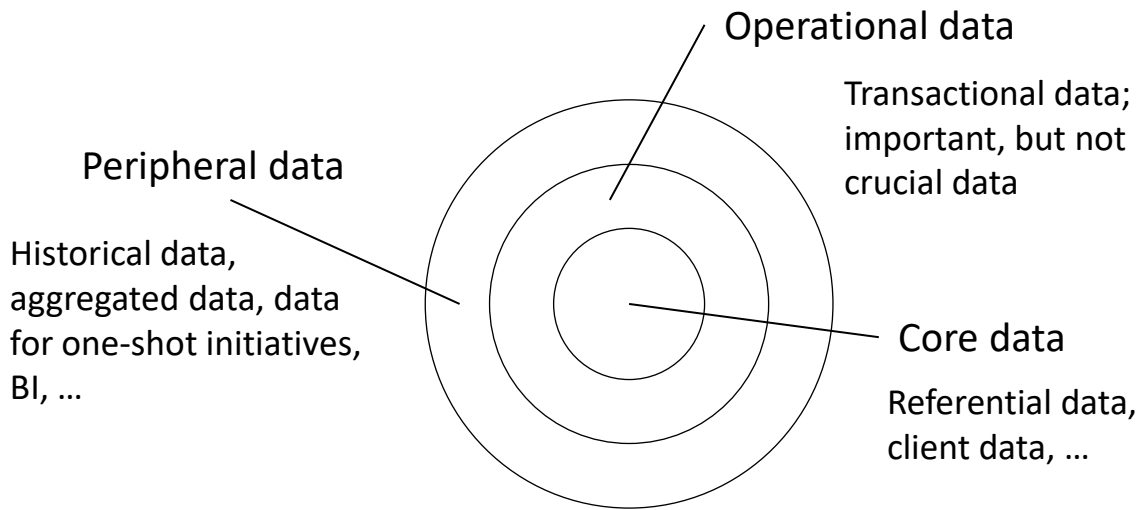
Information architecture brings stability. It is better to develop processes upon good information architecture than to develop an information architecture that suits the present processes.

Why better? The information structure is better which results in better exploitability, expandability and changes will have smaller impact.

Four Perspectives to Organise and Exploit Information



These perspectives have to be combined intelligently.



“information”

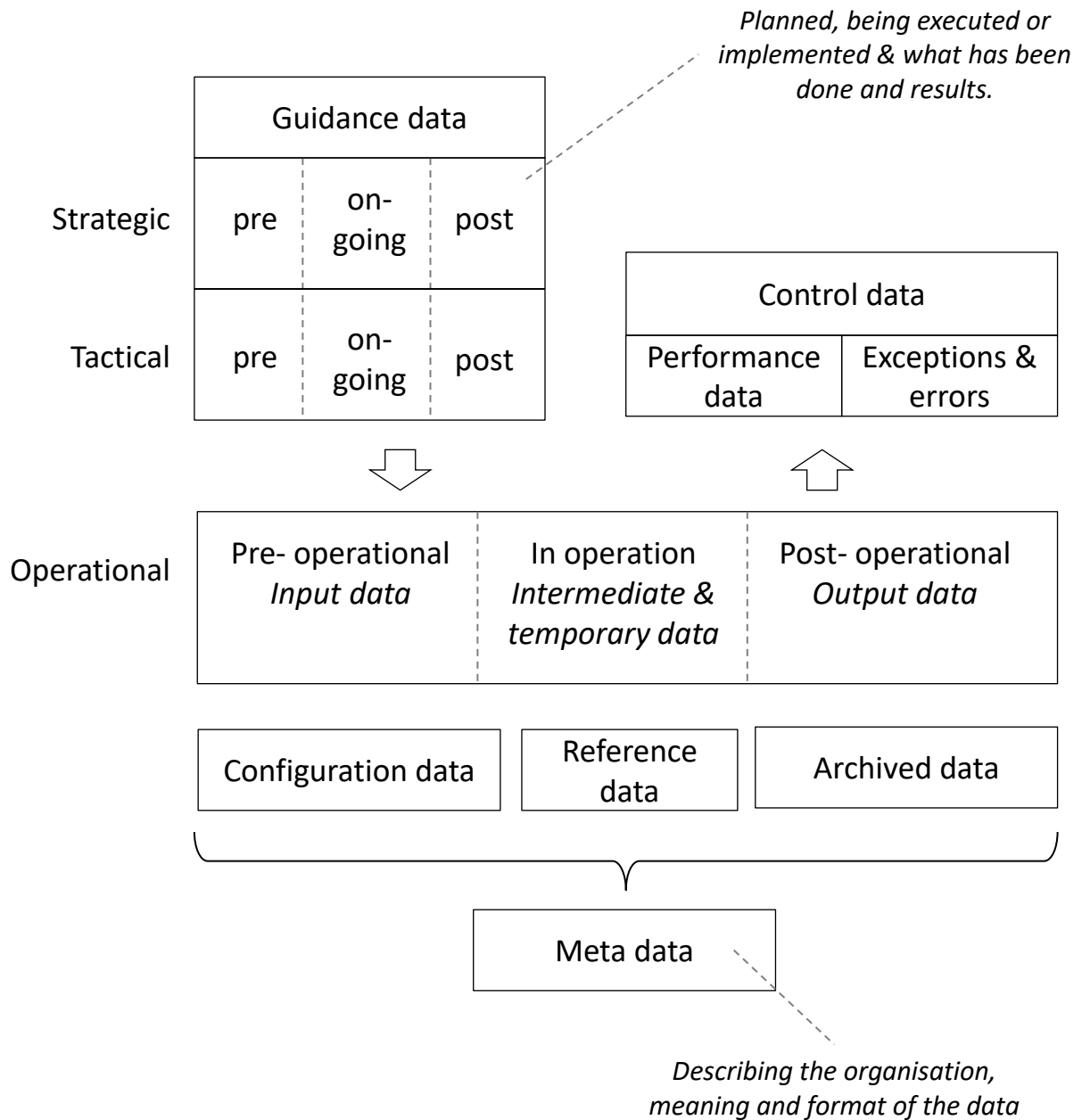
when emphasising the purpose, the usage and the value of data

“data”

when considering information as a matter that has to be captured, processed, stored, ...

Information Organisation

30/10/2018



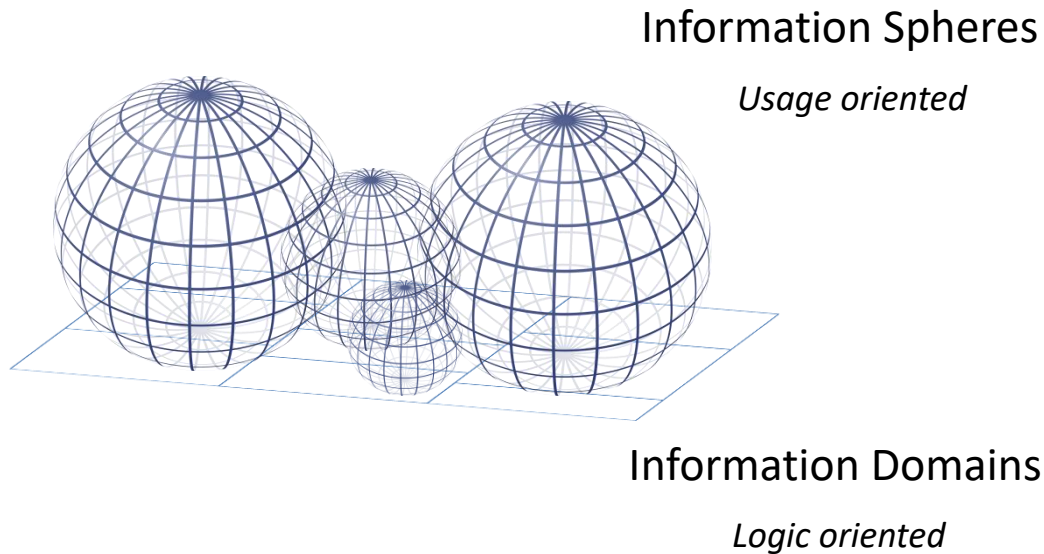
Methods to Organise information

30/10/2018

- Per Concentric layers
 - Core: reference data
 - Operational: transactions, process
 - Peripheral: Managerial info, BI
- Per Business Domain
- Per activity, activity type or purpose
- Per nature
- Per availability

Usually, a mix of the perspectives is used

Information Domain and Information Sphere



Information Sphere:

Whole set of information a user or users group uses, controls or might need to perform a job

Information Domain:

Information relating to a same subject, object, topic, issue, aspect, item (reflects the real world)

Notes

Business people may need to use information from different domains (like products, orders, financial,...) to perform their job.

These two concepts help to manage accesses and usage of information.

A user is inside a sphere together with all available and to him/her accessible information doing his work.

How to maximise the value and exploitability of information ?

1. Relevant, meaningful
2. Correct
3. Up-to-date
4. Precise, Clear, Unambiguous
5. Clean
6. Reliable (verified)
7. Standardised
8. Connected
9. Atomic (elementary, indivisible)
10. Logically organised in an architecture (reflecting real world, criticality, purpose, processing or usage)
11. Structured (avoiding aggregations, documents, ...)
12. At the right place
13. On time
14. Available, being present → need to be captured, gathered, ...
15. Accessible
16. Findable and retrievable
17. Quickly accessible
18. Easy to process
19. Meta-data (described with)
20. Captured, stored
21. Unique occurrences
avoid having to update of multiple occurrences of same information
22. Shared and Re-used
23. Protected
24. Inventoried, Maintained and managed
25. Under control, under own control

Goal of the Analyst
*striving to maximise these
qualities in corporate
information solutions.*

*These aspects have to be taken into
account when conceiving
information architecture,
information systems and
information treatment and usage*

How to improve Information Exploitability

1. Capture as much relevant data, data with a true potential value, as possible
2. Use an Information Inventory
3. Keep control over the information
4. Keep information as atomic and structured as possible
5. Connect information elements as much as possible. It increases their value. A connection is also information.
6. Share information (avoid multiple instances. Though, in some cases multiple instances is advisable)
7. Make information accessible (but secure it as well)
8. Understand the meaning of the information (what precisely does it describe)
9. Use information formats which are easy to process
10. Avoid cryptic and confusing codes
11. Standardise information as much as possible. Limit the number of formats
12. Implement agents controlling data quality (respect of formats, missing information, incoherent information, ...)
13. Clean up (delete, archive, ...) old data.
14. Have data policies and data governance
- 15. Train People**

Notes:

Some items are on the previous list as well.

Main Causes of Information Problems

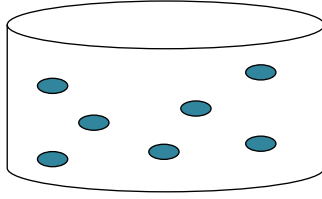
30/10/2018

- Lack of information
- Bad quality of information
- Unorganised or badly structured information
- Inappropriate “solutions”
- Bad information practices

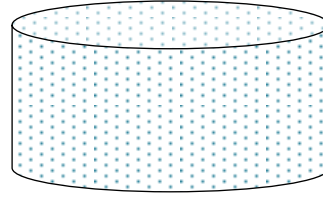
Some wrong, imprecise or unreliable information in a database can destroy the trust in all the information in that database and thus destroying the value of the database.

Trust & Information Contamination

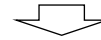
15/03/2019



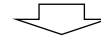
A few unreliable data,
other data is OK.



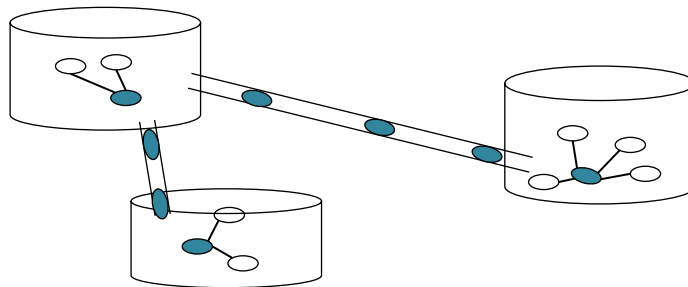
Since we don't know what data is
reliable and what isn't ...



we suspect every data element to
be unreliable.



The whole database is considered
as unreliable & **not trustable** and
not trusted.



- Contaminated data is **transferred** to other databases
- Data is **connected** to other data
- Data is **used** for different purposes, by different persons

Notes:

How to limit this? Automated data input; appropriate and facilitated data input; appropriate checks on input; Data Quality Checking software running through the DB; get employees verify data when using it (if possible)

- What information has **value**?
- How to **capture** information (easily, efficiently, early, reliably, ...) ?
- How to **identify** information?
- How to **verify** information?
- How to **structure** information?
- How to **link** information (relations)?
- How to keep information **process-able**?
- How to keep information **up-to-date**?
- How to keep information **consistent** throughout the company?
- How to keep the information **useable**?
- How to **evaluate** information?
- How to **share** and **reuse** information?
- How to **control the usage** of information?
- What is the **lifecycle** of information?
- What **meta-data** should describe information? How to determine and manage meta-data?
- How to **preserve or increase the value** of information?
- How to **secure** information?
- How to **innovate** with information?
- How to **clean up** information?
- What information can be **erased**?
- ...

We need to develop the practices that responds to these questions.

Information is a

- 1) Critical resource
- 2) Extremely valuable
- 3) Hard to manage

Think “INFORMATION”

- Be vigilant about any “information destroyers” and “information risks”
- Use your best information experts
- Sharpen information skills
- Avoid giving away control over (critical) information



ENTERPRISE ARCHITECTURE

Architectures

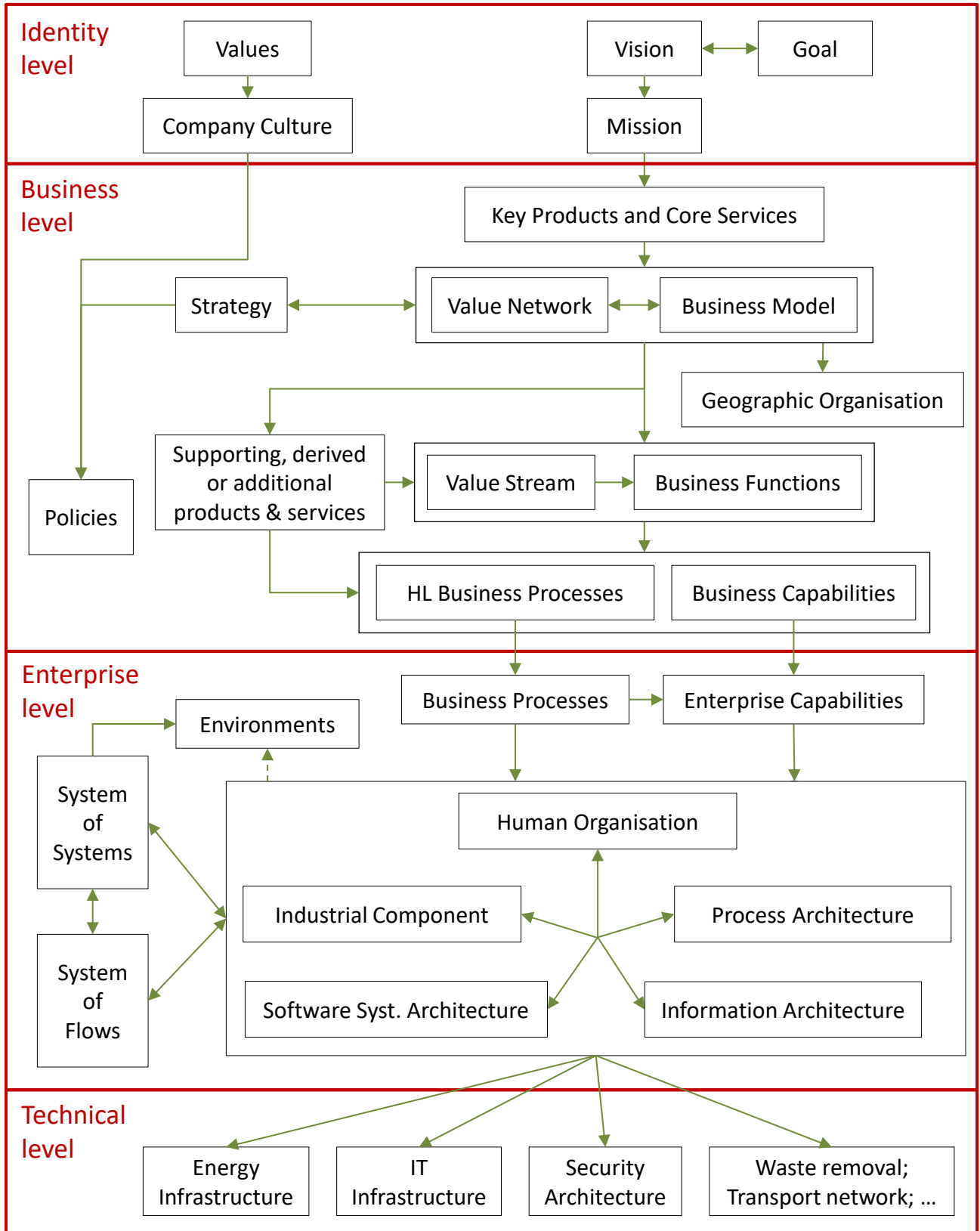
Some Major Structures / Architectures

- Goal Structure (Architecture, Model)
- System of Plan (organised and managed set of all/major plans)
- Decisions Hierarchy (?)
- Capability Structure
- Geographical organisation
- Organisation
- Process Architecture
- Information Architecture
- Production / Assembly Line Architecture
- Information Flows
- Communication Flows
- Transportation Infrastructure
- System of Information Systems
- System of Software Systems
- Functional Software Architecture (FSA)
- IT Architecture
- Security Architecture
- Network Architecture
- Energy Network
- Water Network
- Waste Network

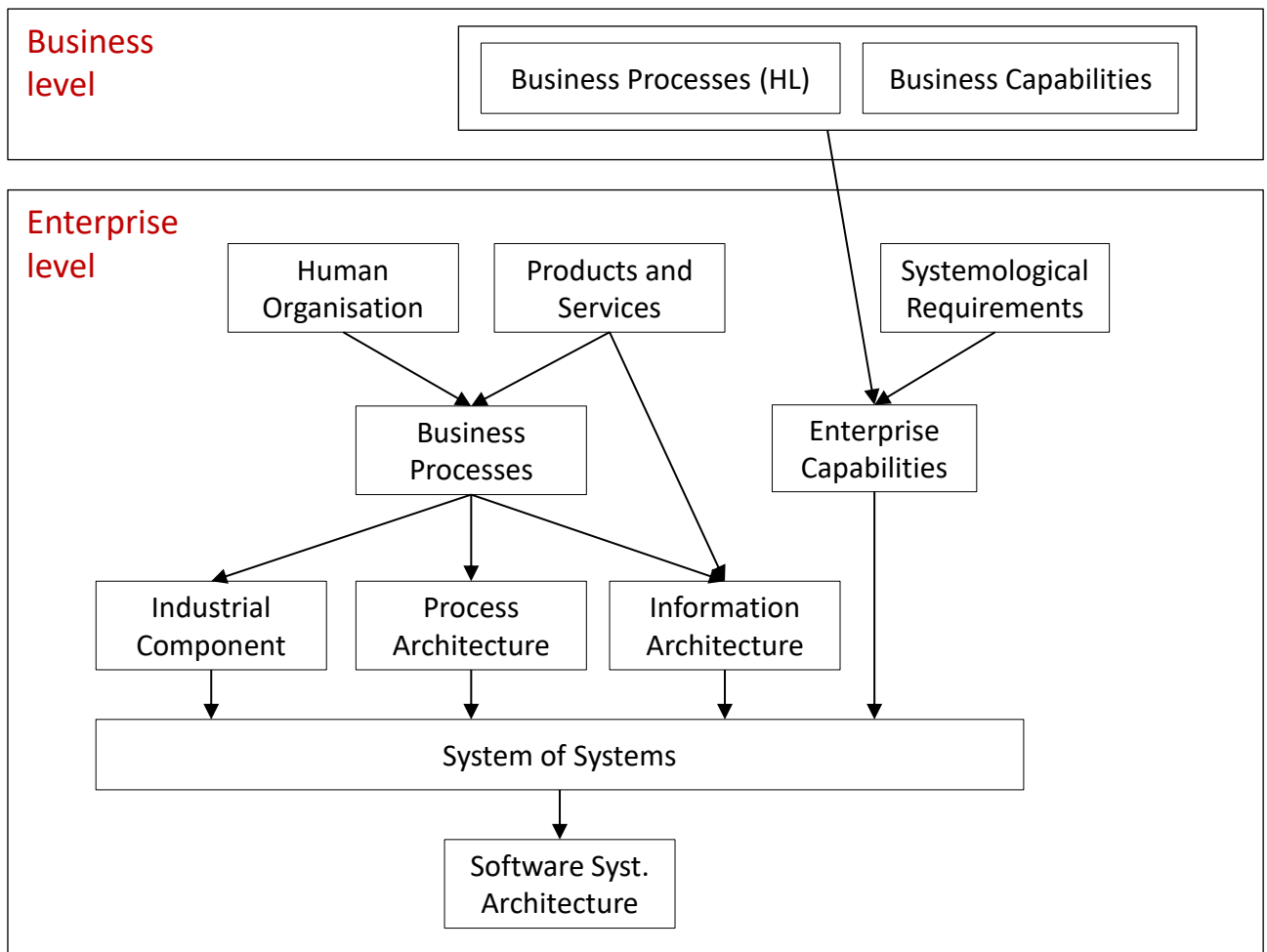
Notes:

- This set of organisations, structures, architectures and systems gives an idea how things are entangled. It emphasises a holistic approach.
- Each architecture requires specific knowledge area specialisation and skills. Yet, they need to be aligned. All these engineers have to collaborate very closely.
- Some architectures may be not applicable to an organisation.
- There is more required to achieve alignment, balance and sustainability than aligning architectures.

Enterprise Engineering Framework



Adapted extract from the previous diagram:



Notes

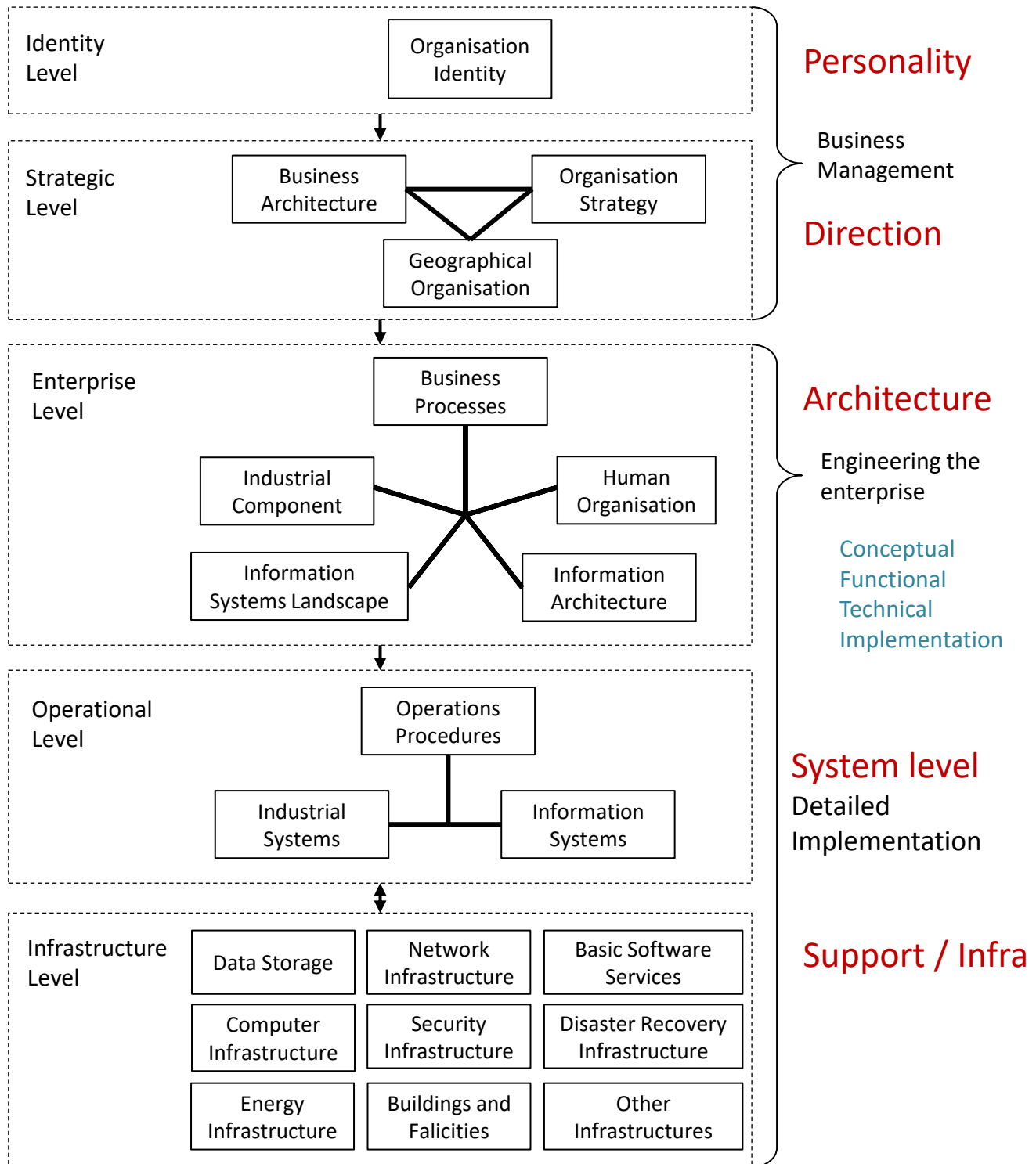
These diagrams present some important elements of a company and from where these elements can be derived. A model is always a reduced picture of the reality. And since every company and situation are different, the relations between these elements are specific to each case. They vary from case to case.

For example, the human organisation can be existing and be a given for an initiative.

It is the duty of practitioners to analyse and to understand the company and situation they deal with. These diagrams can serve as a basic template helping to understand and which then can be adapted to the particular situation.

Another Variant

30/10/2018



The **Business Architecture** is a business description of the system by which business activities are deployed, the way of executing the business activities and the way it will achieve the realisation of its vision, mission, purpose and objectives. The Business Architecture is a business perspective.

Principle: The business architecture has to be defined by

- Management
- Business Experts
- Business Consultants

Consequently, anything requiring non-business expertise to be established, doesn't belong in the Business Architecture. It is a part of another Architecture.

Content of the Business Architecture

- Actors, Entities, Organisations
- Partners
- Suppliers
- Main Resources
- Sources of resources
- Key Products
- Core Services
- Key Business Capabilities
- Geographic Locations
- Geographic areas
- Customer Segments
- Interorganisational Flows
- Value Streams
- Business Functions
- Business Policies, Rules, Principles
- High Level Business Processes
- Financial Structure

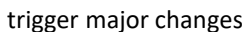
Complementary to the Business Architecture

- Strategy
- Metrics and Measures
- Initiatives and Projects
- Decisions

Notes:

This definition of business architecture differ slightly from the mainstream definition.

30/10/2018

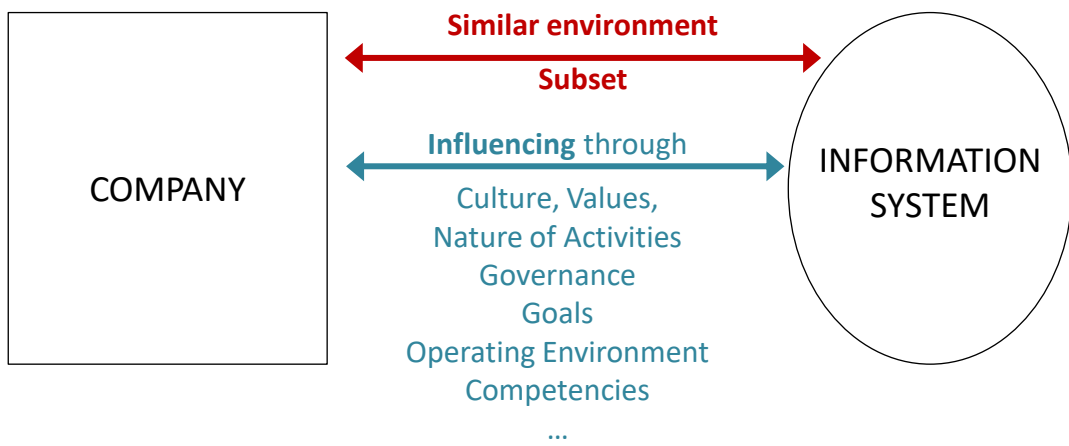
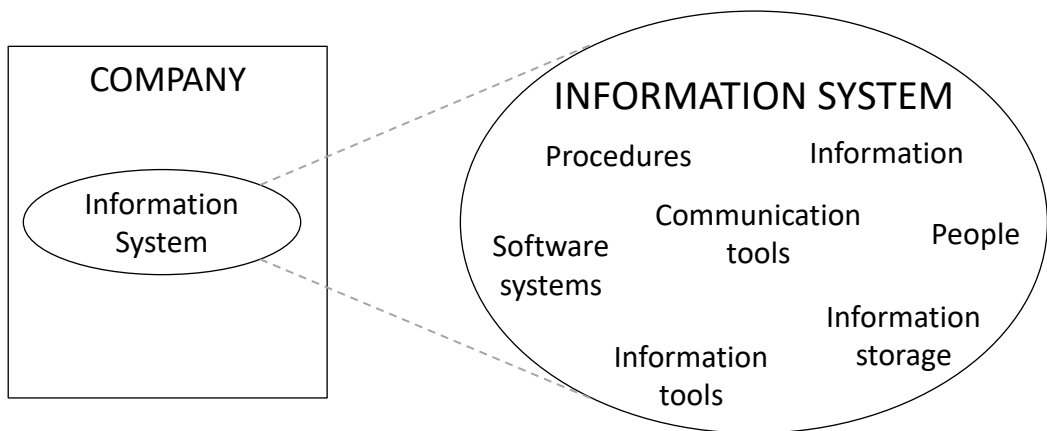
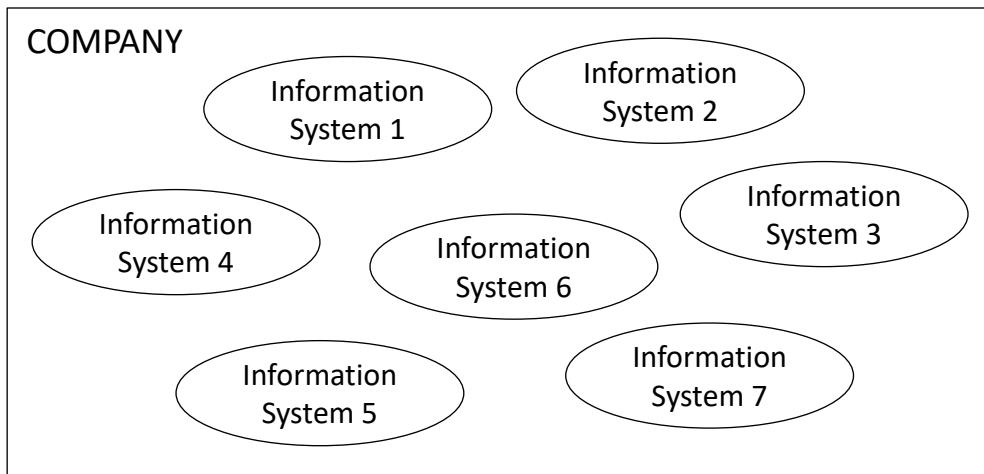




INFORMATION SYSTEMS

Company's Constituents

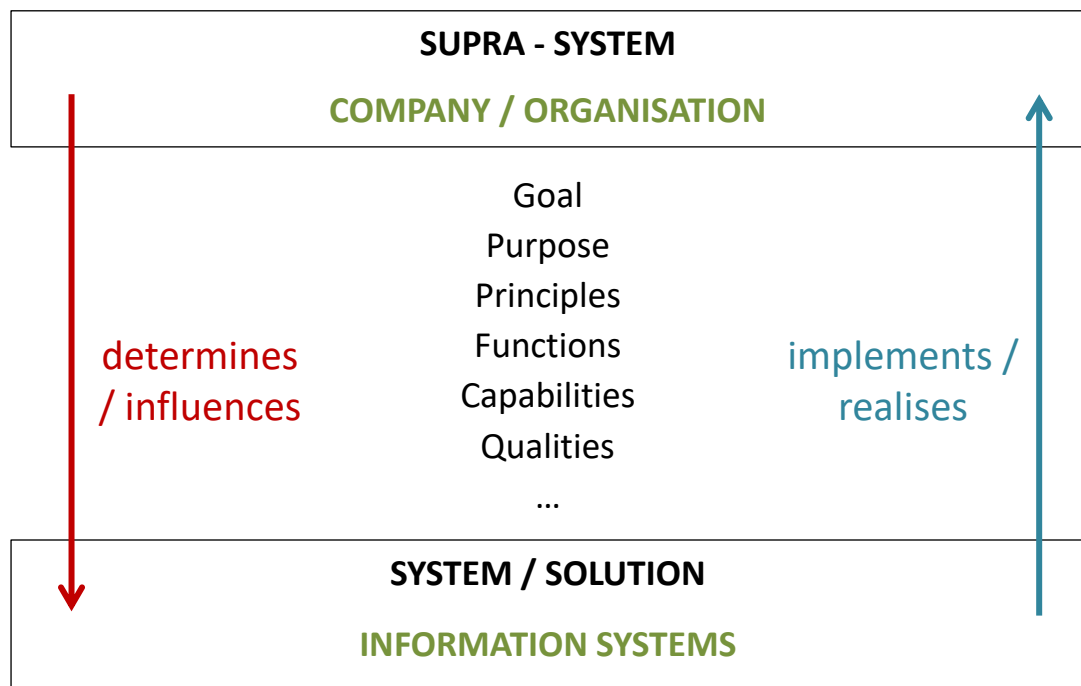
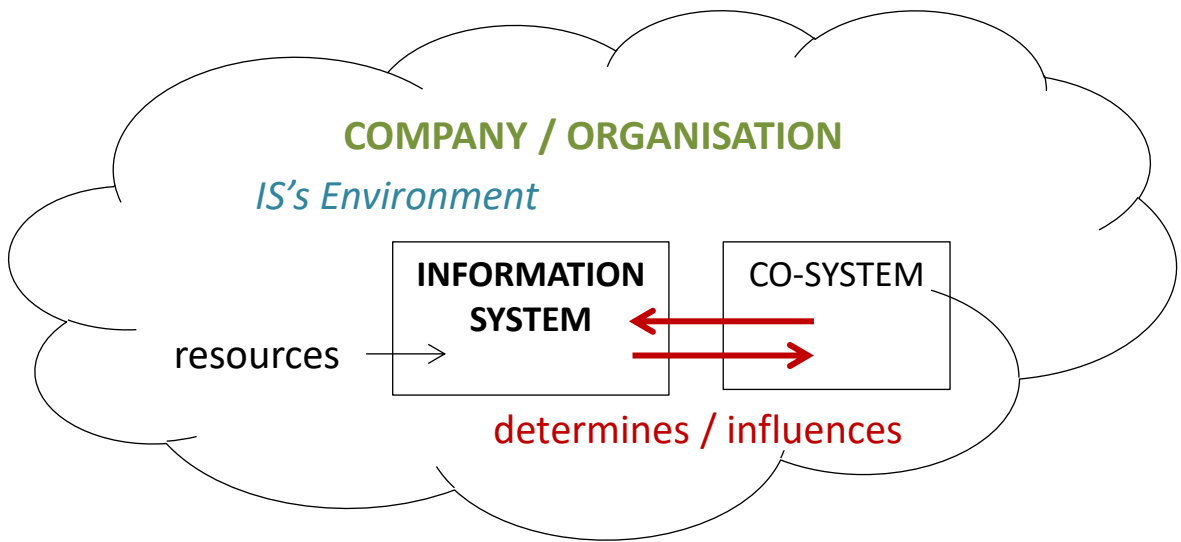
30/10/2018



IS's are part of the company. They partly form the company. The company, its activities and the overall environment have to be taken into account when designing an IS. Therefore they should be understood.

COMPANY = SUPRA-SYSTEM & ENVIRONMENT

30/10/2018



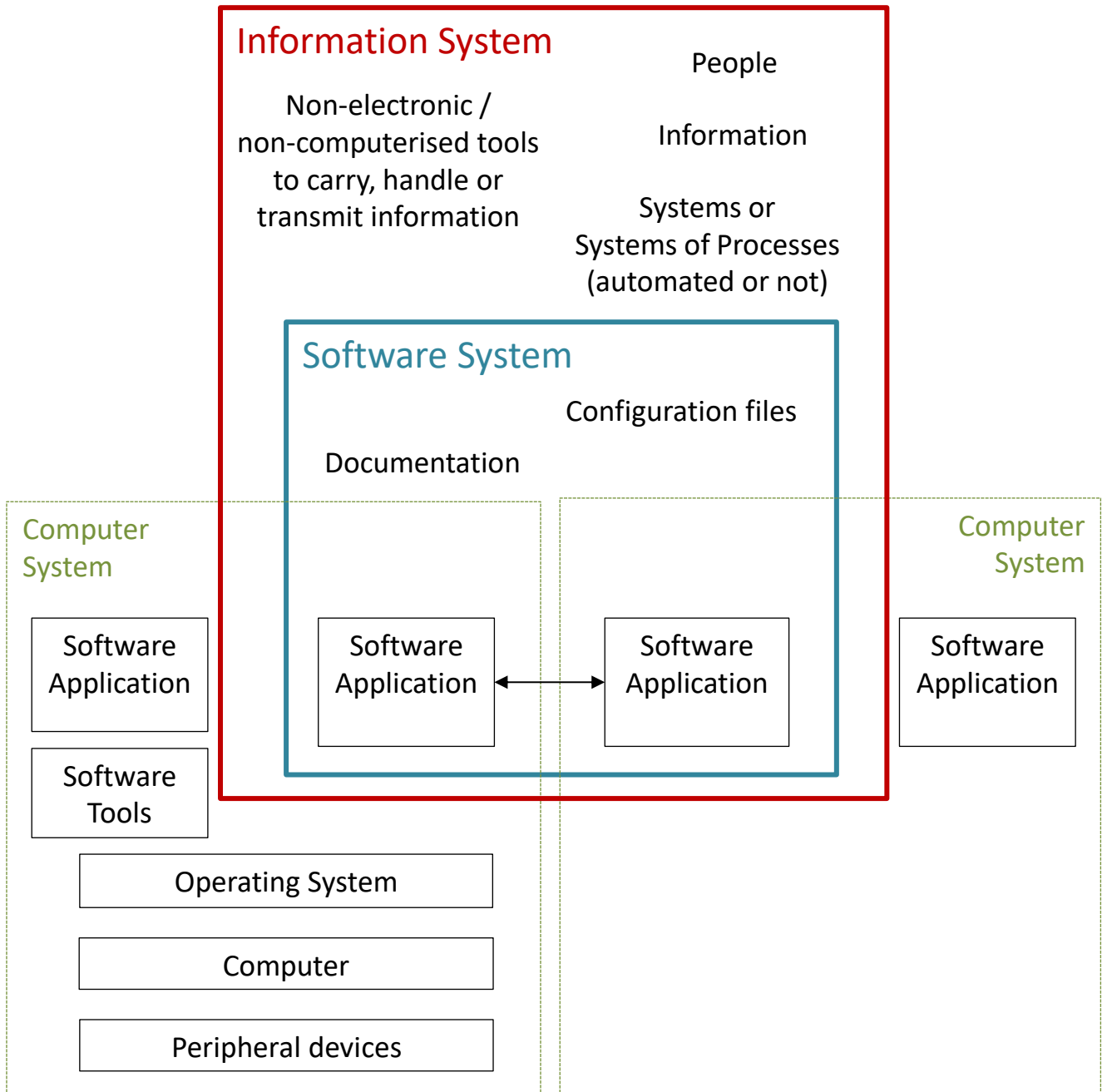
Supra-system determines or strongly influence the design of its parts

Reflection:

When designing a solution / IS, can we ignore the supra-system?
What is the relation between the company and the IS?
What is a company / organisation ?

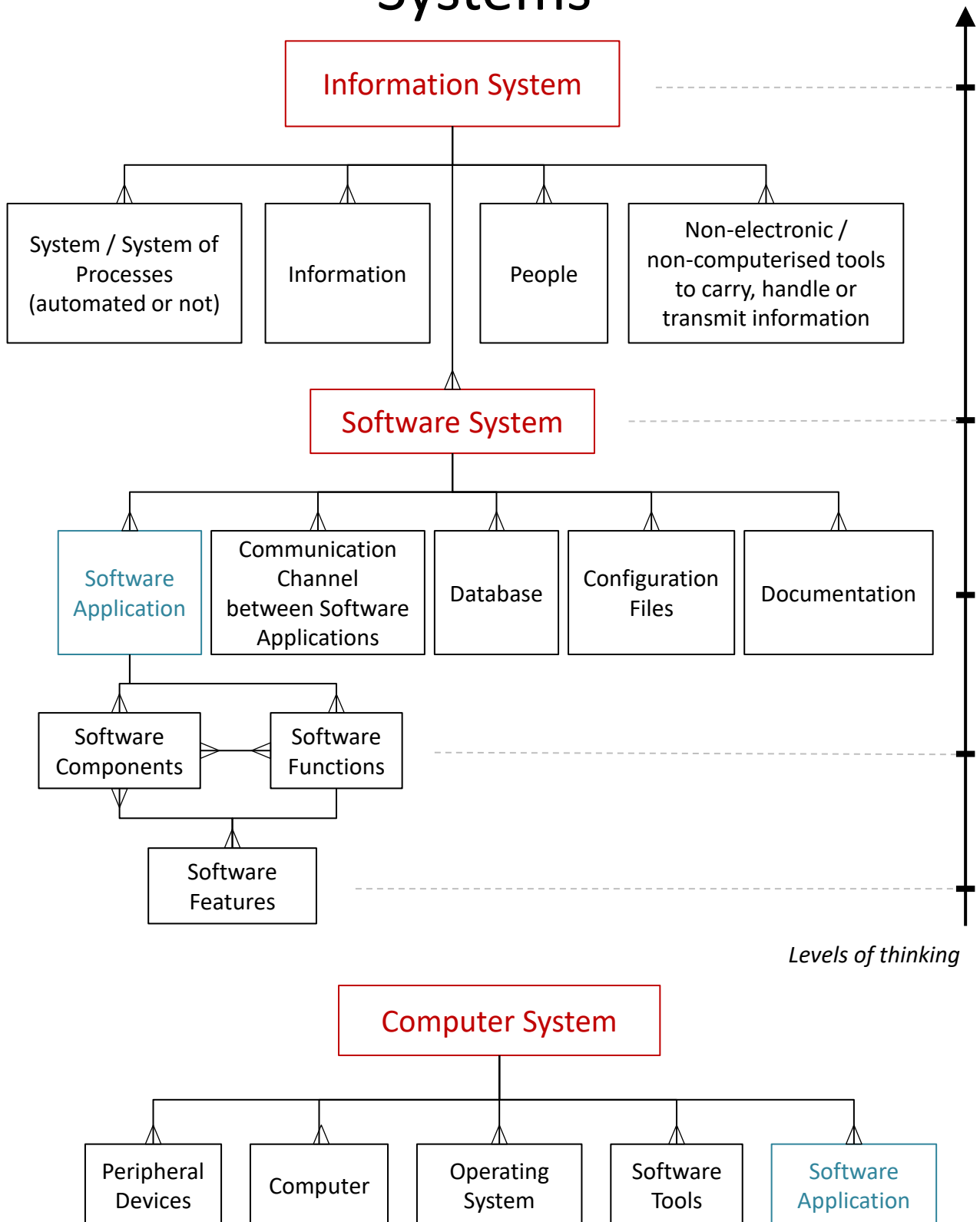
Information / Software / Computer Systems

10/01/2020



Information / Software / Computer Systems

10/01/2020



1. Transformation mechanism
2. Queuing mechanism
3. Filters
4. Buffering mechanism
5. Finding mechanism
6. Selection mechanism
7. Sorting mechanism
8. Prioritisation mechanism
9. Decision making mechanism
10. Measuring mechanism
11. Conversion mechanism
12. (Re-)Formatting mechanism
13. Decomposing mechanism
14. Grouping or assembling mechanism
15. Matching mechanism
16. Identification mechanism
17. Connecting mechanism
18. Integrity checking
19. Balancing / Assignment mechanism
20. Evaluation mechanism
21. Messaging system
22. Routing mechanism
23. Push / Pull Mechanism
24. Transaction / rollback mechanism
25. Feedback mechanism
26. Pattern detection/recognition
27. AI mechanism
28. I/O management system
29. Encoding / decoding mechanism
30. Log-in system
31. Access Control mechanism
32. Alarm mechanism
33. Notification system
34. Control unit
35. Monitoring mechanism
36. Debugging mechanism
37. Tracing mechanism
38. Logging mechanism
39. Archiving/ Restoring mechanism
40. Deployment mechanism
41. Patching mechanism
42. Configuration mechanism
43. Version control mechanism
44. Deployment and activating mechanism
45. Start, Pause, Resume, Stop mechanism
46. Clean-up mechanism
47. Back-up/Restore mechanism
48. ...

Functional Suitability

Helpfulness: deals with most important and valuable issues, create most value
Appropriateness: (solution is appropriate)
Process / Feature Accuracy: (solution is accurate)
Degree of automation
Degree of control over the processes

Operability

Technical accessibility
Understandability / Recognisability
Ease of use
Attractiveness
Learnability

Reliability

Availability
Fault Tolerance
Recoverability

Performance Efficiency

Time Behaviour
Resource Utilisation

Security

Authenticity
Accountability
Confidentiality
Integrity
Non-repudiation

Compatibility

Standard compliancy
Co-existence
Interoperability

Data Suitability

Data importance
Data organisation
Process-ability
Reliability
Completeness
Accuracy
Consistency

Integrate-ability

-

Maintainability

Maturity
Degree of control over the system
Supportability
Configurability

Adaptability / Evolve-ability

Expandability / Extendibility
Scalability
Replaceability
Documentation
Source code organisation
Analysability
Changeability
Modification Stability
Modularity
Reusability
Testability

Transferability

Installability
Portability
Adaptability

...

Sources:

ISO9126

ISO25010

SEI: Profiling Systems Using the Defining Characteristics of Systems of Systems (SoS) (CMU/SEI-2010-TN-001)

Other qualities:

Geographical spread

Cohesion, coherence, fragmentation of responsibilities

Cohesion, coherence, fragmentation of technologies

Degree of structured information

Matching between implemented concepts and reality

Degree of alignment

Degree of neatness and elegance of the source code

...

Needs of the Informatics Department

30/10/2018

The responsibility for the administration and management of information systems creates needs:

- Deployment system (updating, activation, upgrading, ...)
- Release & Version Management
- Configuration tools
- User and Access Management
- Start-up, On-Hold, Stop (shut down) procedures
- Monitoring
- Logging Management (tracing bugs, errors, exceptions, ...)
- Alarms
- Data Quality Control (E.g. consistency checks)
- Data Synchronisation
- Queue Management
- Performance measurements
- Environment Management
- Configuration Management
- Resource Management
- Capacity Management
- Hardware Management
- Archiving & unarchiving
- Problem Analysis
- Back-up & Restore
- Data Recovery - Disaster Recovery
- Clean-up
- Security
- ... ?

The Informatics Department
is a MAIN USER
of software application systems



ENGINEERING

Tale of the 3 Little Pigs

30/10/2018



Good
Enough ?



Seemingly
strong



Appropriate

Not (only) about materials (technology). It's also about

- Vision
- Mind set
- Norms
- Approach
- Skills
- Seriousness & Professionalism

LEARNING A LESSON: HOW NOT TO

Winchester Mystery House

Owner: Sarah Winchester (widow of William Wirt Winchester (firearm magnate))

Approach

- From the ground-up
- Owner and Builders meet daily
- No architect (no analyst)
- No global plan
- Based on sketches



A few numbers about the building:

161 rooms, 40 bedrooms, 40 staircases, 47 fireplaces, 17 chimneys, 6 kitchens, 13 bathrooms, 2 ballrooms, 2 basements, 3 elevators, 10.000 window panes, one shower

Building Process

16 carpenters worked in shifts 24 hours a day, 7 days a week during 36 years (1886-1922). Sarah met the carpenters daily. She hastily sketched designs on paper to explain what the carpenters had to build. There was no architect. Sarah built, demolished and then rebuilt. Sarah ordered errors to be torn out, to be sealed, to be built over or around it, or sometimes simply to ignore it.

Oddities

The house is a labyrinth. Windows are overlooking other rooms. Cabinet and doors open onto walls. Small rooms are built within larger rooms. Some chimneys don't reach the ceiling. Stairs have odd-sized risers or leading to nowhere. A door opens out into nothing. Balconies are inside. Floors have skylights. This approach led to a chaotic architectural jumble. Rooms were hidden and lost behind new construction. Just to name a few.

Today, the mansion is a tourist attraction

LEARNING A LESSON: HOW NOT TO

How many systems are “engineered”

Movie: “The Pentagon Wars” – 1998

Movie is about the 17 year development of the Bradley Fighting Vehicle. The movie depicts how all kinds of hierarchical interventions in the engineering process of the system lead to bad or even unusable systems.

Watch the movie here:

<https://www.youtube.com/watch?v=ir0FAa8P2MU>



- Easy
- Quick
- Not much thinking
- Little effort
- Instant gratification

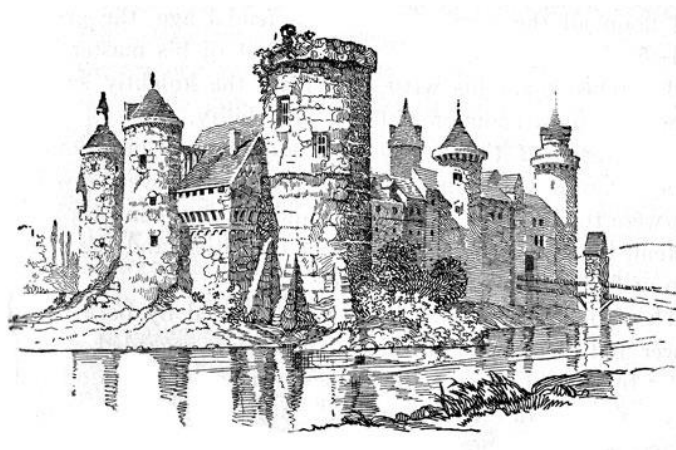
Little short-term successes

Works for little things that don't matter too much



A few items are still manageable. Many little unorganised and unmanaged items may quickly create chaos.

Are we still able to build things that do matter and which last?



Notes:

The castle is a system that is vital, well-thought, functional, practical, lasting, operating as one system, ... and critical for thriving. (It's not about being outdated or monolithic). It is about the focus and mind-set to build great systems.

This metaphor helps thinking about what corporate IT is required.

Customers may find it long to wait for their meal. They want it quick, cheap and tasty. They want a great experience. The restaurant may deliver that.

To deliver a decently cooked meal, it takes some time. The customer has to have some patience. If he can't wait, either he will have a badly cooked meal.

The restaurant may also try to look for simpler meals taking lesser time to cook. Possibly, it may end up transforming itself into a fast food. Fast food restaurants are "restaurants" without chefs.

The restaurant may seek to offer a great experience or to offer them nice looking and very tasty meals.

The restaurant may seek to solve the hunger of their customers. And customers may be glad not to have hunger anymore. However, since hunger is just a symptom, it ends up solving symptoms.

Resolving the desires of experience, taste and hunger can be fine. However, in the long run, it may have devastating effects. The human body needs a healthy diet to be healthy and to live long. It needs specific nutritive elements on regular basis.

Patience and decent meal is maybe not what the customer may ask for, but it is what he or she needs, particularly when we consider the broader picture.

Decisions made based on pleasing or on short term may undermine the system, the foundation and the long term, particularly if they are made frequently.

A chef needs to understand cooking tools, cooking techniques and taste. He or she needs to understand what (s)he creates and the materials (s)he deals with. (S)He has to understand food and nutritive elements, their value, their nature, what happens when they are processed, and so on.

A good restaurant can not be satisfied with a chef that knows only his tools and techniques or one who can only cook by using recipes. Recipes are fine. But they are only the basic process. It has always need to be adapted to the specific situation. Being able to follow a recipe doesn't make a great chef. Recipes don't replace the knowledge of the chef or the cooking skills.

Engineering is the application of **science** and technologies to **design** and **build** machines **structures, systems** and other things.

- **Science**
 - Knowledge + understanding + thinking + principles + methods + techniques + tools
 - Way of thinking (methodical, abstract, logical, ...)
 - Attitude (curiosity, sceptical, doubting, truth seeker, sharing, seeking confirmation, ...)
 - Specific intellectual skills
- **Design and build**
 - Conception (decision making) + specific intellectual skills
- **Structures, systems, ...**
 - Understanding of structures, mechanisms, systems, processes, ...

Notes

Can all building activities be called 'engineering' ?

Often, engineering is limited to imitating the visible actions of engineering and to produce a concrete outcome. This completely misses the essence of it.

Myth: Users can engineer the system they use.

Man-Made Systems

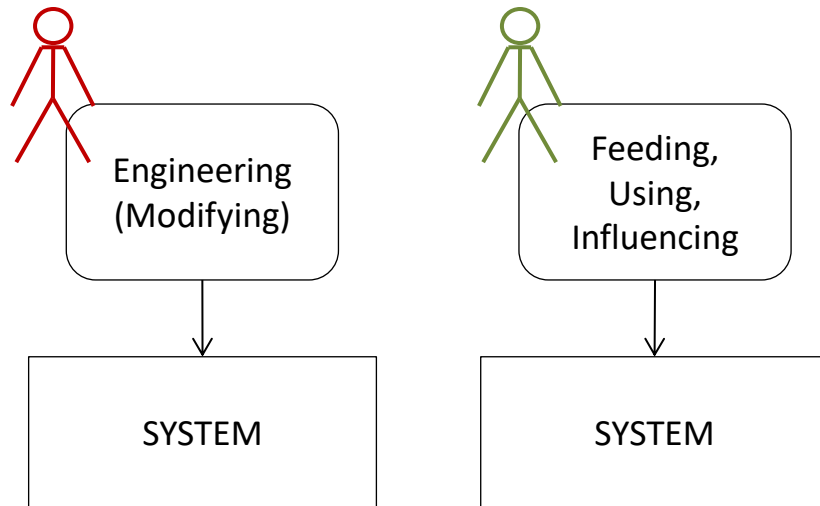
A man-made system is the
creation of our brain

Therefore

- our understanding
- our vision
- our thinking patterns and habits
- our norms
- our mental flexibility
- the openness of our mind
- our creativity
- our knowledge
- our competences
- our experience
- our priorities
- our collaboration
- our emotions

but also our interpretations, assumptions, misunderstandings,
distortions, unknowns and other weaknesses

are reflected in the design
of the systems and solutions we create.



Why we need to learn and understand systems?

Each automated system has a purpose. It implements a mechanism based on principles. It has characteristics, limits and constraints.

A person, as a user, a manager or as a part of the system, feeds, uses and influences the system. Dealing in a wrong way with the system may create damage. The **system will inevitably and indivertibly respond accordingly to its own logic**, irrespectively of that person's knowledge, opinions or intentions. Even when it breaks, it will do it by obeying to its own logic and to natural laws.

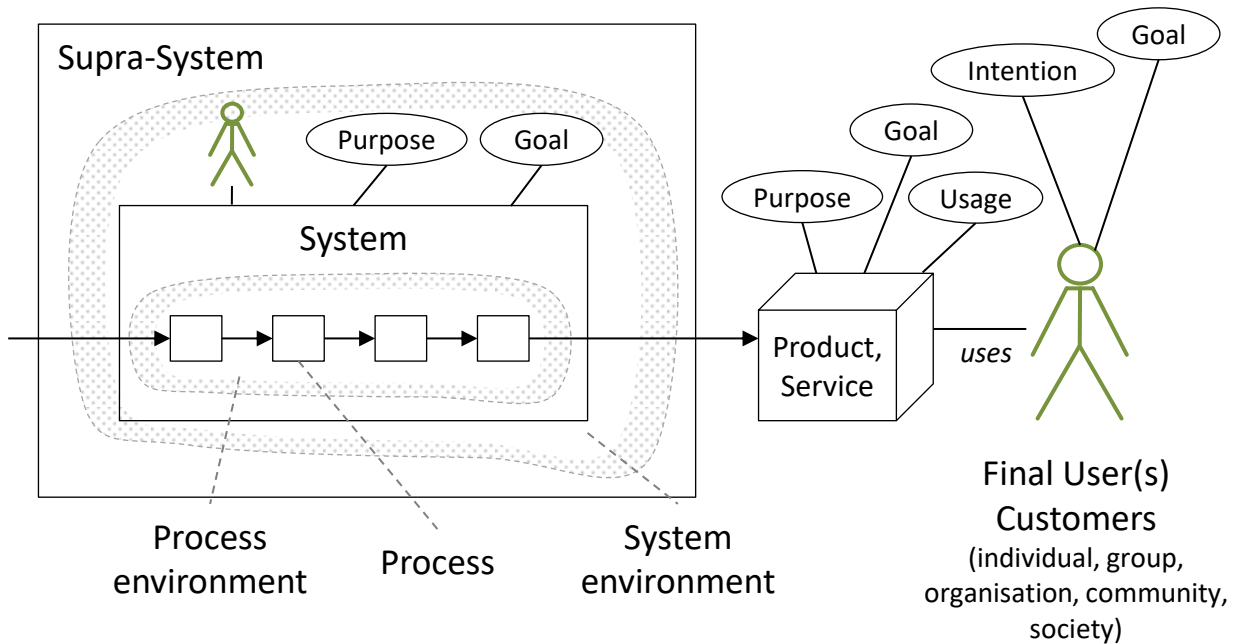
A system always obey its own nature, its natural laws, its embedded principles, its built-in logic its characteristics corresponding and to its limits. Therefor, these logics and qualities have to be respected.

MEDIOCRE DRIVING IDEAS

Fill the gap
Get the problem solved
Get the job done
Follow the masses, tendencies
Respecting standards
How does anyone else do it?
What do you want me to do?
Analogy, Copy, Buying, “this is similar to”
“Having it working is just fine”
Local, short term

SUPERIOR DRIVING IDEAS

Higher goals
Purpose-based
Root causes-based
Holistic
Innovation, New ideas
First Principles
Natural laws
Multi-dimensional
Sustainability
Long-term
Evolution



Product can be: service, information, knowledge, energy, object, system, organisation, network, infrastructure, experience, ...




GOALS DRIVING INITIATIVES

- Contributing to the Goals of the Supra-system
- Improving the Supra-system
- Satisfying the User of the System
- Helping the User of the System
- Contributing to the Purpose (Role) and Goal of the System
- Improving the System
- Improving the Process
- Improving the Environment and Actors executing the process
- Contributing to the Purpose and Goal of the Product
- Improving the Product
- Satisfying the Final Customer
- Helping the Final Customer
- ...

Goals Driving Initiatives

30/10/2018

OR →

		Goal : Customer Satisfaction	
		Dissatisfied	Satisfied
Goal: Company / System Design	Excellent	Dissatisfaction is possible, probably fixable	
	Inappropriate	 Undermines the Org./System Hard to fix	 Undermines the Org./System Hard to fix

Excellent System Design

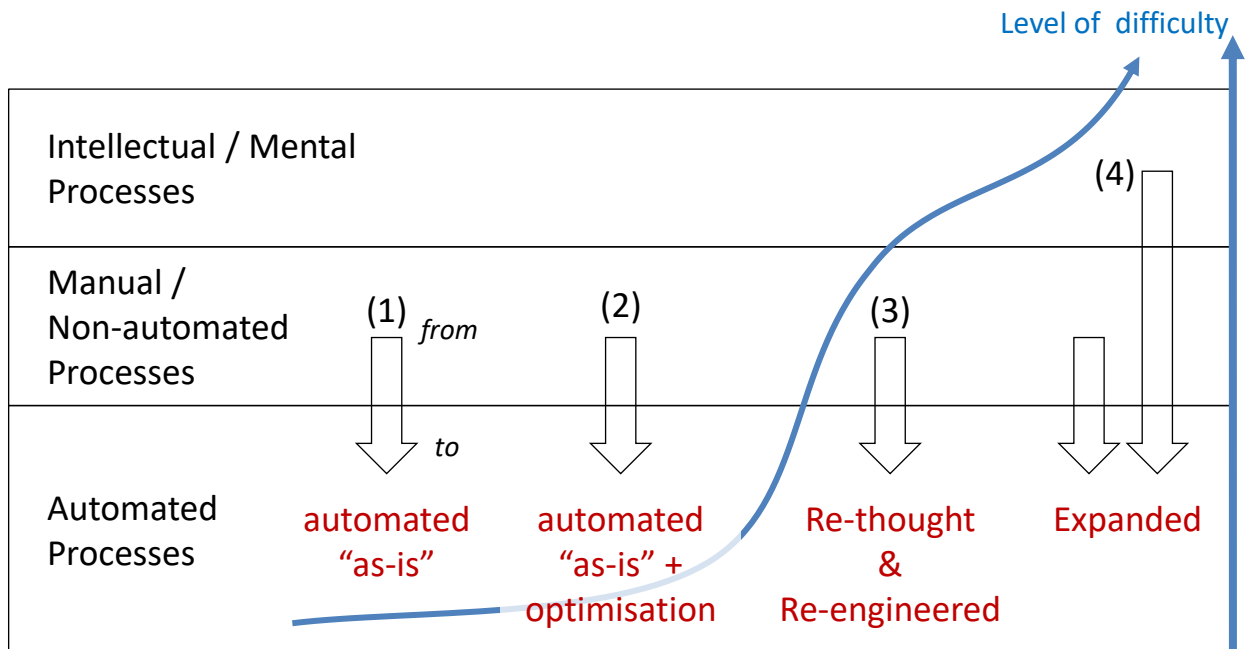
Everything that shapes systems,
affecting the Structural Design and Core Mechanisms

- Core logic of processes
- Core information concepts
- Organisation of the core of information,
- Landscape of systems (roles of systems, areas, interactions and collaboration between systems, (business) functions per (sub-)system)
- Systems solving wrong problems
- Integration, fragmentation, separation, duplication, interoperability, scalability, ...
- Missing sub-system and mechanisms necessary for the control, management, evolvability and survival of the system
- Architecture

It does not include: issues with features, adjustments of processes, missing information, end-user interface issues, ... they can create dissatisfaction, but does not require rethinking and reengineering core aspects of the system to be solved.

Inappropriate design

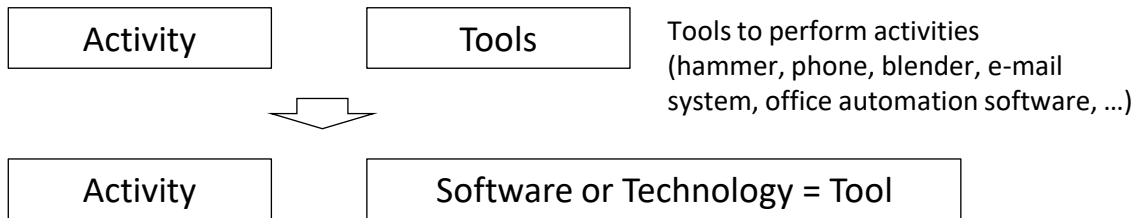
The core of the system is inelegant, weak, creating imbalances, internally chaotic, ...



1. Automation of the existing non-automated processes. "Show me how you work, I will computerise it." Same processes, but faster execution. Risky. Computers aren't people. These sets of processes may have grew organically. They can be very inefficient.
2. Automation of the existing manual processes, but introducing optimisations, such as faster information transfers to other organisational units, removing unnecessary manipulations, etc.. Some processes can be added, but they don't revolutionise the system.
3. The whole system and processes are reviewed, re-thought and re-engineered. Great gains can be obtained.
4. Implementation of the intellectual processes of people. This requires the analyses of how they use information, how they interpret it and how they think based on that information.
5. **Conception of NEW system** (not in diagram)

Thinking Pattern: Activity + Tools

10/01/2020



Traditional Salesman's View / Consumer's View of IT:

“Buy, build or provide tools, be it software applications or technologies, to support or perform our activities.”

“I sell what you need” – “I buy what I need”

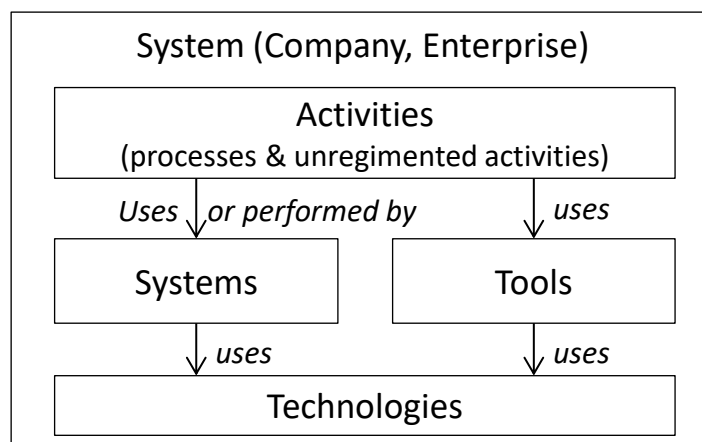
“Adding (what is needed) allows a better functioning”

(“IT is about delivering tools & technologies”)

This thinking is applied in a **household**. A company's budget is much larger than an household's budget. A company is not a household.

This way of thinking, in a business setting, is linked to local & operational level, users level. Compartmentalisation at lowest level, Fragmentation, stovepipe-thinking, sub-optimisation, ...

The company's situation is a bit different:





Small Projects

Usually :

- Simple, easy and short
- Can be done with limited understanding of the discipline, limited skills
- Can be done with little organisation, simple environment, little management
- No or little incentive to learn about the discipline and to improve
- Everybody can do it
- Easy to be successful (small success after a few days)
- Small, unimportant, superficial things can be achieved



Large Projects

Usually :

- Complex, difficult, long
- They really **test** our capabilities, understanding, ...
(one can't pretend to be skilled or expert and do only basic, small, easy, basic work)
- Requires great insight in the discipline, advanced skills
- Requires a more developed, formal and managed organisation and environment
- Obliges people to learn and to improve their skills
- Not everybody can do it
- Is NOT synonym of / does NOT imply "big bang" development/implementation !!!
- Serious stuff, things that do matter, sizeable challenges, ...
- Not necessarily harder to be successful (much greater result over longer period)
- People built great things to show their skills. Societies built great things to show how advanced they are.

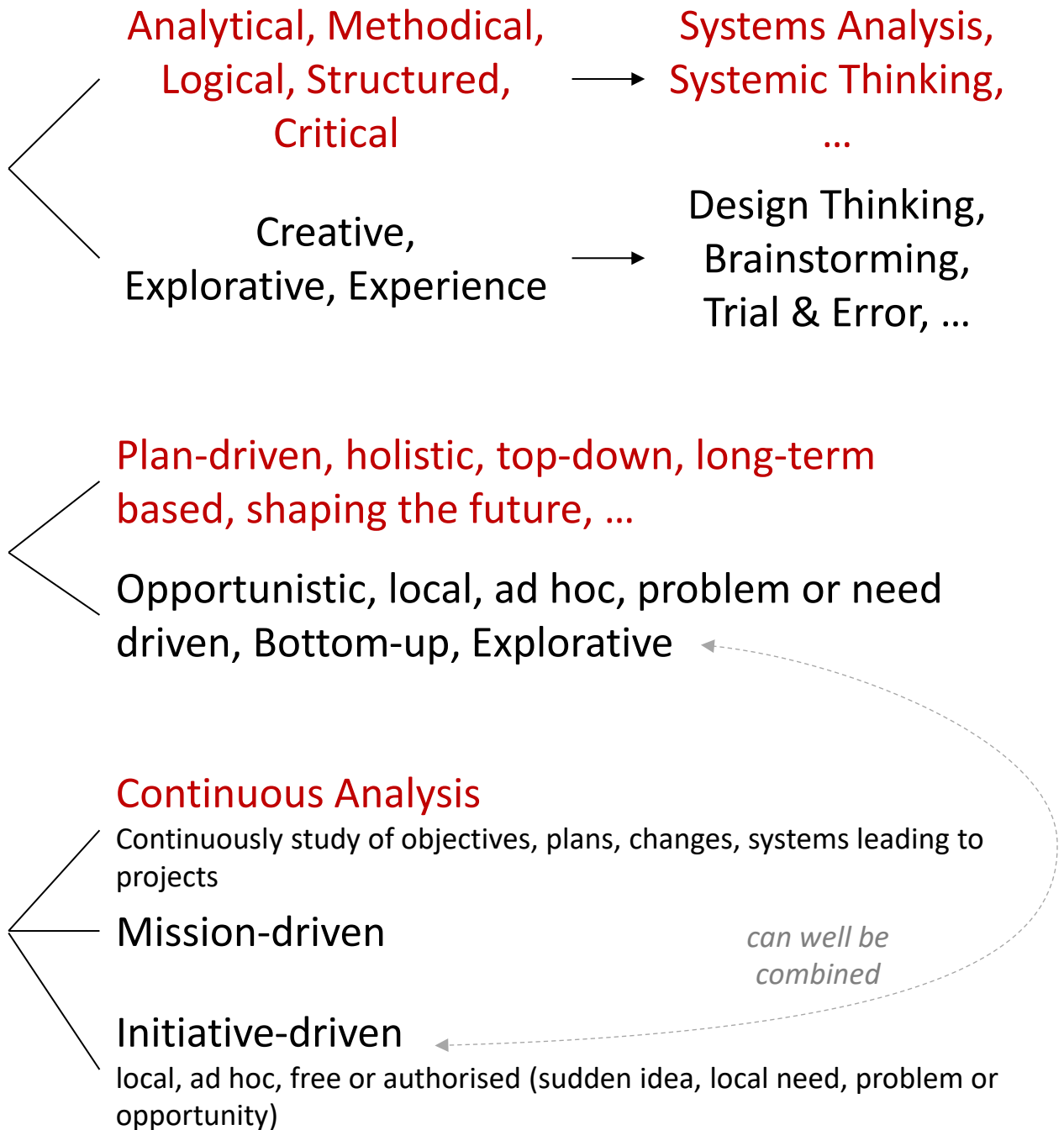
Reflection: Should large project be avoided because

- We can't do it?
- We want to be successful? Fear of "Failure" ???
(What if success is defined by tangible results instead of by "meeting estimations"?)
- Impatience & quick wins
- Development of solid skills, practices, insight, ... ???

Or, on the contrary, are there good reasons to learn doing
sizeable and larger projects?

Some Major Engineering Approaches

30/04/2019

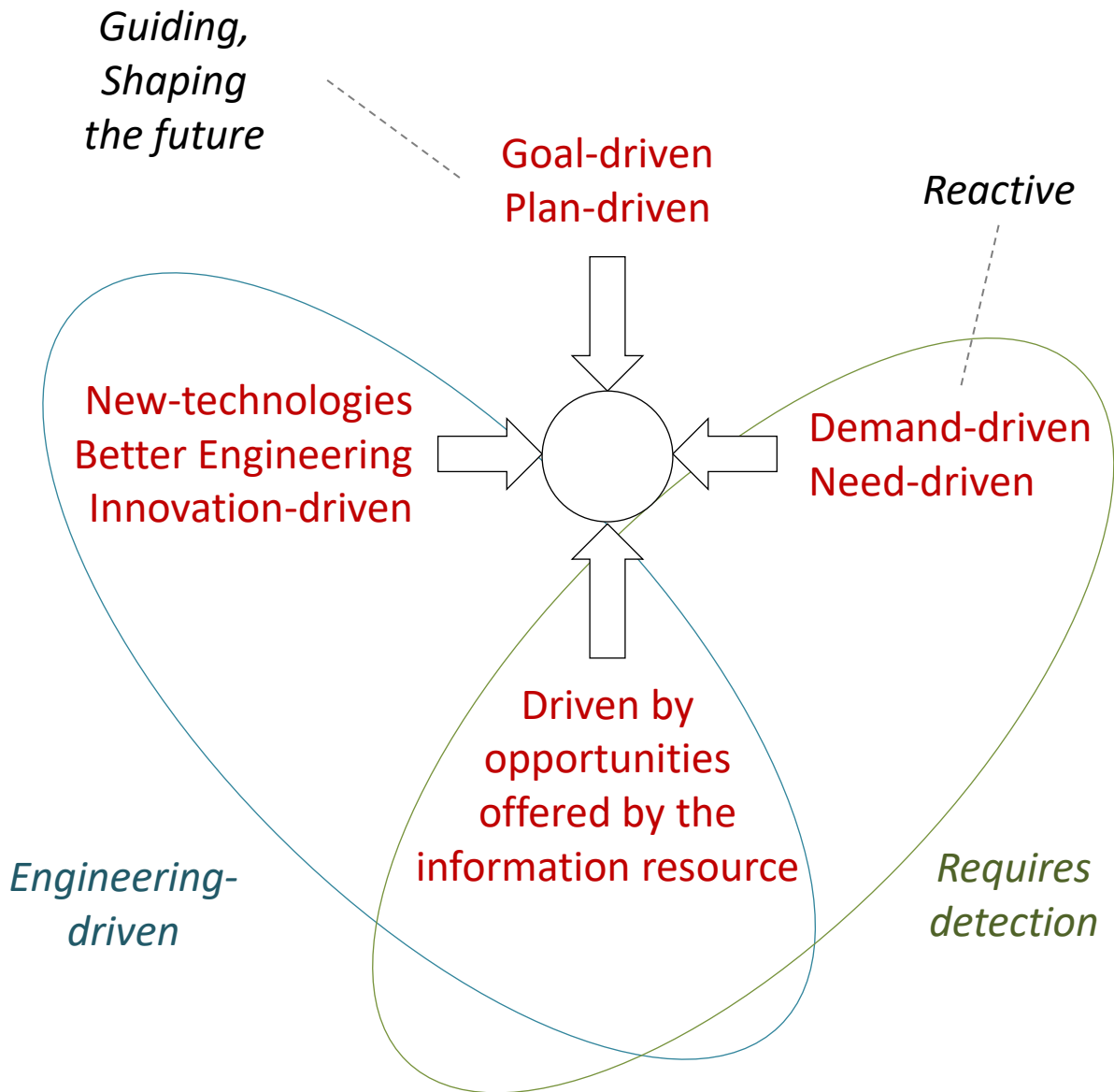


Notes

- Some approaches can be combined in one.
- A company may need to have or allow different types of approaches. It's not about choosing one single approach for the whole company. Usually, one main approach can make sense.

Types of Approaches

17/09/2019



Notes:

Companies need three, and preferably the four types of approaches.

All approaches based on existing needs, existing problems or on demands

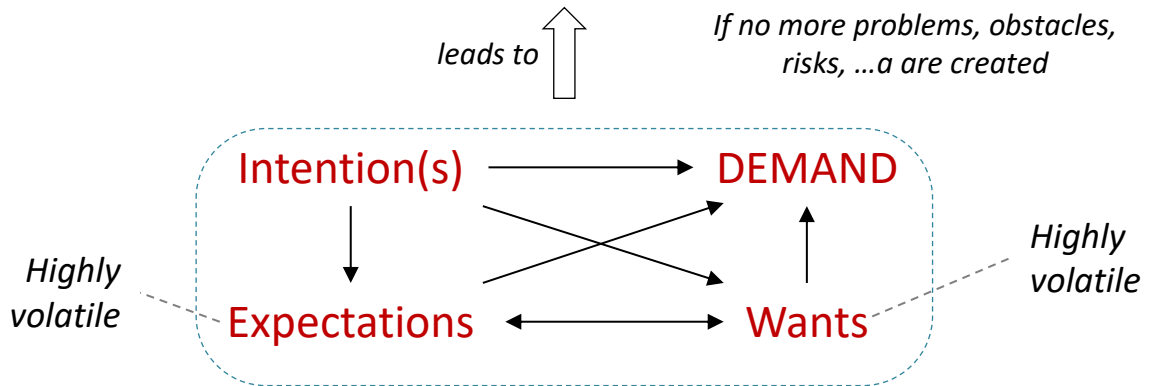
- Problem exists and creates already damage.
- Need exists and is already a missed opportunity (cost)
- No control over when what problem, need or opportunity will pop-up
- No control over the direction
- Following, not leading or shaping the future
- Always lagging, always gap between problem, issue, need or opportunity and solution
- Existing problem and opportunity creates frustration and impatience
- Creates pressure
- Assuming received information is reliable and true. No verifications.
- Pressure, assumptions, lack of verifications lead to mistakes, bad solutions, misunderstandings, conflicts, stress, exhaustion, ...
- Focused on solution design, instead of on (understanding) the problem (understanding the demand or symptoms \neq understanding the problem)
- The less time is spent on verifying, learning, thinking and testing, the more time can be spent on building and the quicker something can be delivered.
- Focus on building without understanding. Building based on assumptions.
- Limited to translating the demand into design of 'something' to be built.
- Less competencies are used.
- Not proper for innovation

Is a reactive approach appropriate as main stream type of approach?

15/03/2019



CUSTOMER SATISFACTION



*Intention, wants, expectations and demands
of **business stakeholder**, customer, end-users, sponsor*

What is the **QUALITY** of the Intentions, Wants, Demands and Expectations ?

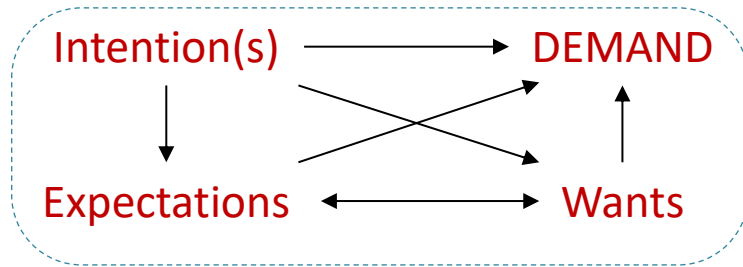
MUST be:

Justified
Possible
Coherent
Complete

Stable
Reliable
Good / Optimal
...

Demand and Want Analysis

15/02/2019



The average analyst has **no (or little) clue**
on what this is based.

He/She **doesn't know the reasoning** behind it.

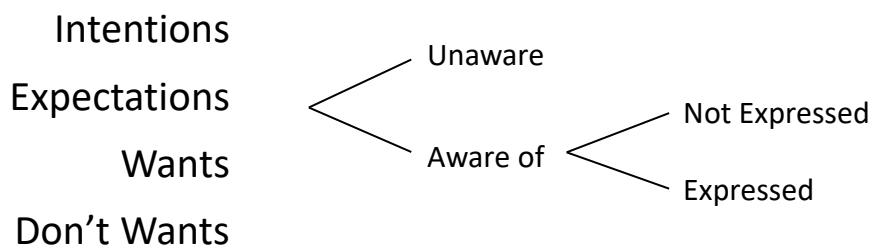
How much is this **in line with reality** ? (cfr practical versus choice problems)

What changes these expectations and wants?

- New information
- New insights, deeper insight
- Detection of contradictions
- Uncertainty, hesitation

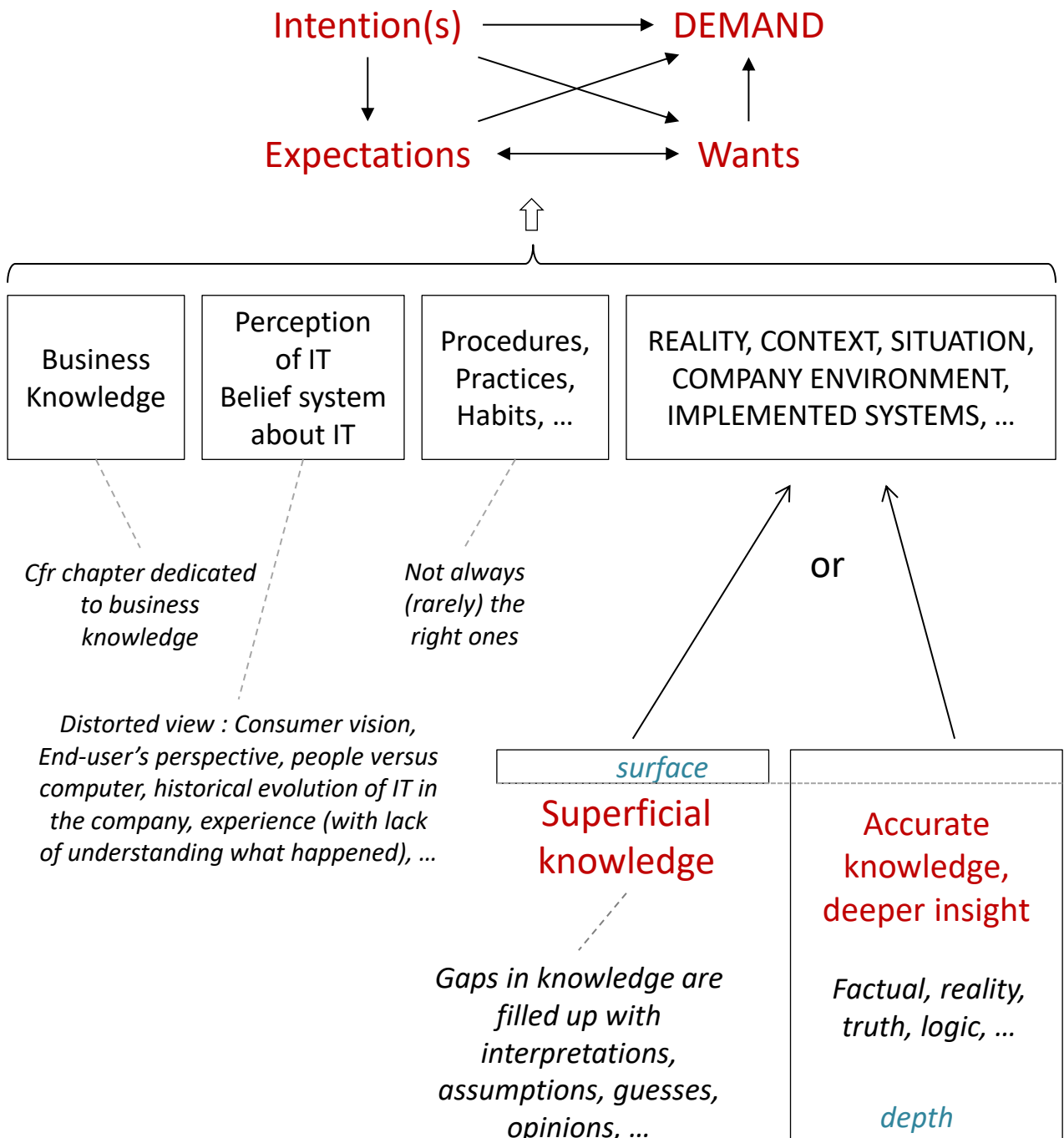
We don't know ...

- whether the demand, the wants, the expectations fit with the intentions
- whether it solves the problem,
- we don't know if responding to the demand, expectations, wants that customer satisfaction is created. We simply assume this. Projects and actions based on assumption ... ???

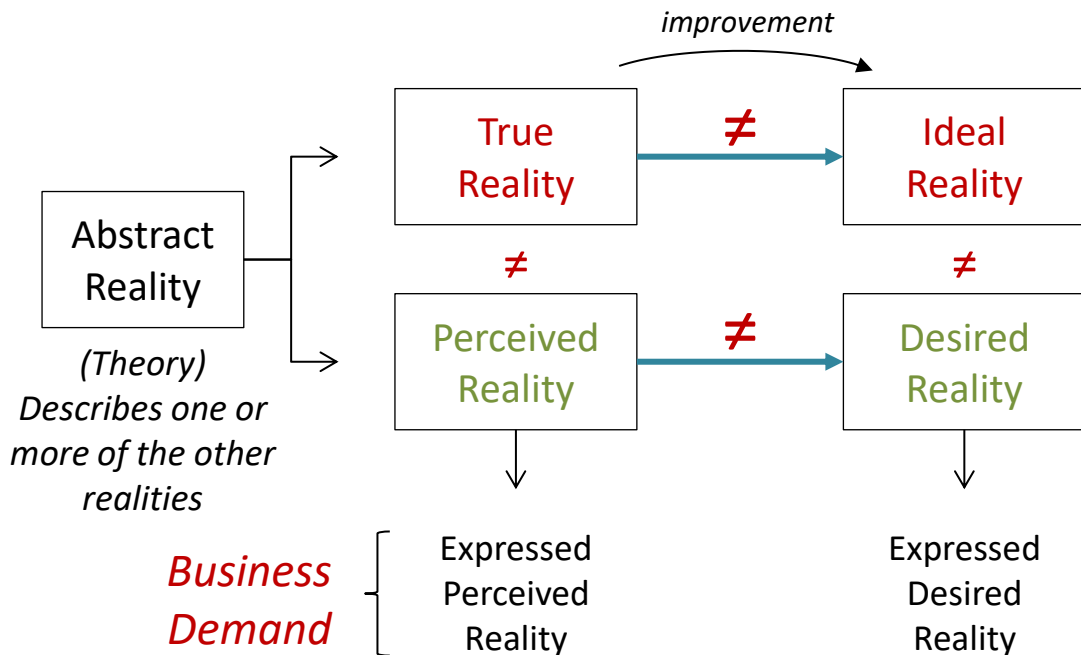


Demand and Want Analysis

15/02/2019



Much Work for the Analyst



It is problematic when

- the perceived reality (subjective reality) doesn't match the or true (objective) reality.
- the desired reality is based upon a distorted perceived reality
- the desired reality is too different from an ideal reality

The Analyst has to

- Study the true objective reality.
- Identify the
- Align perceived reality with the true reality and the desired reality with the ideal reality
- Identify the path between the present objective (true) reality to a more ideal reality

Do not rely and build upon distorted, incomplete or disadvantageous reality.

Beliefs based on the ASSUMPTION:

The demand is correct, complete, reliable and stable.

- The demand describes the needs.
- The demand describes what is required.
- Responding to a demand = solving the problem
- Responding to a demand creates satisfaction.
- Understanding the demands = understanding the problem
- Understanding the demands = understanding the situation
- Understanding the demands = sufficient to create a solution
- The demand takes the situation and all aspects and parameters fully (sufficiently, reliably) into account.
- If the demanded solution is built, value is created.
- Projects (always) produce solutions.
- Business community can diagnose its information problems and conceive information solutions.
- Business community knows what software and computer systems can do and can't do. They understand their power, constraints and impossibilities.

Information is critical for the company.
Information systems, and their automation, constitute a huge benefit especially for larger companies.

Building Software Systems is ...

- very Complex
- very Slow
- very Demanding

Techniques to speed up development:

HARDER

- Tackling the right problem rightly from the first time
- Planning
- Creating an appropriate environment: relational environment, appropriate work atmosphere and physical environment
- Availability of people and competencies
- Smooth decision making processes
- Project team support
- Organisation of artefacts
- Sound architecture and concepts
- Internal organisation of source code
- Parallel development
- Good choice of standards, tools, technologies
- Modular software components and Reuse

Other techniques:

EASY

- Increasing pressure (shorter deadlines)
- Increasing the number of people
- Taking all kinds of shortcuts
- Leaving out intermediate steps and work
- Higher risks
- Lowers the quality
- Defers costs to the future, which may turn out to be much higher and/or hidden

Notes:

Shortcuts: ignoring complexity; ignoring the purpose, the value and the importance; pretexting "it has no value", "it's faster", "we can already start" (leaving unsettled questions for later), ...

These decisions require understanding of projects and of ISE/SE.

- Delivering well after the need or problem exists
- Not fully exploiting the conceptual possibilities
- Not fully exploiting the information
- Solved consequences leaving true causes unresolved
- Exaggerated and unnecessary complexity
- Oversimplifications
- Built-in obstacles and limitations (to be circumvented)
- Fragmentation
- Ballast, unnecessary and unused source code
- Confusing explanation and logic (documents, source code, ...)
- Loss of knowledge and insight
- Unreliability, vagueness, contradictory information
- Information glut
information chaos, duplication, not inventoried information, various formats, lack of connections, lack of meta-data, unreliability, bad information organisation, unnecessary and outdated information, ignored information, inaccessible info, ...
- Chaotic documentation, information, source code, systems architecture, ...
- Lack of alignment
with needs, with objectives, with business knowledge, functional alignment, conceptual alignment, rules, technological alignment, misalignment in standards, ...
- Unappropriated and limiting concepts, processes and architectures
- Systems, software applications and source code that are hard to change
- Unnecessary processes (like data transformations)
- ...

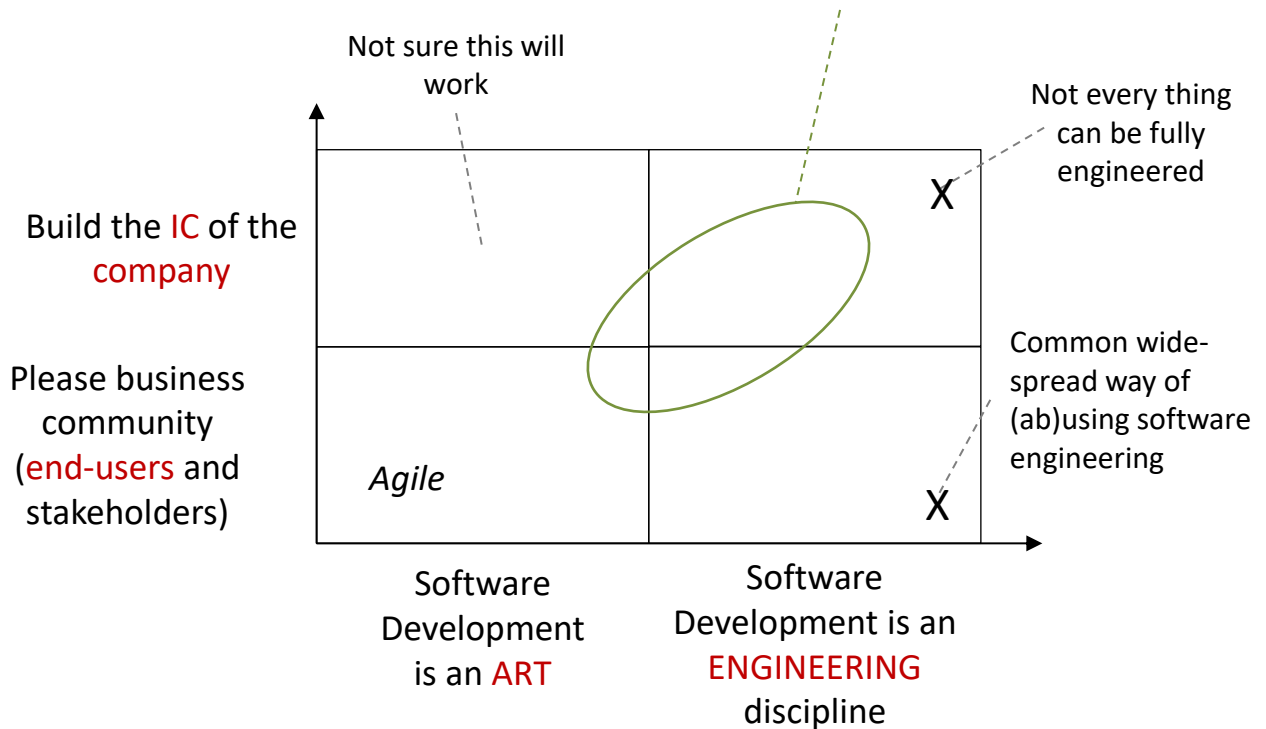
Note:

A lot of hindrances are not about programming or technological
These “causes” are consequences in their own right.

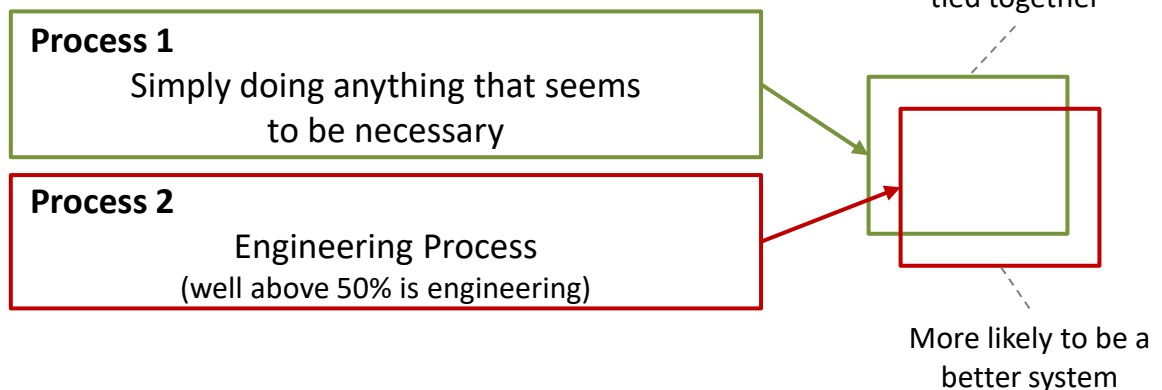
Art vs Engineering

15/02/2019

- Mainly an engineering discipline, partly also an art.
- Building the IC of the company, when possible also please end-users

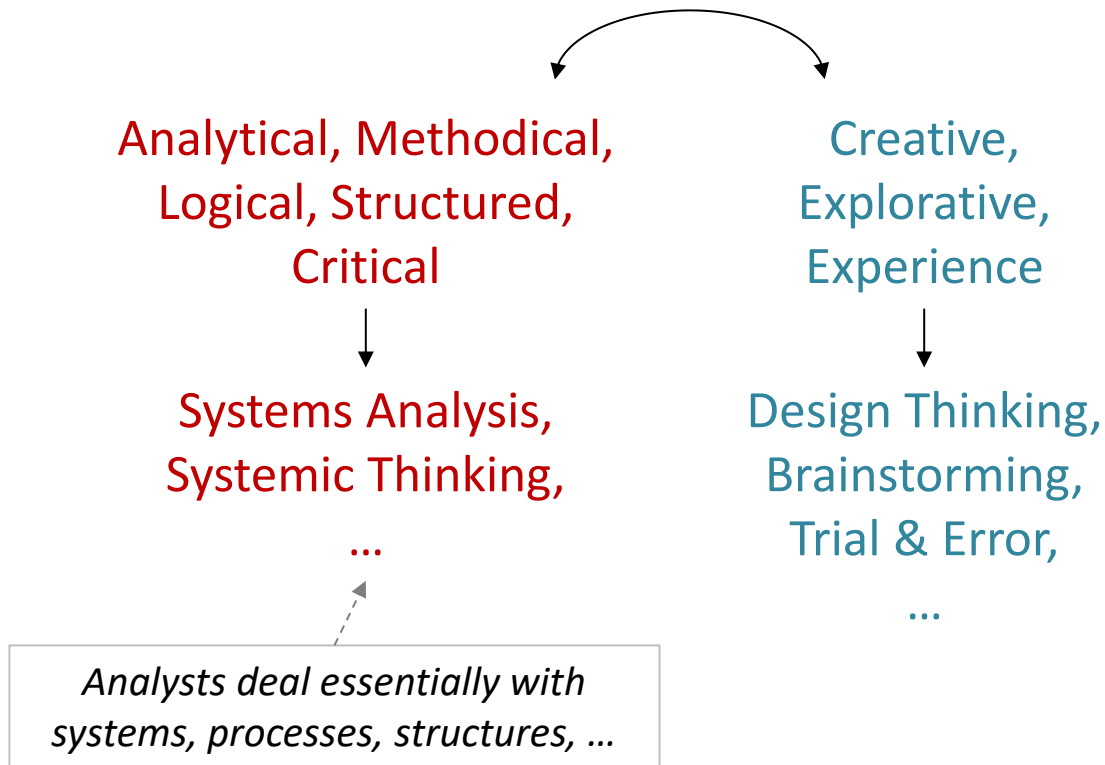


Two System Building Processes

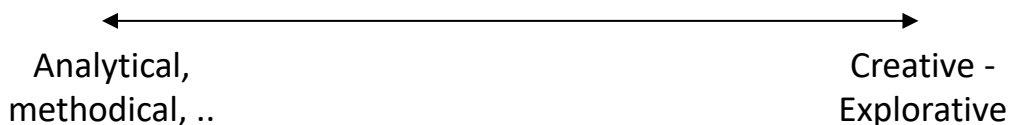


Not every system development process can be called an “engineering process”

Two Major Engineering Approaches

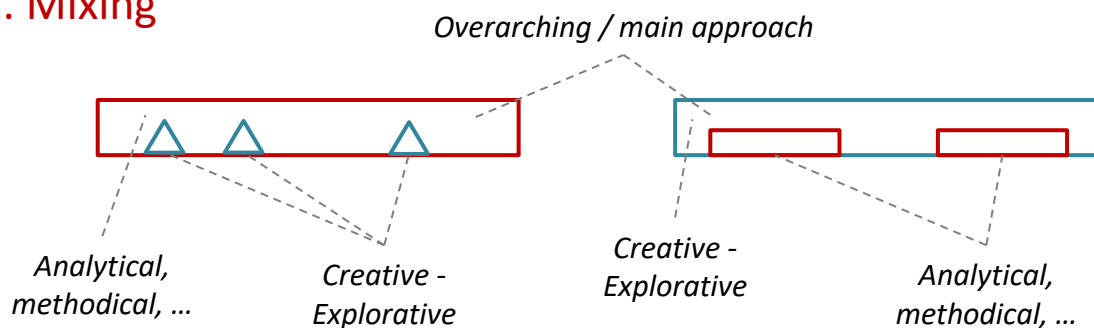


1. Continuum of possibilities



The extremes (or pure forms) are clearly different. However, there is a continuum of possibilities between these two extremes.

2. Mixing



Two Major Engineering Approaches 15/02/2019

LOGIC - driven

- Rational, based on logical thinking, reasoning
- Product, result is more important than taste
- System-oriented
- Architecture-centric, component based
- Requires understanding, holistic, knowledge-based
- Top-down, integration
- Plans and models
- Some more control
- Rigorous, methodical, structured
- Longer iterations (short iterations is possible)
- Examples: Systems Design, Systems Thinking

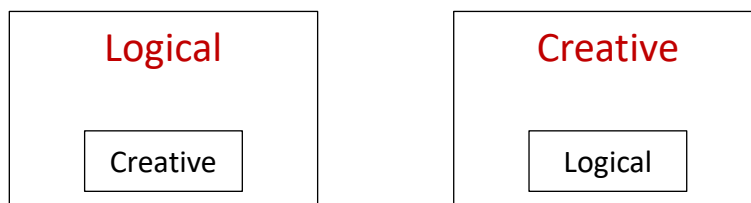
CREATIVITY - driven

- Experimental, trial and error, experience
- Usage-centric, people-oriented
- Can be matter of taste
- Creativity, idea generation
- Ad hoc, unpredictable
- Short iterations, fast feedback
- Lesser structured approaches
- Requires lesser understanding
- Lesser (supra-)architectural
- Bottom / Bottom-up
- Integration is lesser important
- Examples: Prototyping, Design Thinking, user-centred design, ...

Creativity-driven gets more attention these days increasing the uncertainty, unpredictability and amount of changes.



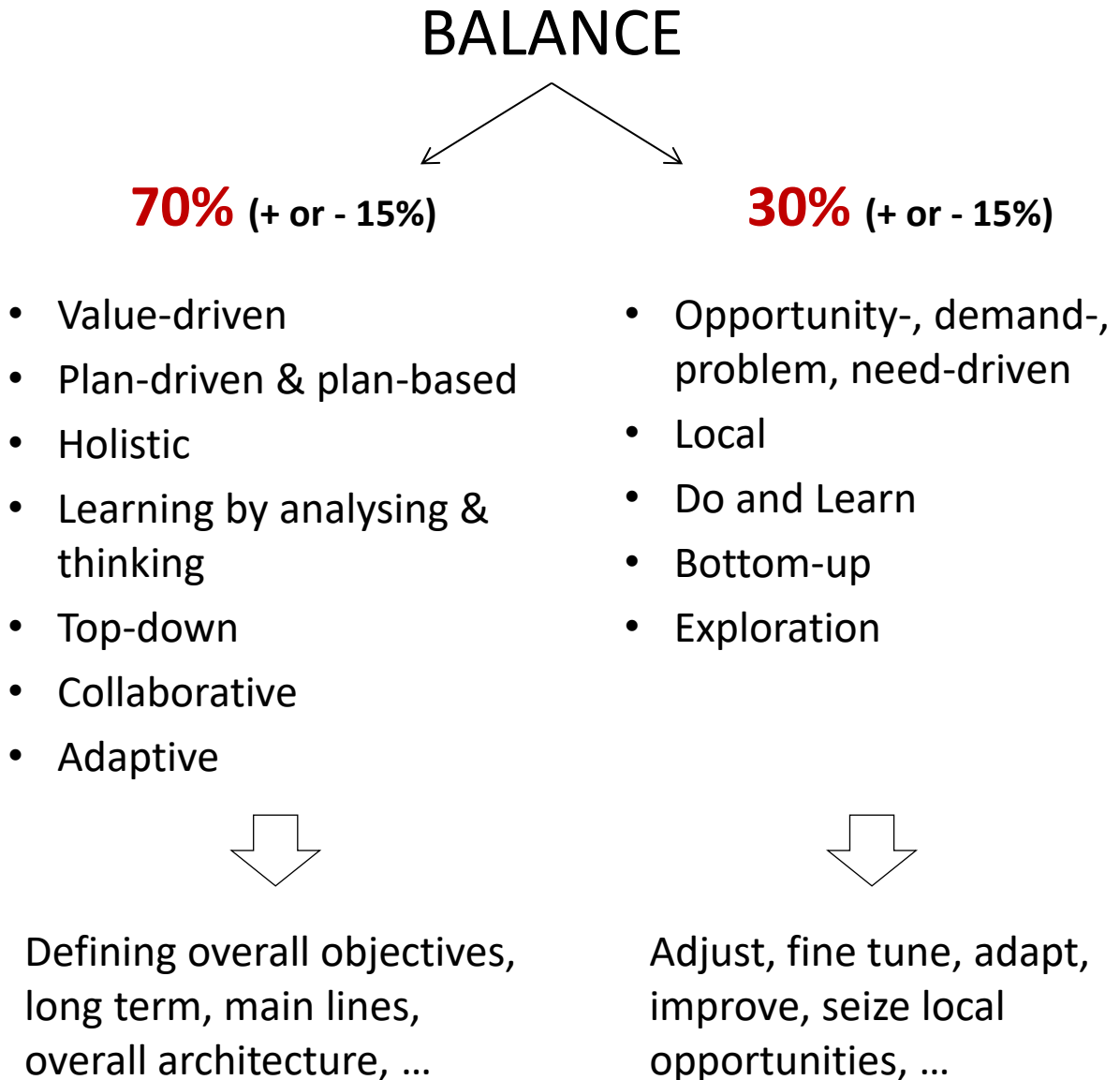
Each type of design approach uses the other, but to a lesser degree (more locally, sporadic, for a specific usage, as subordinated approach, ...)



Notes:

It is the nature of the challenge; the nature of the system, its purpose and usage that determines which type of engineering is required.

Two Major Engineering Approaches



Notes

This depends largely on the industry, on the type of system (core vs peripheral) and on the customer (example: operations vs marketing)

1. Mission-driven

- **How:** The Analyst receives missions to execute. These missions define the work of the Analyst, the work to be done by a project, and so on.
- **Advantage:** The work is more likely to be inline with the overall evolution and objectives of the company.
- **Drawbacks:**
 1. It relies on the knowledge and understanding of people other than the analyst. The mission may not be defined correctly.
 2. Weaknesses and flaws may remain undetected. Important corrections, improvements and opportunities may never be applied leaving the company inefficient and weak.

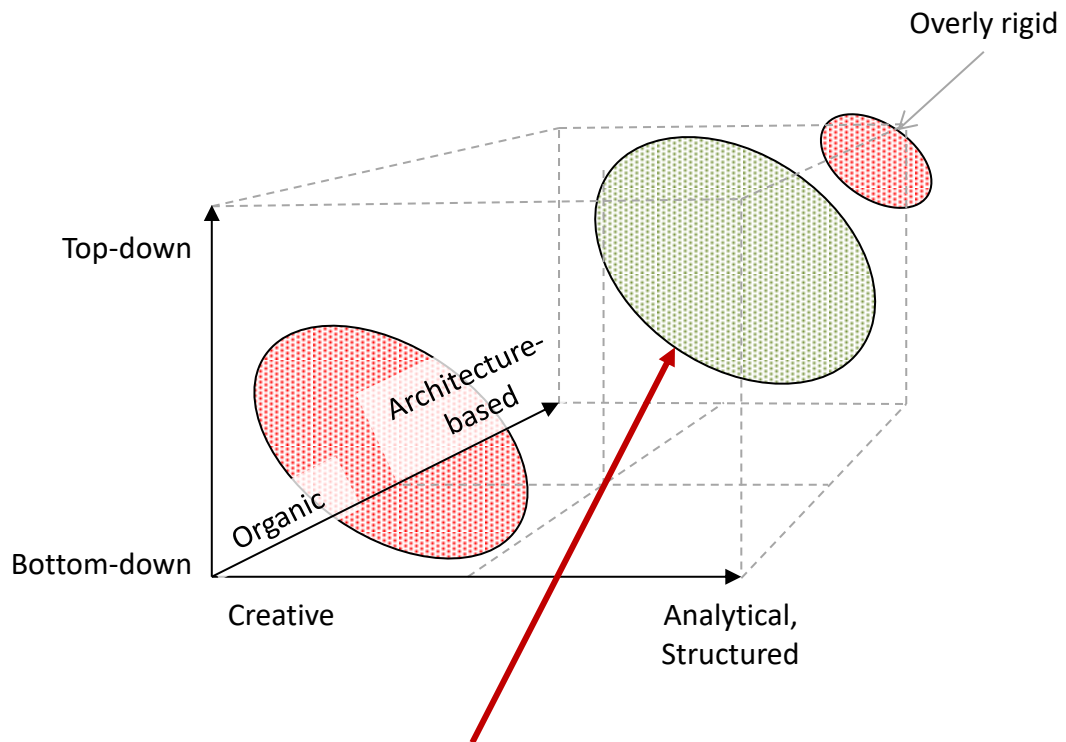
2. Based on Continuous Investigation (Continuous Analysis)

- **How:** The Analyst does continuously investigation to find issues, weaknesses, opportunities. The Analyst also study plans and architectures to see what impact they have on the present implementations and ongoing initiatives. He or she either has the authority to take initiatives of (some) changes. Or, he or she report detected issues and proposes initiatives and solutions.
- **Advantages:** The implemented systems can be improved in ways the management isn't aware of. Bottom-up improvements are possible. Systems and processes are better understood because they are under continue investigation.
- **Drawback:** Alignment with plans and priorities might be somewhat more difficult (but not impossible).

Notes:

Both approaches are indispensable. However, missions should be adjustable to the situation in order to get the most out of it.

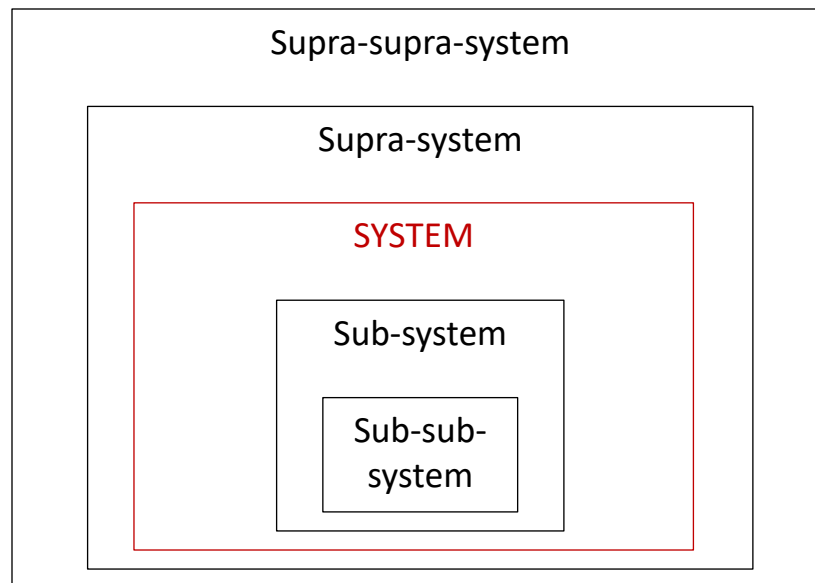
Different Types of Approaches in Man-made Larger Complex Systems Conception



Good balance for this type of systems.
The approaches should be in this area.

It is mainly top-down, analytical and architecture-based.
It make also possible to have creativity, some bottom-up and some organic growth.

This combines the strengths of all tendencies.



THINK “HIGH LEVEL”

The mind has to be focussed on and to be driven, inspired and guided **by the higher systems** and their environment.

- **Goals** are deduced from the goals of the higher systems (supra, supra-supra-, ...)
- **Requirements**, **Design** and **Functional Integration** are deduced from these higher systems, their goals and purposes and from their environments.
- **Technological Integration** takes the supra-system into account.

The supra-systems may be of a different nature than the system and can be heterogeneous. Same is true for the sub-systems and components.

A mind driven by and focussed only on change, on the features, on the system or on the solution to be delivered is likely to create inefficiencies and over time, through the multiple repetition, chaos.

Also important for methodology development

PRINCIPLE: SEPARATION OF CONCEPTUAL SOLUTION and CONCRETE SOLUTION

KEY PRINCIPLE: Conceptual thinking precedes and must be separated from concrete thinking.

The brain has first to think about how to solve a problem conceptually. Once this is done, it has to think how to realise it with technologies, with matter, with concrete pieces.

Process:

- 1) Define the conceptual solution
- 2) Define the technological solution, the technologies, the materials

Implication:

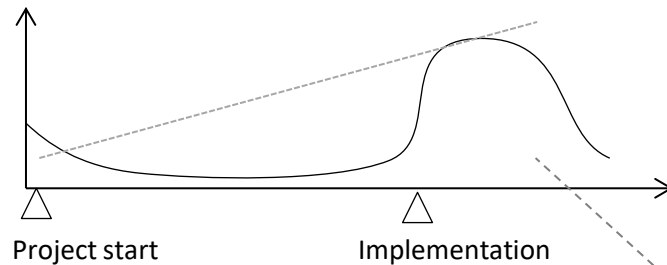
It is way better to separate the conceptual/functional/non-technological roles from technical/technological roles of architects, analysts, engineers.

Note:

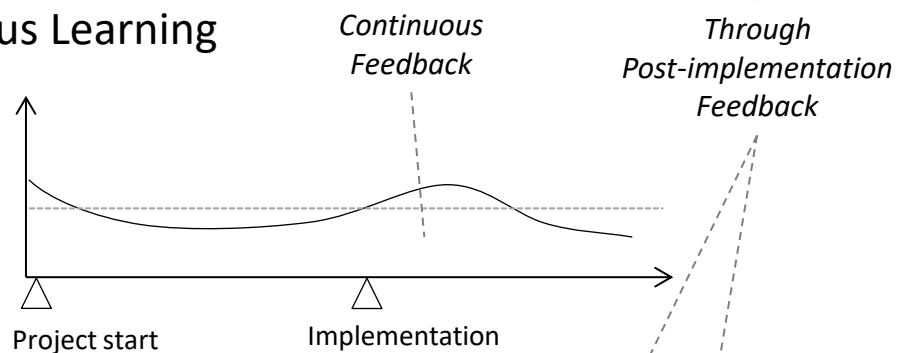
It is favourable for the conceptual thinker to know (or to have a good idea) of what technology can do and what it can't do.

The way business is ran and how information is used in the business activities and how it is organised, stored and processed in the company should be separated as well.

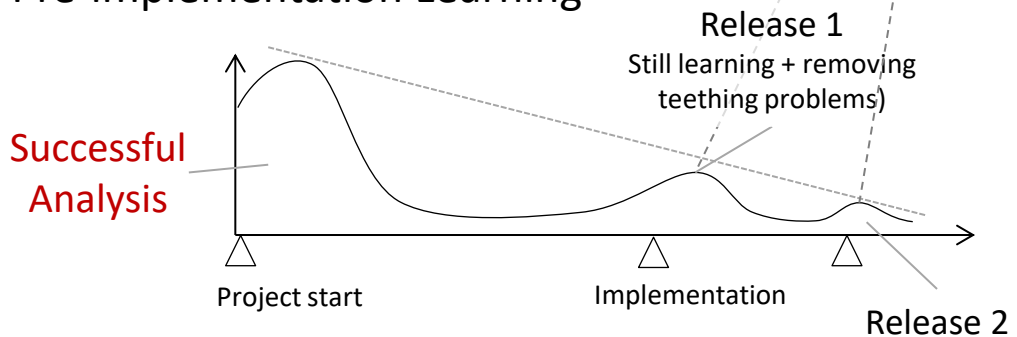
Post-implementation Learning



Continuous Learning



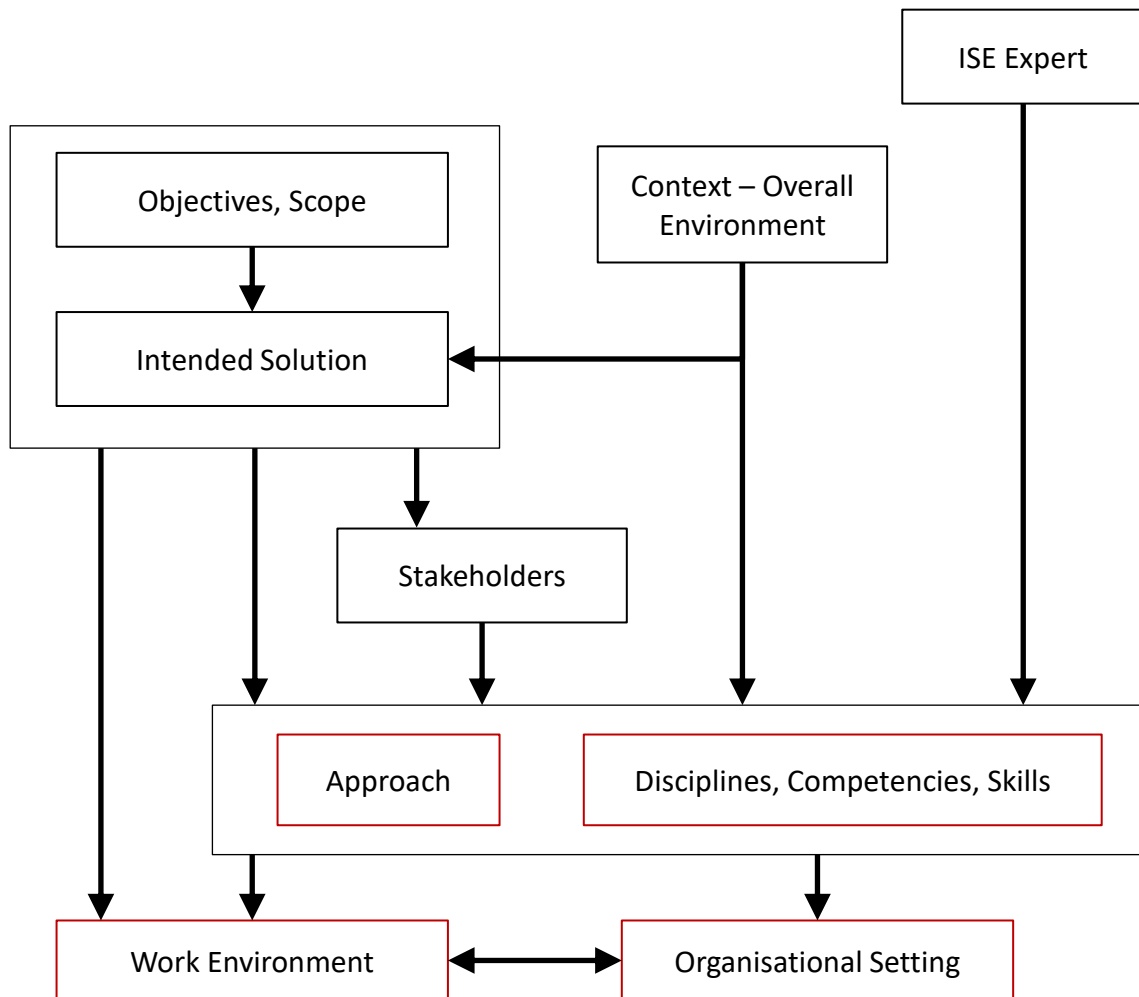
Pre-implementation Learning



Notes:

There is always some pre-implementation analysis (learning), continuous learning and post implementation learning.

Setting up for IS Development^{30/10/2018}



The objective, the scope, the intended solution (if present), the stakeholders, the global context and overall situation and the ISE experts and/or project team determine the approach required to achieve the objective or required to produce the intended product. They also define the required discipline, competencies and skills.

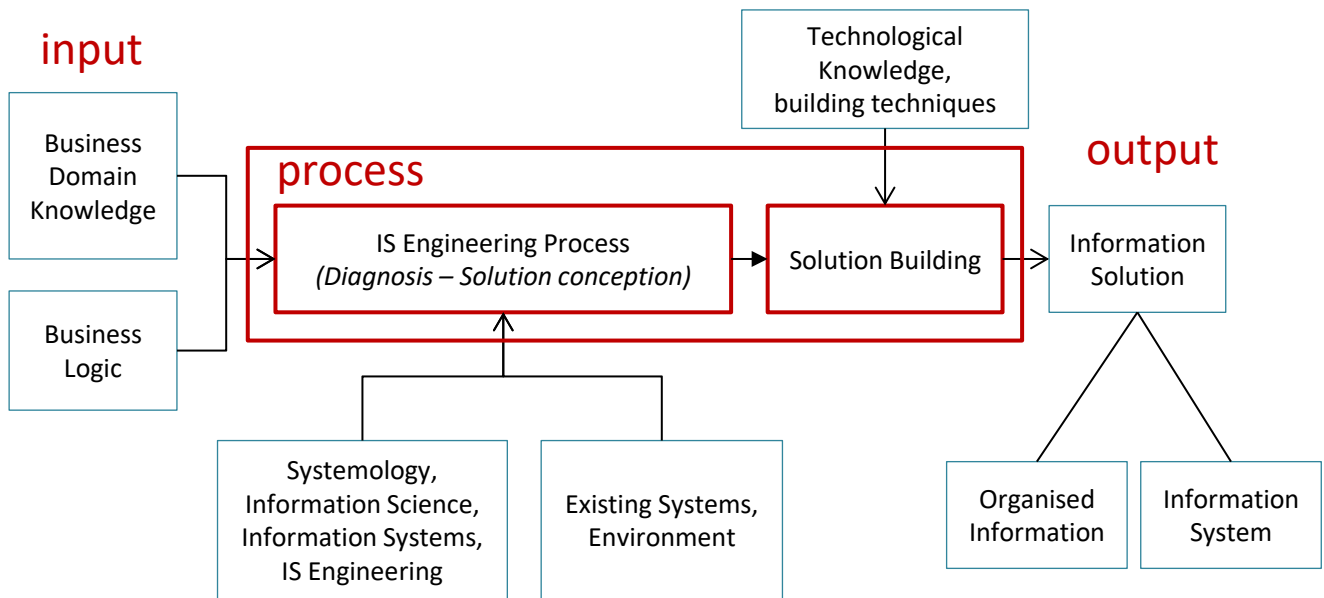
The approach defines the required work environment (room, space, tools, processes, schedules, ...) and the organisational setting (roles, responsibilities, collaboration, ...).

Performing jobs like sculpting a statue, writing a book, repairing cars, conceiving an advertisement or managing accounts or business information, trading, require a specific environment and work atmosphere.

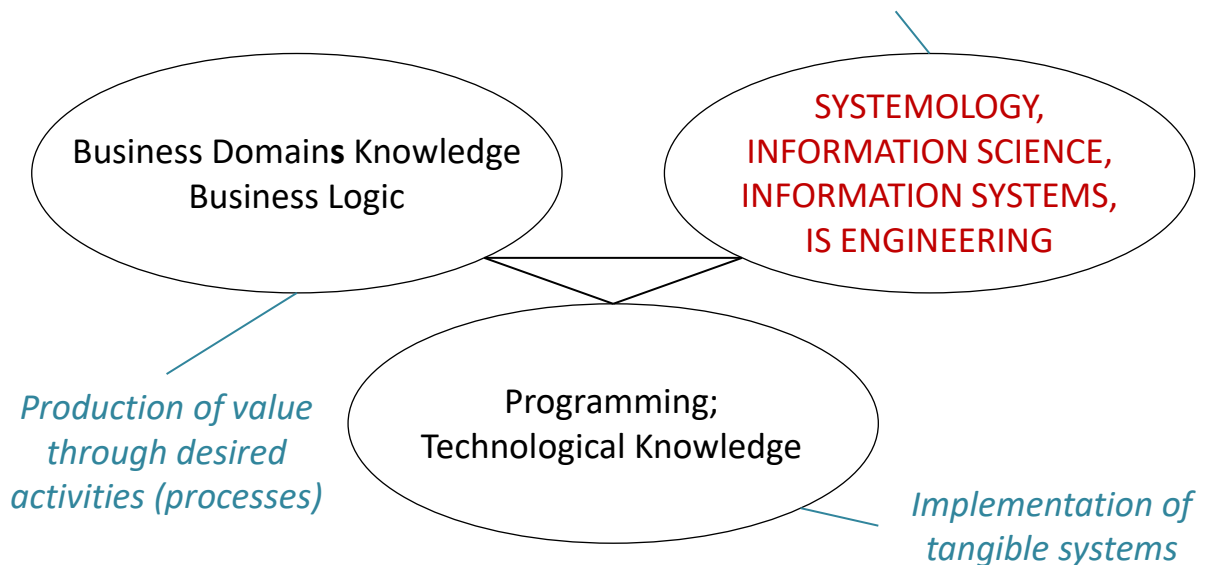
It is up to the project team to define what work environment is required. This is not an easy task. For now, a lot assumptions are made about this topic.

Engineering Process and Knowledge

30/10/2018



*Preservation and thriving of the system
Increases coherence, robustness, exploitability,
responsiveness, flexibility and evolvability
Reduces risks, cost of future evolution, ...*



Notes:

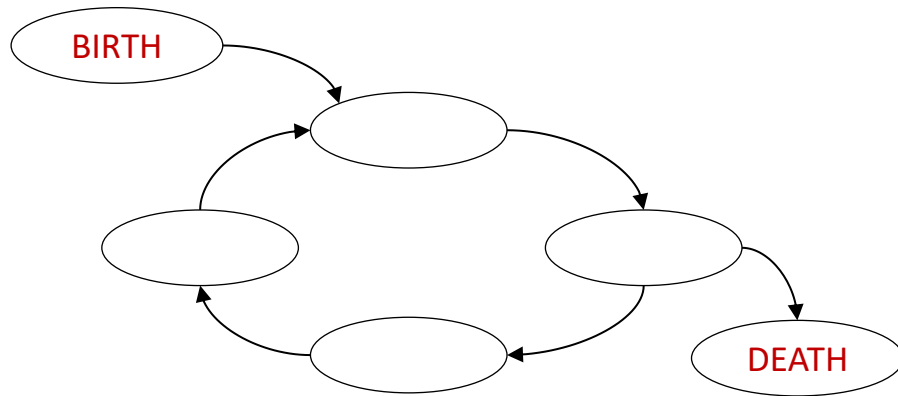
"Systemology" is the disciplinary field studying systems.

Building ≠ Solution Conception; Using / Executing ≠ Solution Conception

Life Cycle – Development Cycle

30/10/2018

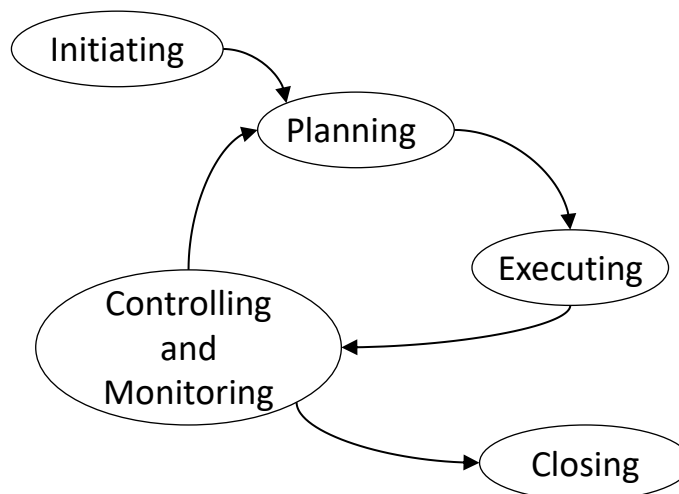
Life Cycle



Life Cycle

Describes the evolution, the phases and major events that may happen during the lifetime of an entity.

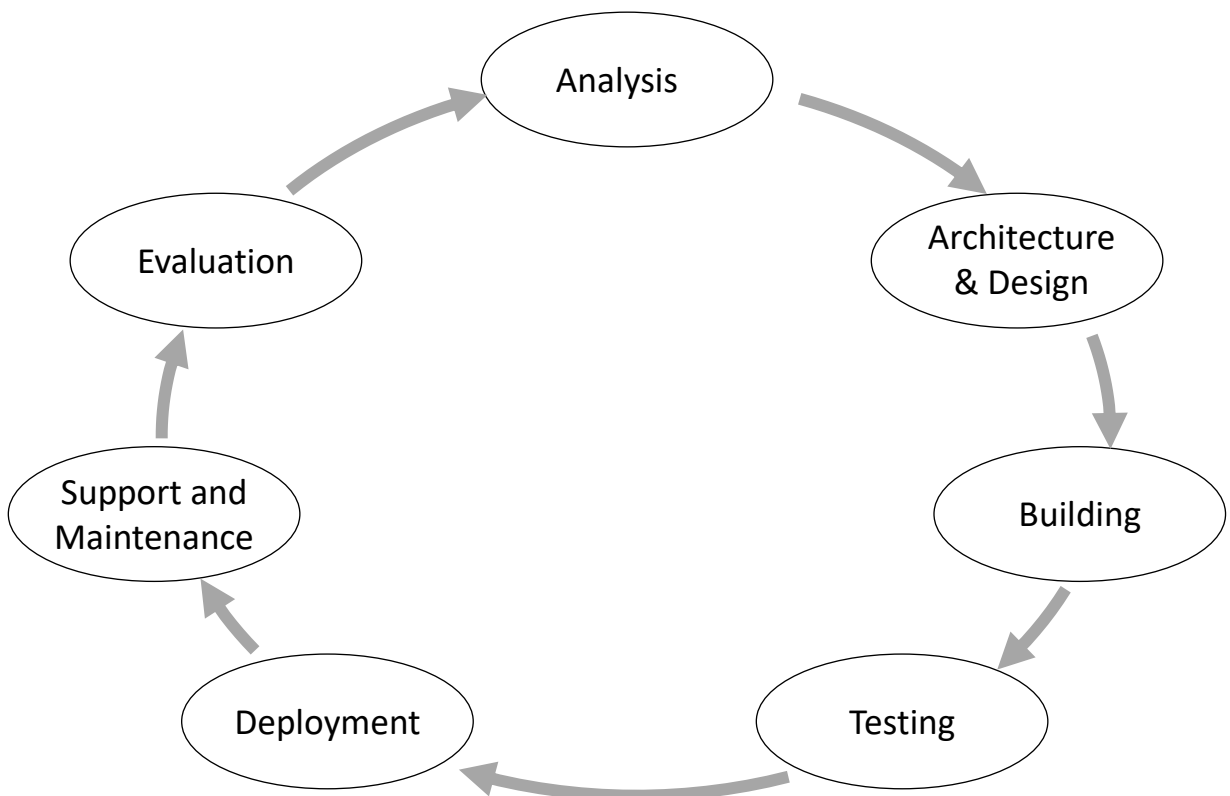
Project Life Cycle (example of PMI's PMBoK)



Development (Life) Cycle (example)

Development (Life) Cycle

Describes the phases, stages of a development of an entity.



SOLUTION - SYSTEM

- Not always a single best solution
(≠ No (/never a) single best solution)
Sometimes, but not always, there is a single best solution.
We may neither assume there is a single best solution or that there is no single best solution. Often we don't know and can't know.
- If there are several solutions then
 - Those solutions **differ** from each other
 - Have their own **specific strengths and weaknesses**
 - Some **are better** than other
- Try to find better solutions and select the best of the known solutions
- Continuing to find alternatives and improvements

SIMILARLY

PROCESS – APPROACH – METHODOLOGY - FRAMEWORK

- Not always a single best APPROACH
(≠ Never a single best way)
Sometimes, but not always, there is a single best way.
We may neither assume there is a single best way or that there is no single best way. Often we don't know and can't know.
- If there are several approaches then
 - Those approaches **differ** from each other
 - Have their own **specific strengths and weaknesses**
 - Some **are better** than other
- Try to find better approach and select the best of the known approaches
- Continuing to find alternatives and improvements

The design of systems must respect the laws and principles governing systems. Some of these laws and principles are essential to the survival and thriving of the system.

Information systems, as a specific type of systems, have to respect laws and principles governing a proper organisation, processing and use of information.

A system that doesn't fit into its environment, one that doesn't respect the nature and laws of its environment, can be destroyed by its environment or it can become irrelevant to this environment.

A system which doesn't respect laws and principles may also risk to die from within. A system dies or run into trouble, not so much because of external events, but much more because of its own bad design attacking the system from within. A good system should have a alerting mechanism warning that it is going to the end of its life.

Adapting such a system without understanding it fully is a very risky action. The system will always react and behave accordingly to its nature and logic.

The understanding of systems and information both as specific knowledge domains, are critical in the conception of information systems. This knowledge is neither to be found in the business domain knowledge nor in technological knowledge.

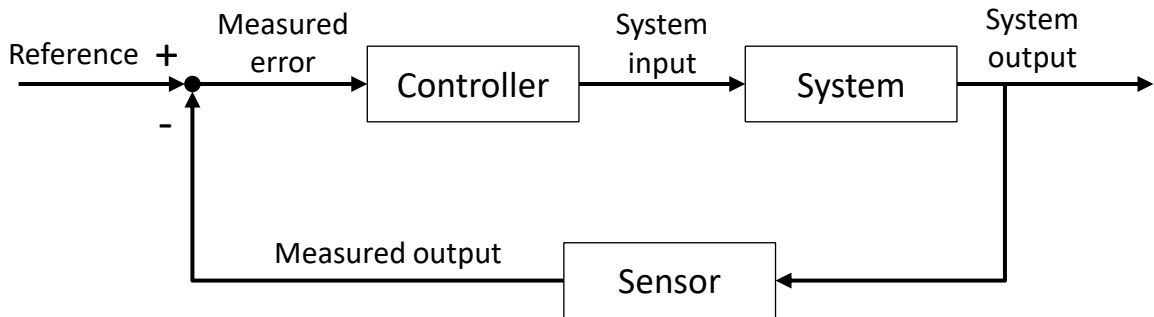
They are critical to determine how information is and can be used in the system and how it can serve the purpose of the system.

1. Implementing qualities by design vs adding it

- Not about features. Some can be added easily to a system
- Adding qualities like integration, quality, control, security, flexibility, connectivity, scalability, as a post-design (or late-design) activity is very hard. Rarely good results.
- Much better results (more powerful and lesser complex) when added from the beginning.

2. Change of purpose, function, role

- “Designing a screwdriver and trying to make a hammer from it.” Never as powerful as when it was designed from the ground up for that purpose, function and role.
- Like using a spreadsheet as DB

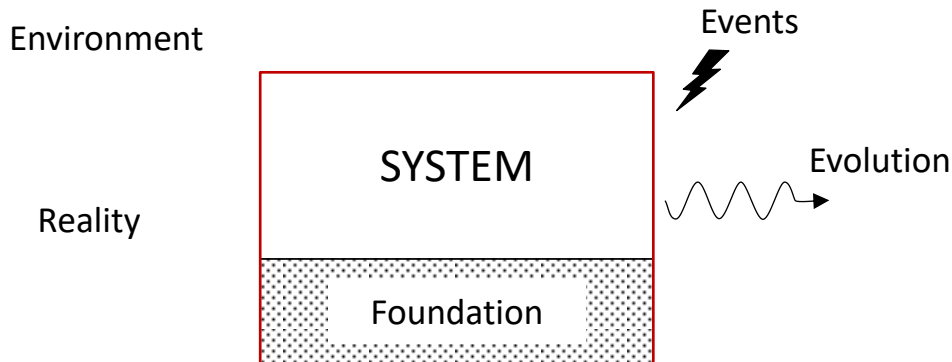


The concept of feedback loop to adjust the behaviour of a dynamic system.

Any dynamic system must have one or more feedback loops

- Company
- Project
- Development Methodology
- Information Systems
- Individual
- ...

This theory answers the question why we need control in dynamic systems.



GOOD SYSTEMS ENGINEERING

Foundation

Based on solid foundation. Systems built upon weak foundations are weak. Solid foundation: TERMS (based on right definitions; already a huge problem in IT), natural laws, first principles, based on understanding of nature of things, atomic elements, building blocks, ... Business demands are useful, but commonly, they are not a solid foundation.

Reality

Reflects reality. Critical to understand nature of things, relations between things, nature of these relations, abstractions of reality, ...

Fitness for Use (Quality)

Obvious

Environment

Respecting the rules of the environment and respecting the environment

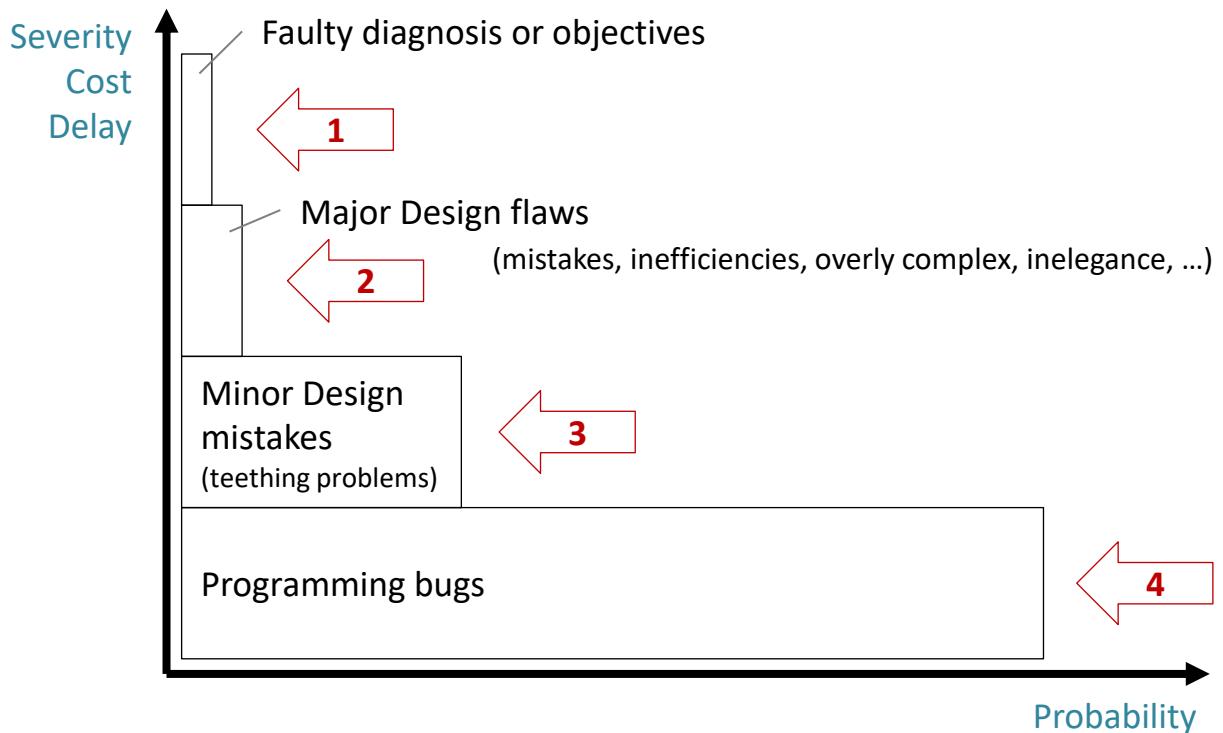
Events

Able to respond and to cope with events

Evolution

Able to evolve. Adaptability doesn't come by its own. It is engineered in the system. The development environment has continuously to capture the needs and opportunities to evolve.

These are areas/sources for requirements



← **Force aimed at reducing the probability (and the severity of impact) of flaws:**
Approach, Norms, Techniques, Skills, Resources and Time used to reduce the probability of flaws and impact. These 'forces' are not equal. Their 'size' and nature depends of what quality (or flaws) is acceptable for the project. This depends of the project, its environment, its stakeholders, ...
Example: a technique aimed at reducing design flaws will not eliminate a faulty diagnosis



Requires different perspectives, different measures, different methods and skills, different knowledge (thus often by different persons) ... at different levels, at different moments in de development lifecycle !

Notes:

These types of mistakes form actually a continuum. These categories are not strictly defined.

Major Design flaws implicate keeping the objective and (mainly) the scope, but need to review the architecture and overall design.

Faulty diagnosis, objective implicate the halt of the project. A brand new project has to be started.

A methodology and project process have to take this into account and implement a maximum of ways to prevent these fault to happen.

When talking about Integration – 3 Questions

- 1) **What** is integrated?
- 2) **In what** is it integrated?
- 3) Which **aspect** is integrated? From which perspective?

Integrated Information Flow

The most basic form of integration, through automation, is to add some technologies to ensure exports and imports of data. It limits the integration between the systems it communicates with. A more advanced form consists of a deliberate and purposely engineering of the entire set of information flows.

Technological Integration

Technological integration is realised by using same technologies as those already present in the company or by using technologies allowing interoperability among the existing systems and technologies.

Architectural Integration

Architectural Integration integrates the new system, components, interfaces and changes in the landscape of systems or in the system of systems. The thinking about integration happened deliberately during architecture activities.

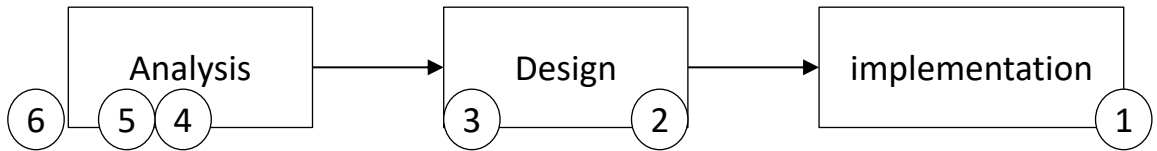
Functional Integration

Functional integration is based on domain knowledge, functions and purpose. Functionalities are logically organised. There is no duplication of features, functions, processes, ... All functions are well-thought.

Holistic Integration

Holistic integration combine all previous integrations. It also includes the integration in supra-systems and in their environments. All perspectives are taken into account and everything is aligned, including all the purposes and goals.

When do we start thinking about integration ?



No integration

System is in isolation. Information is exchanged manually.

1) Post-implementation (basic integration)

Once the system is operational: “How can we automate information exchange?” This may bring in new technologies (middleware or other technology allowing information exchange).

2) Post-design integration

After the solution itself has been designed, the concern is raised of the exchange of information.

3) Integration as part of design

During the conception of the solution, the question of sharing or exchanging information is raised. This may influence the choice of technologies.

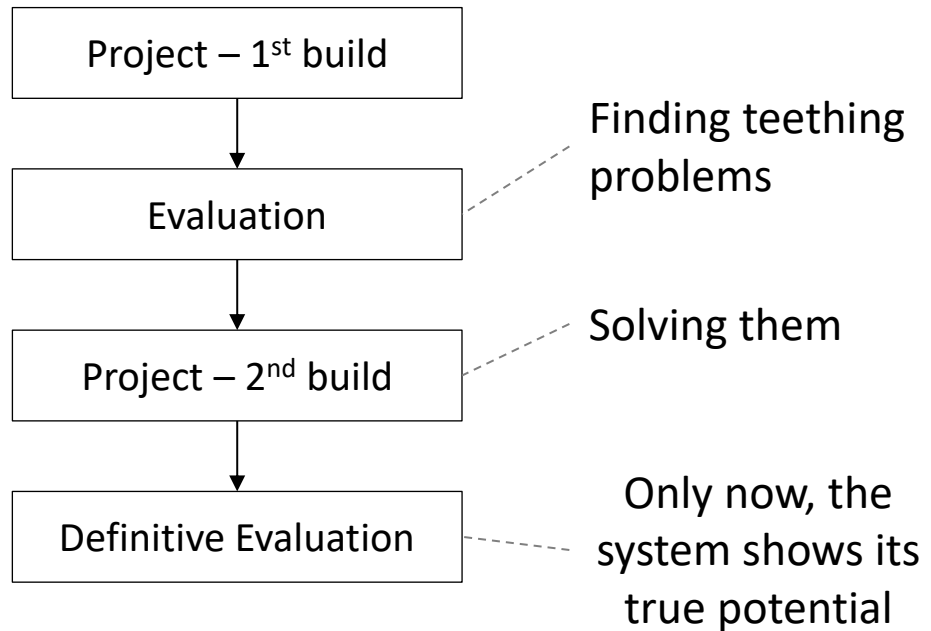
4) Architectural integration

5) Functional integration

Architectural and functional integration happen essentially during the analysis phase.

6) Holistic Integration

Holistic integration happens during the whole engineering process. However, it starts with the definition of the objective, the mission and the scope. It has to start as early as possible.

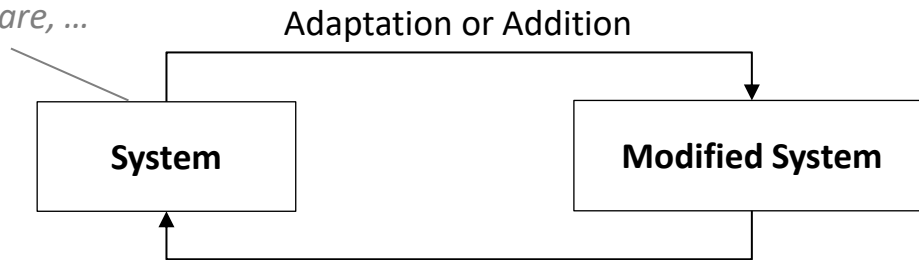


Teething problems are problems due to a lack of insight, additional information surfaced, imperfect solution logic, unexpected system's behaviour or unexpected ways of using the system, missing features, and so on. It is better to make a difference between teething problems and programming bugs. Teething problems require additional analysis or design activities.

Changes Eroding Systems

30/10/2018

*Company,
software, ...*



Eroding System

- Oversimplification
- Overly complex
- Unnecessary elements
- Inelegant internal organisation
- Lack of clarity

Per iteration:

Increased size
Increased complexity



Decreased Manageability
Decreased Adaptability



For future changes:

Increased risks
Increased time and effort (→ paralysis)
Increased cost

Benefit/cost (ROI) tend to decrease
Tendency to entropy

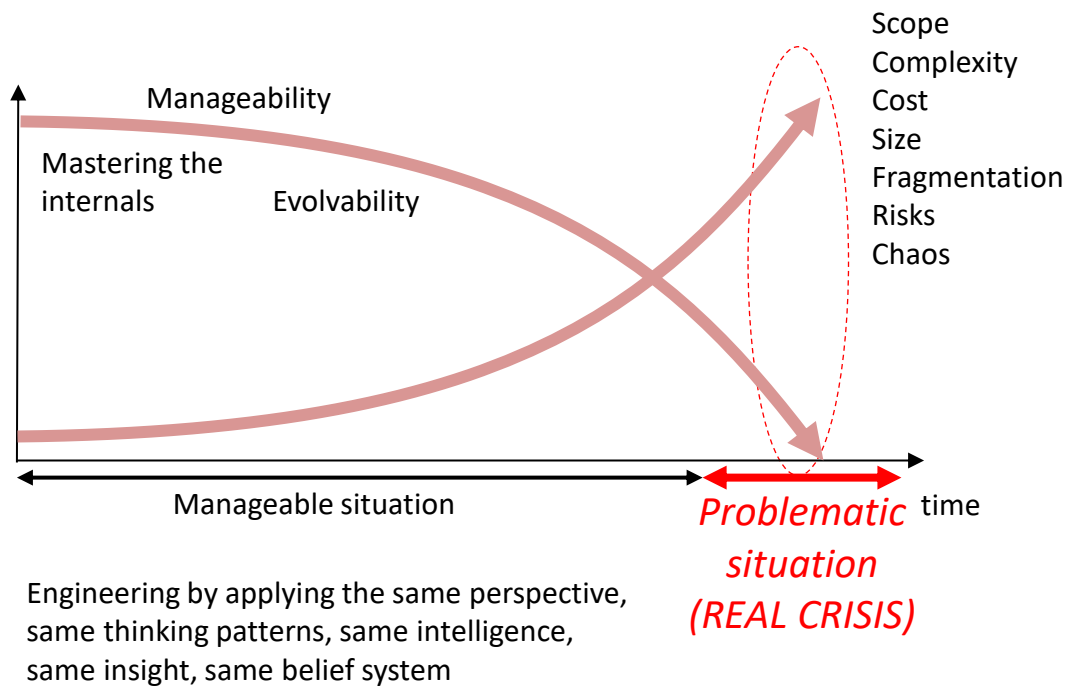
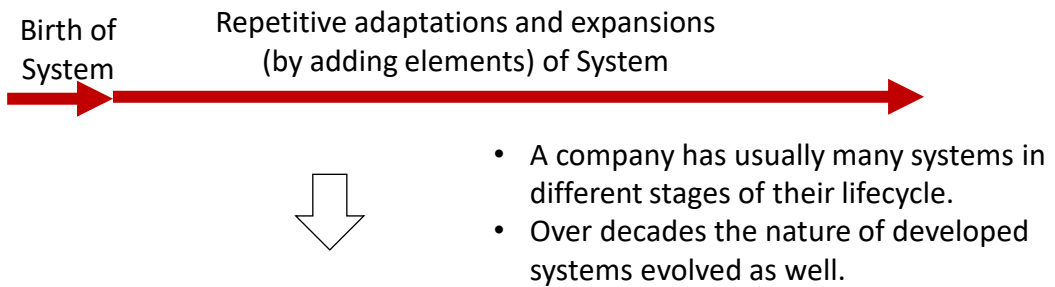
➡ **CRISIS**

Common Causes:

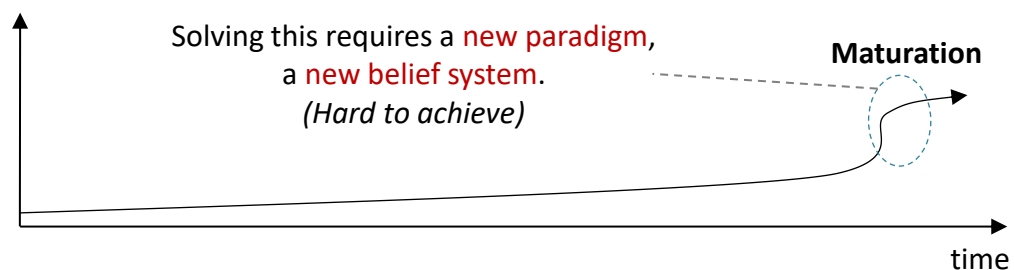
- Time pressure
- Lower norms ("if it works, it is good enough" ☠)
- Unawareness of decreasing quality of the system
- Adaptions are made with the same perception, same assumptions, same priorities, same skills, same level of thinking, same beliefs and assumptions.

Evolving Systems and Crisis

30/10/2018



Time for reengineering, paradigm shift or other radical measures and shift



Writing vs Adaptation

15/04/2019

Assumption: Adaptation is faster than the first writing of source code or than recoding from scratch

Adapting source code	more time	First writing
Adapting source code	approx. same time	First writing
Adapting source code	lesser time	First writing

Depends of change, source code organisation, spread, ...

Implications: Softw. Dev. Approach, Estimations

Similarly, bug correction can be hard to estimate.

A bug may require small change in code, or larger or many changes of the code.

Depends also of the type of bugs:

- Programming bug
- Design bug
- Analysis bug

Synonyms

- trial and failure
- trial and practice
- generate and test
- guess and check

Characteristics

- Every trial teaches something
- No upfront knowledge or insight can be acquired
- Plenty of time and resources
- Attempts can be done safely
- Limited number of cases to test or some options have better chances of succeeding
- Only 'a solution' is required, and no deeper understanding, ...

Trade-off between 1) number of possibilities; 2) effort, duration and cost of each attempt and 3) importance of the problem being solved.

Weaknesses and risks

- No alternatives or optimal solution. Aim to find "a solution", not all or "the best solution".
- Delivers single solution to single problem. No attempt is made to generalise the solution.
- Does not do a diagnosis. No cause/problem identification. Assumes the diagnosis is right.
- A solution may appear after a few attempts, only after many attempts, or not at all.
- May be time consuming (for example, due to many attempts)
- May consume a lot of resources
- May turn out to be costly
- Making an error may be very detrimental. Only one or very few attempts are possible (think of surgery)
- Doesn't search to explain "why" it works and no attempt is made to get this understanding. Maintaining superficial insight or ignorance. (Learning lessons?; Control and management of the broader whole?; Evolvability?)
- Found solution can be hard to optimise
- Quality? Reliability? Strength? Scalability? Behaviour in variable circumstances?

- BizBoK
- OMBoK
- EABoK
- TOGAF, Zachman, TEAF, FEAF, PERA, GERAM, ...
- BABoK
- DAMA
- CobIT
- ITIL
- PMBoK
- Prince2
- SE BoK
- DA BoK

Knowing bad practices help to recognise and to avoid them

- Throw it over the wall (limited collaboration)
- Accept anything at face value.
- Don't question earlier findings, decisions, ...
- Don't explain the higher goals, intentions
- Retain information (filtering)
- Freeze everything made or validated by the business stakeholders
- Apply principles, rules, methods because they are 'prescribed' or because of habits
- Build upon understanding of business demand and/or requirements as only insight.
- Be positive. Be a "YES"-(wo-)man. Everything is fine. Everything is possible.
- Do only the work you prefer. Work you don't like is not necessary anyway.
- Perform unnecessary work
- Ignore the purpose, reason, value, right usage of concepts, tools, ...
- Do not collaborate or collaborate as little as possible
- Decisions can be taken without knowledge and understanding. Either the decisions is pretended to be right regardless of the cost and consequences, or it doesn't matter because any decision can be changed anyway later on.
- Don't do what is necessary. Instead, do whatever is asked hoping to meet the expectations
- Do not disagree. Avoid disagreeing with the chief, with the group or with the majority.
- Let others take decisions and responsibilities, that should be yours.
- Don't review made decisions. Don't ever backtrack.
- Be reactive and follow.
- Don't doubt, even if the world is complex, dynamic and unpredictable.
- Pretend to know everything.

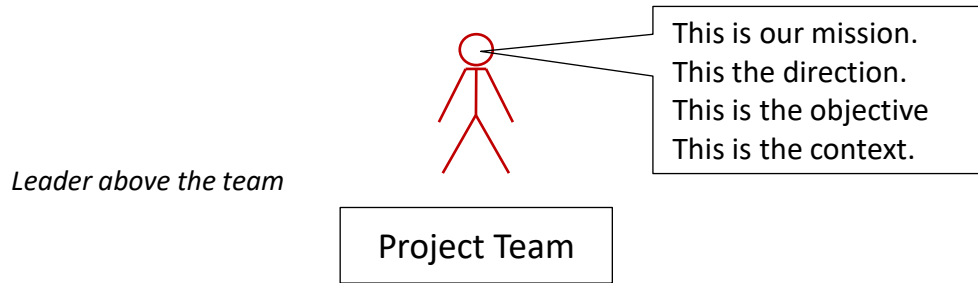
Note:

Regardless of the project approach, these practices are neither prescribed nor advised.



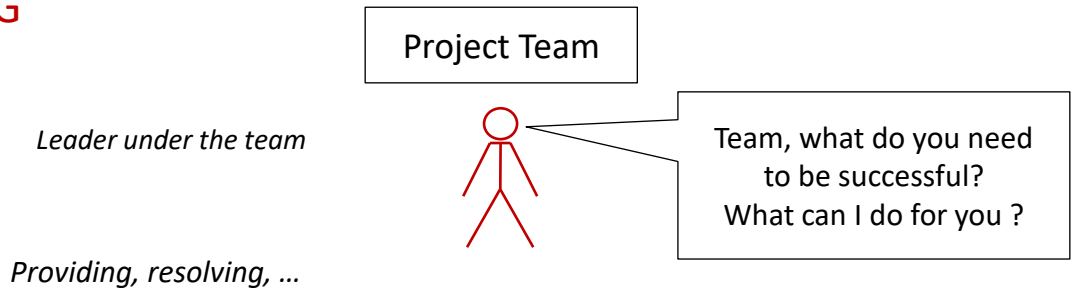
MANAGEMENT

LEADING

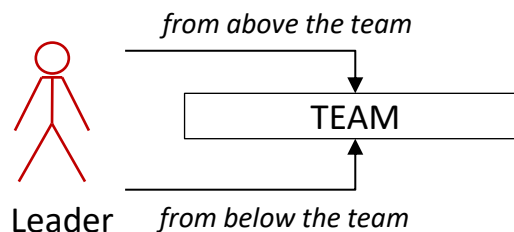


Guiding by setting the objective, Organising, Guiding by defining the process, ...

SERVING



PROTECTING (*)



*: interferences, conflicts, pressure, risks, disturbances, politics, ...
Management ≠ Leadership



WORK ENVIRONMENT

Work of Architects, Analysts and Engineers consists mainly of learning, analytical thinking, critical thinking, logical thinking, systemic thinking and creative thinking.

The work environment greatly influence this intellectual work. If it is not suitable, it will hinder it leading to disappointing results. Or, often people won't even be aware of the missed opportunities and benefits that could have been obtained.

Some **principles** and **ideas**:

- Thinking doesn't happen linearly.
- Creativity is spontaneous. It doesn't work on command. It can't be managed. But circumstances may be more favourable to creativity and techniques may support it.
- Creativity leads to innovation. Everything that hinders the mind hinders innovation.
- Stress, fears, uncertainties and other negative feelings and emotions hinder the mind. Drive out fear.
- Conflicts, uncertainty, firefighting, urgencies, pressures and chaotic environments should be avoided.
- Eliminate elements that distract.
- Eliminate or avoid interruptions.
- Administrative workers can work easily perform an amount of work in 1 hour or lesser. Therefore, a common day in an administrative environment is divided into slices of 1 hour. Architects, Analysts and Engineers may need 15 minutes to an hour or more simply to get into the matter. Depending on the work, they may really need to be able to work without interruption for half of a day, a working day or even more.
- Ideas may come while walking through a park, when watching through the window, when sitting on the toilet, when being alone in a calm, quiet and secure place.
- Very flexible hours.



ORGANISATION



A Manager is not an Engineer

- A manager manages
- An engineer engineers

Managing and engineering

- have to different goals
- require two completely different mind-sets
- and require different ways of thinking

Exceptions: Some engineers may become managers and some managers have engineering degrees.

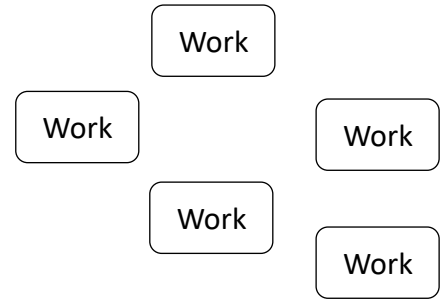
Conceiving a solution requires understanding

- how to maximally exploit information
- what is possible, difficult and impossible in IT
- what is necessary for an information system to survive and to thrive
- what undermines an information solution
- what architecture fits the endeavour and environment
- understanding how to preserve or increase the value of information
- understanding information risks
- etc.

Who possesses this insight? Or, who is the most likely to possess this insight?

Group

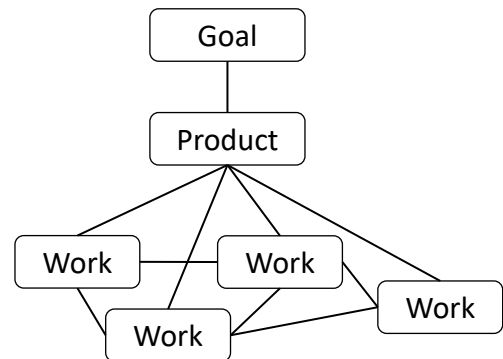
- Unrelated work
- No collaboration and little interactions are required



→ No team is required. A group is fine

Team

- Working towards a same goal
- Working on a same product
- Work is related
- Each other's disciplines and qualities help or are necessary to achieve something in common

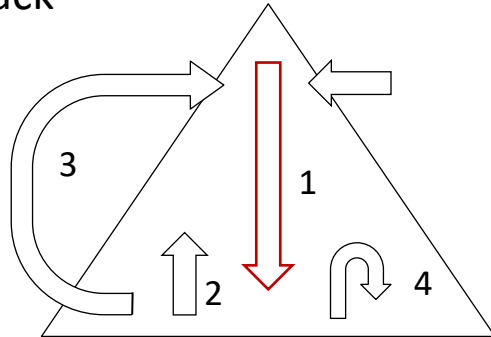


→ A TEAM is required

Creative with hierarchy – a few ideas

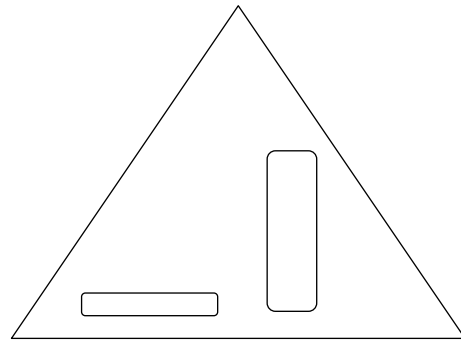
Top-Down – Bottom-Up Feedback

1. Top-Down
2. Bottom-up
3. Bottom to high top-down
4. Bottom to low top-down

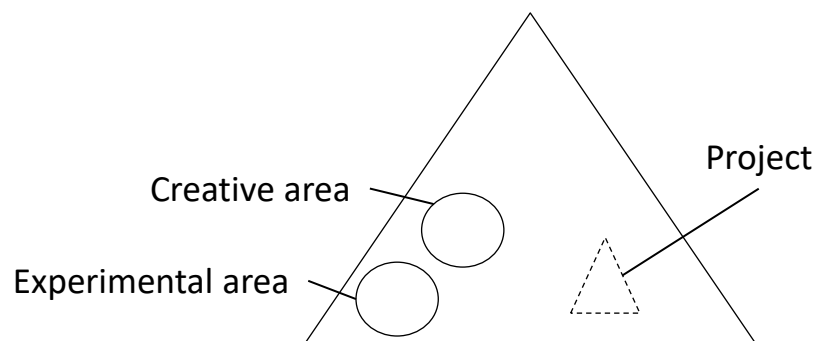


Cross-border collaboration

- Horizontal and/or vertical
- Persistent, temporary or ad hoc




Area's, Zones



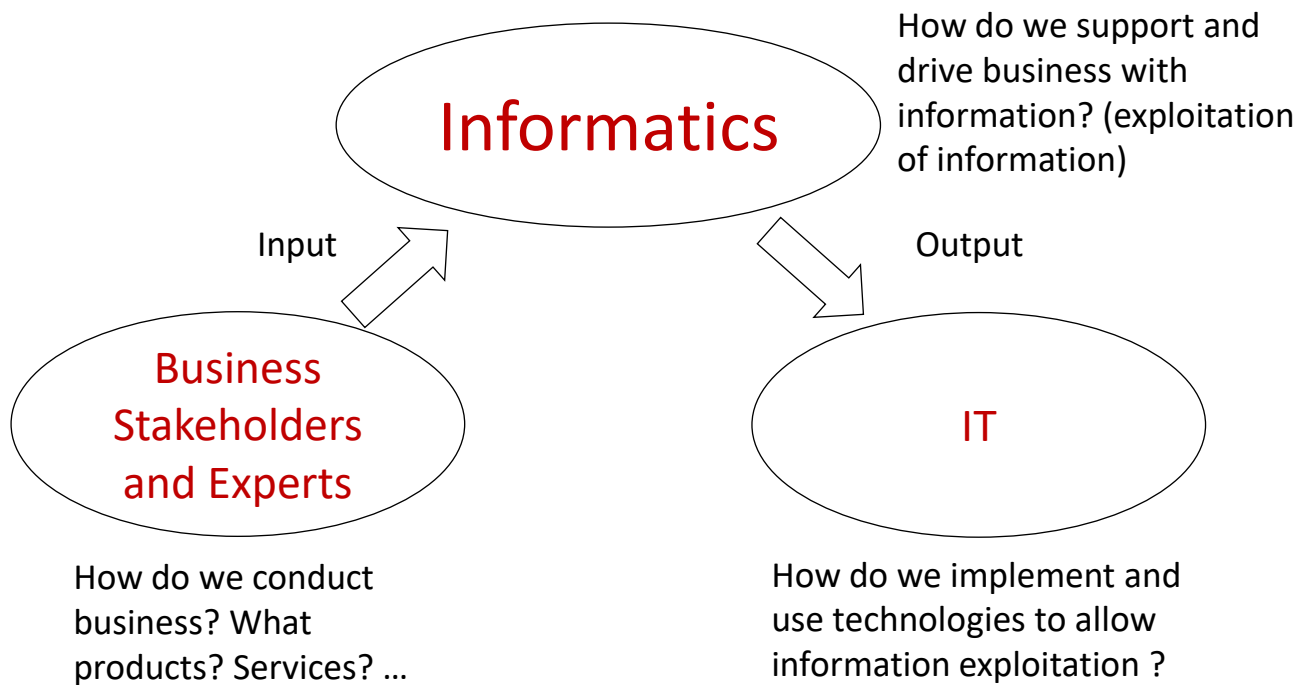
Notes:

Can be vertical, horizontal or both. Permanent, temporary or ad hoc, guided or free, measured or not, controlled or not, ...

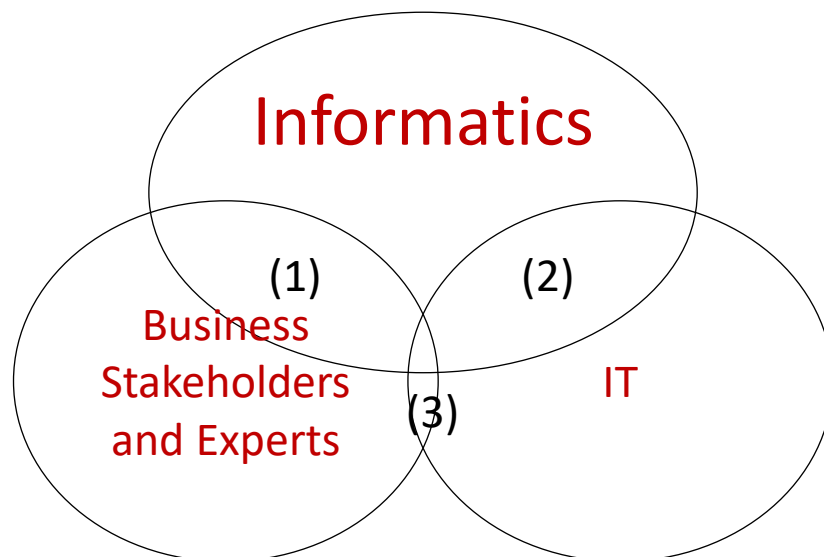


ROLES & COLLABORATION

Basic Information Flow



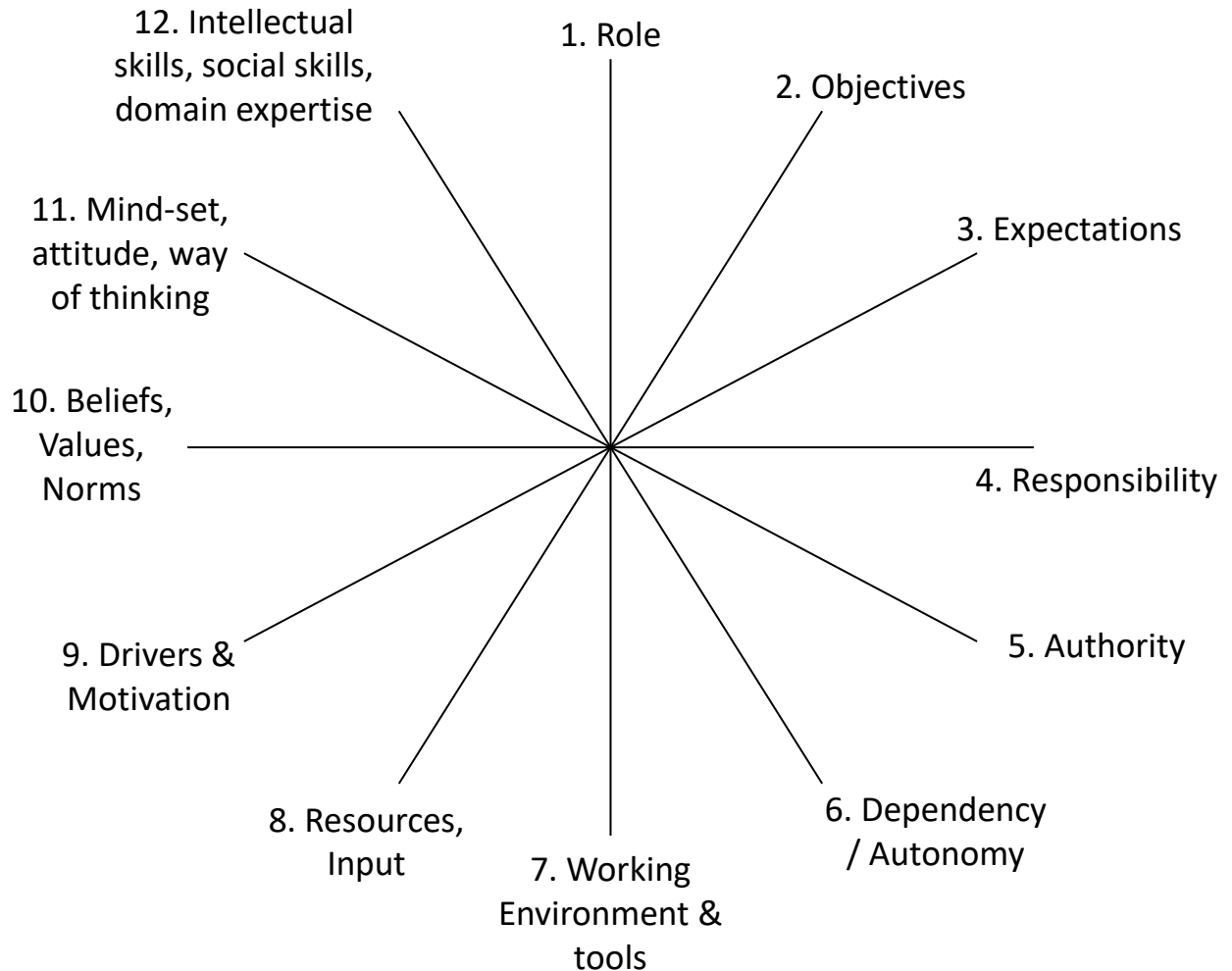
Suggested Collaboration Model



Notes:

Collaboration (1) and (2) are extensive. Collaboration between (3) is useful, though somewhat lesser (when business-informatics issues are settled: reviews, wrap-ups, feedbacks, ...).

Operational Role Alignment Model



These factors have to be aligned in order to successfully perform a work. They are operational prerequisites for a job.

More alignments:

Cultural alignment

Alignment with "the boss"

Alignment with the group

Alignment of the evaluation criteria, evaluation system, reward system, ...

KEY PRINCIPLE

Conceptual, functional, logical roles
and
the technical and technological roles
should be assigned and performed
by different persons

The brain has to be able to focus freely on the global solution, on the big picture, on the higher goals, on the purpose.

The attention of the brain is easily directed on what technologies to use and how to use them, on how to build the product, on what to do now, on the short term. These are very practical matters. When the brain is busy with these matters, the attention is diverted from the overall conceptual and global thinking, higher goals and long term. Great opportunity to get real or much better solutions can be missed. The risk of building non-solutions increases.

Also important for innovation !!

Positions, functions, job titles, job definitions should either concern conceptual work or technical work, but not both.

Collaboration is the **Key** to **Project Success**

1. Clear Common Goals
2. Motivation: Wanting the project to succeed
3. Mutual Understanding (communication, listening)
4. Availability / Time
5. Agreements
6. Knowing and respecting the **roles** (responsibility, authority, action)
7. Respecting the principles of the different **disciplines**

The project's success and the created business value
depend of
every stakeholder and of every **project team member** !!

Note:

Not all definitions of 'stakeholder' include the project manager and project team members. Frequently the term is used in statements to indicate business stakeholders, clients, ... but not the project manager and project team. This definition is more practical.

Hindrances in Collaboration (1)

30/10/2018

Consumer IT	Corporate IT
Oriented towards consumer (selling and end-user experience)	Oriented to groups of end-users and serves the company's interests
Solves personal needs, responds to personal tastes, personal choice	Solves various needs, primarily company's needs, within a broader heterogeneous system of systems
Single product (except cloud, social media,...)	Set of complex interconnected systems part of the enterprise around an architecture, using an infrastructure
Focus on end-user features, appearance, ease of use, preferences and taste, experience, fun	Focus on value, processes, services, ROI, ...
Same product developed for many consumers	Developed for limited number of end-users
Usage: individually, accordingly to own taste	Usage: used by community of people, with different roles, but as part of a same organisation with a same objective. Shared, regulated, standardised
Easy, fast, cheap, flexible & powerful	Slow to build, expensive, more rigid, powerful in a different way
Fuzzy, approximation, guessing end-user's wishes	Abstract, detailed, consistent, complete, clear
A problem usually impacts individuals	Problem has impacts on the company
Easy to acquire and replace (buy and dispose)	Difficult to acquire. Requires customisation, tests, integration, courses, maintenance, security, recovery, ...
lifespan: 1 to 5 years	Has to evolve – lifespan: sometimes decades
More use of unstructured data and multimedia	Uses preferably structured data

Consumer IT are the IT products and services developed for personal consumption (usage). These products and services can be bought, downloaded or online accessed by consumers.

One can not confuse Consumer IT and Corporate IT (enterprise IT). The users and goals are different. There are different expectations, priorities and selection criteria. It is not wise to deal with Corporate IT as if it was Consumer IT (not black & white. It is a matter of balance)

IT Literacy

What do end-users/consumers capture of IT ?

- Philosophy: Intuitive, explore and discover, guessing, trial & error, experiencing, ad hoc
- Ideas: Quick and easy acquisition of technologies, “The sky is the limit”, “One click”
- Plenty of software: office automation, multimedia, games, tools, ...
- Software characteristics: attractive, intuitive, fancy graphical user interface, flexible software, rich features
- Installation: easy, automatic configuration, Plug-n-play, immediately usable
- Usage: Drag-n-drop, Cut-n-paste, Copy-Paste-Delete, Do-redo/ undo, wizards, templates, WYSIWYG, install-uninstall, problem? Ctrl-Alt-Del-solutions, ...
- Data dissemination: extensive use of unstructured data and individual data packets (documents), ad hoc copying and data transfers, seamless connections, invisible data exchanges
- Development: presentations, social media, blog, macros, websites, small software
- Other: IT magazines; unrealistic IT in Movies;
- Assumptions: Consumer IT = similar to Corporate IT; User’s Knowledge = IT Literacy; ...

**THIS PICTURE REFLECTS A VERY DISTORTED PICTURE
OF CORPORATE IT**

Not a good base for expectations, decision making, collaboration, ...

Hindrances in Collaboration (3)

30/10/2018

Difference Between Human World and IT World

- Specific Capabilities in Information Processing
- Capabilities of doing what humans can't do and vice versa
- Ability to perform a huge amount of work very quickly
- Omnipresence in the entire company
- Entanglement with the human organisation
- Speed of execution (huge numbers, small differences may have a huge impact)
- Connectedness, Interoperability
- Largest part is invisible to human's world (only computers, GUI, appliances, cables, printers, ...)
- Abstraction of virtual concepts and daily life elements
- Degree of formalism, precision of language and of instructions (process description) versus interpretation
- Range of details from very high-level to extremely detailed and precise
- Complexity (real complexity is hidden from the end-user by the GUI)
- Required consistency, repeatability
- Need to foresee 'everything' → number of individual cases
- Higher norms are required
- Required level of thinking/planning ahead
- Effort required and speed of change – estimations !!
- Effects of changes
- Possible impacts (hidden parts of logic, connectedness, level of details, entanglement, ...)
- Internal tendencies creating risks or leading to chaos
-

Hard to imagine, particularly for non-IT people

Gaps of ignorance is closed with assumptions

Lack of ability to detect the opportunities, to define better solutions, to define the right way of working, right norms, limitations, implications, to anticipate risks, ...

Role of the Business Community in Information Initiatives

INPUT of the business experts, business stakeholders

1. The existing situation (how does the business works now)
2. Objectives, complaints, limitations, expectations, intentions, plans, ... related to business results, work and information
3. Business Expertise, Business domain Knowledge
4. Answers to Analyst's requests
5. Disagreement or confirmation of the proposed solution

What they must not provide as a given:

- Identification, definition and description of the (information) problem
- The (information) solution they want

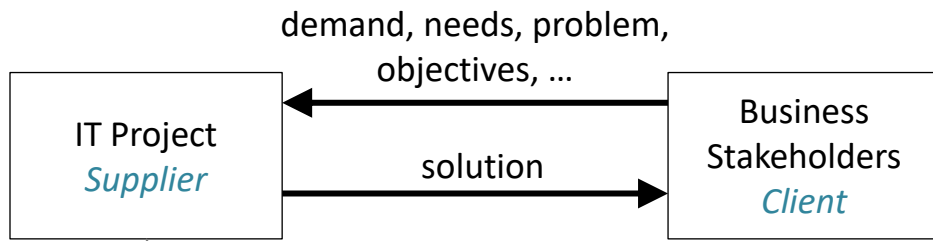
But business specialists may suggest it as a possibility to the Analyst.

Why not? **This** requires "Systems Analysis" skills.

It is related to, applied on the business domain, but it is not business domain.

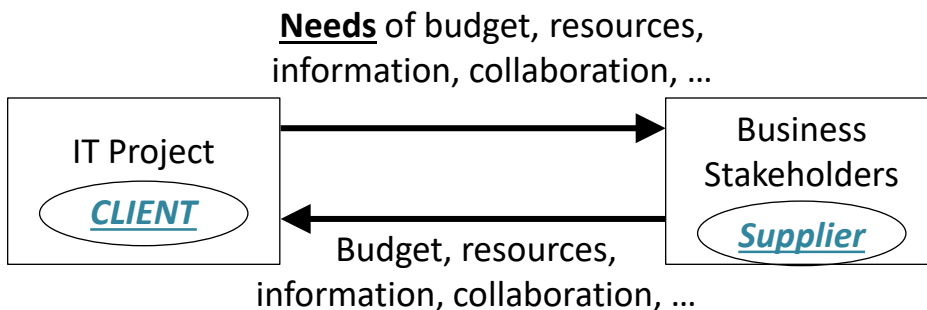
Dual Client-Supplier Relationship

30/10/2018



The mission creates needs on the project side: resources, time, information, budget, tools, work environment, collaboration, ...

So, to start the project ...



IT experts need to identify their needs: time, budget, resource, knowledge, collaboration, work environment, tools, time schedules, level of autonomy, ... to be able to work in ideal conditions. They become a 'client' as well.

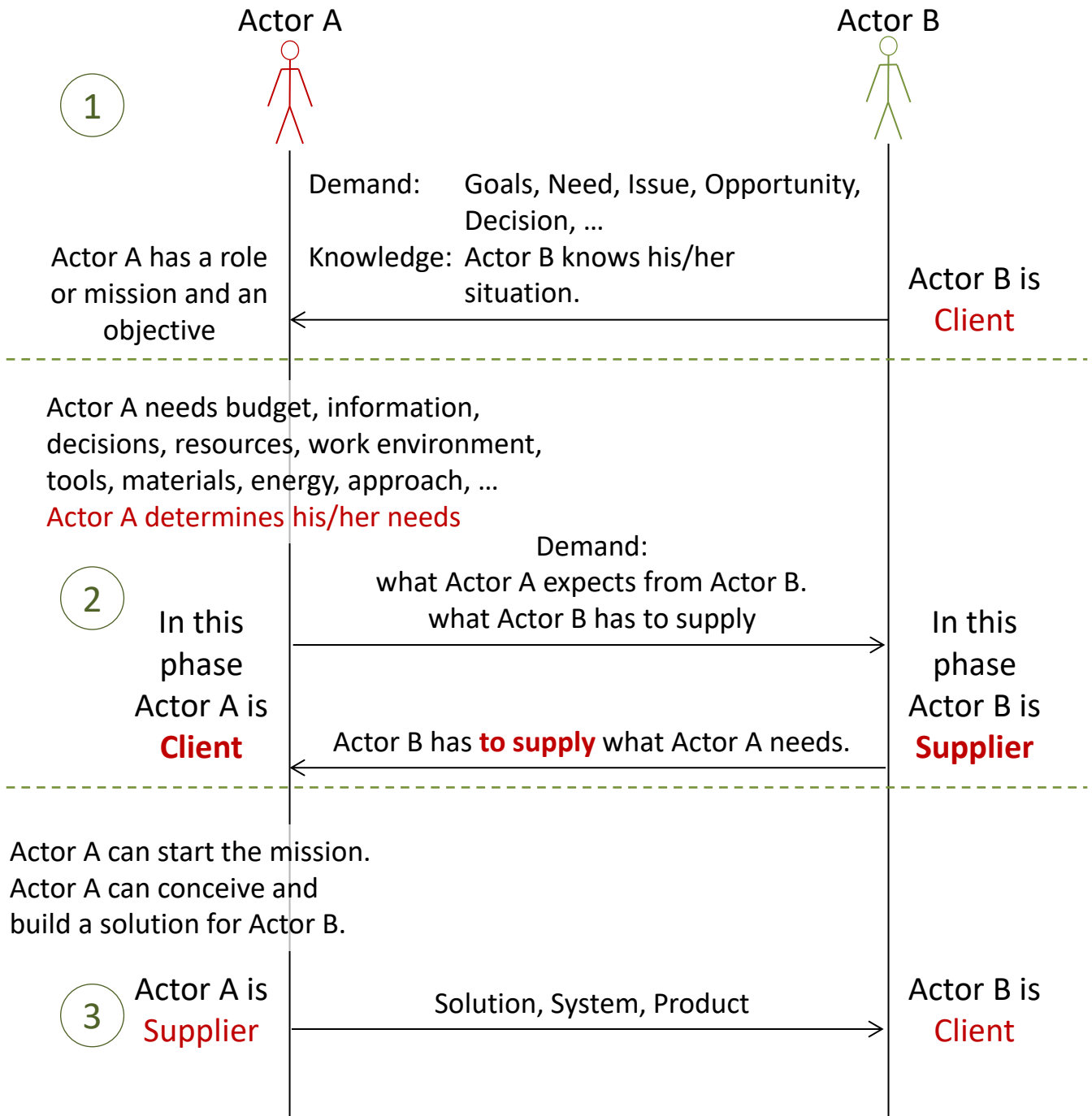
If this is ignored, they can't do their job properly and the final client can't be served well

Note:

The Business Stakeholders are sources of expertise, "guiders" or parts of the system. The main client is (from a systems perspective) the company (organisation) and, from an activity-perspective, the final external client is the real client.

Client – Supplier Relation

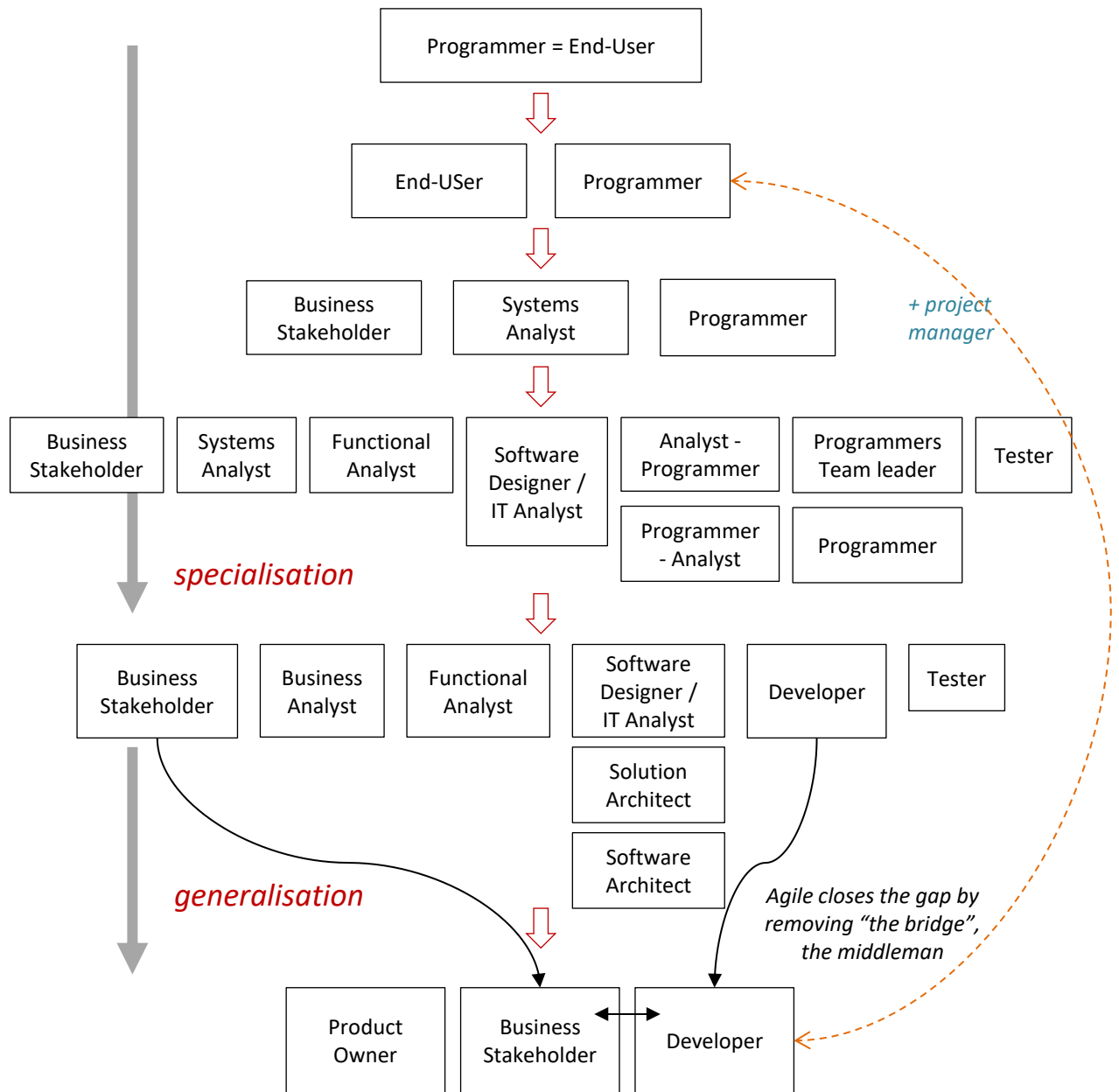
18/12/2018



- If (1) fails, then (2) and (3) will fail as well. If (2) fails, then (3) will also fail.
- (2) Gets often not the right attention. Frequently, the Actor B decides of what Actor A will need and what not, and when (s)he will need it. This is a recipe for failure.

Evolution of Roles in SW Dev. Projects

30/10/2018



Separation and specialisations of functions (roles) emerged as from the necessity to cope with complexity, scope, span, diversity and scale in the evolving software development initiatives.

This is an approximate evolution of the appearance of different functions. Each organisation has its own functions and function titles.

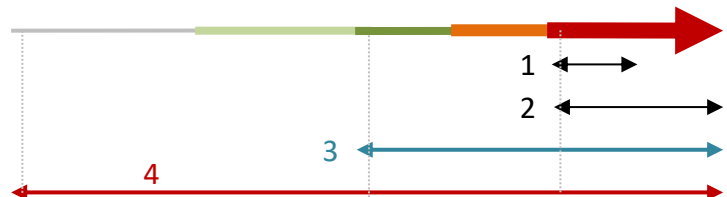
Functions like process analyst, information analyst, information architect, solution architect, application architect, IT architect, enterprise architect and many others are not represented.

Application of Informatics in a Traditional Approach

14/02/2020

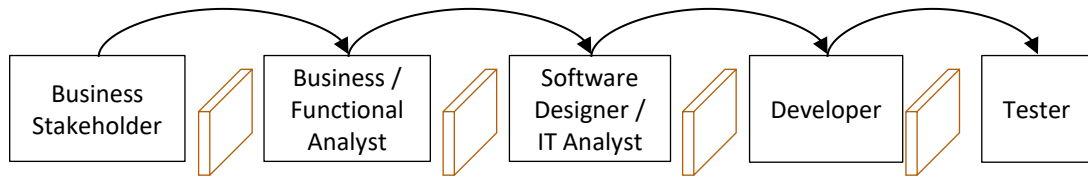


1. Birth of the problem or need.
(hidden, not yet discovered)
2. Business people face the problem or need. Attempts to deal with it. Attempts to solve it.
3. There is awareness about the existence of the problem or need. It is recognised. The decision is taken to submit it to the IT department.
4. Business demand is submitted.
5. Project is executed.



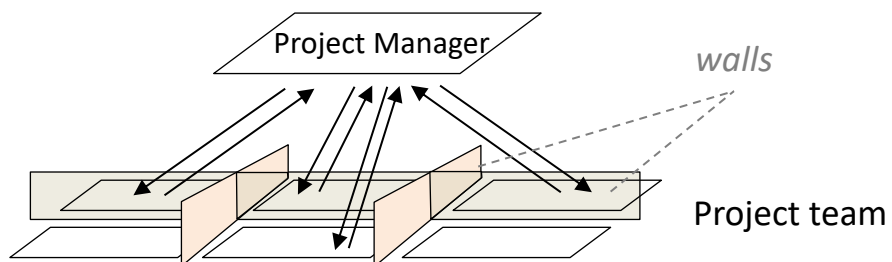
1. Systems Analysis (*approx.*)
2. Official application of informatics
3. Actual application of informatics
4. Possible application of informatics
(Systems Analyse)

1) Throw it over the wall



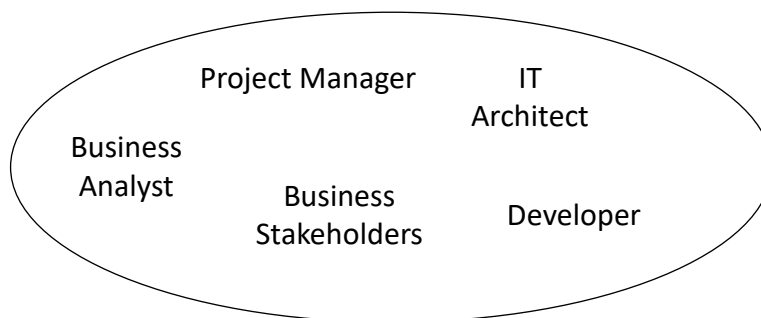
Some collaboration exists between two actors adjacent in the process.

2) Compartmentalisation of the project team and their work



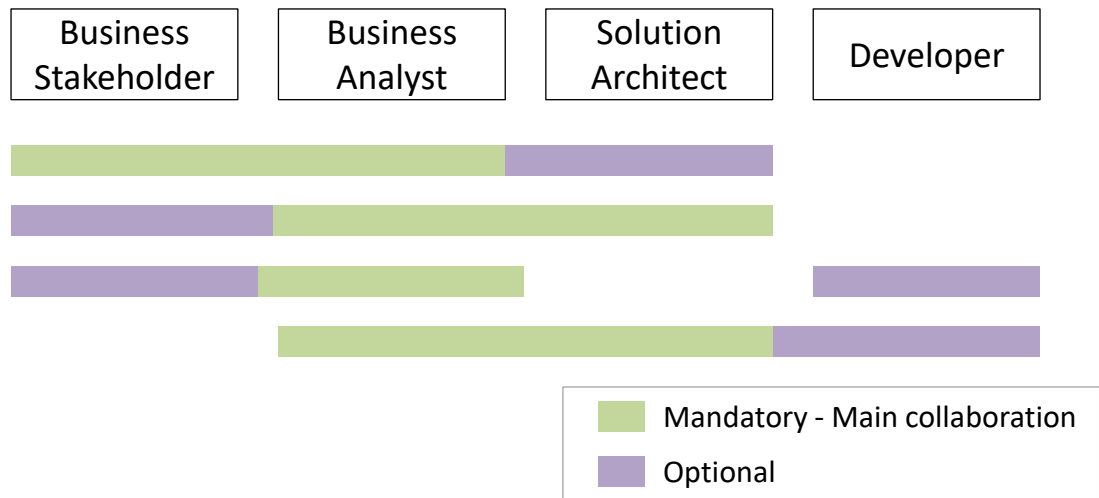
Walls exist between the different roles in the project. The work is subdivided by the project manager and assigned to the individuals or sub-teams. This is all managed from above. Team has no or a vague global overview of the project and there is little collaboration among them.

3) Full & free collaboration



This form of collaboration has drawbacks and poses risks. Which ones? And why?

4) Suggested collaboration



To perform a work or to obtain a result, some parties are mandatory. In some occasions (meetings discussing settled (global) issues, not unsettled details), the presence of other upstream or downstream parties is useful

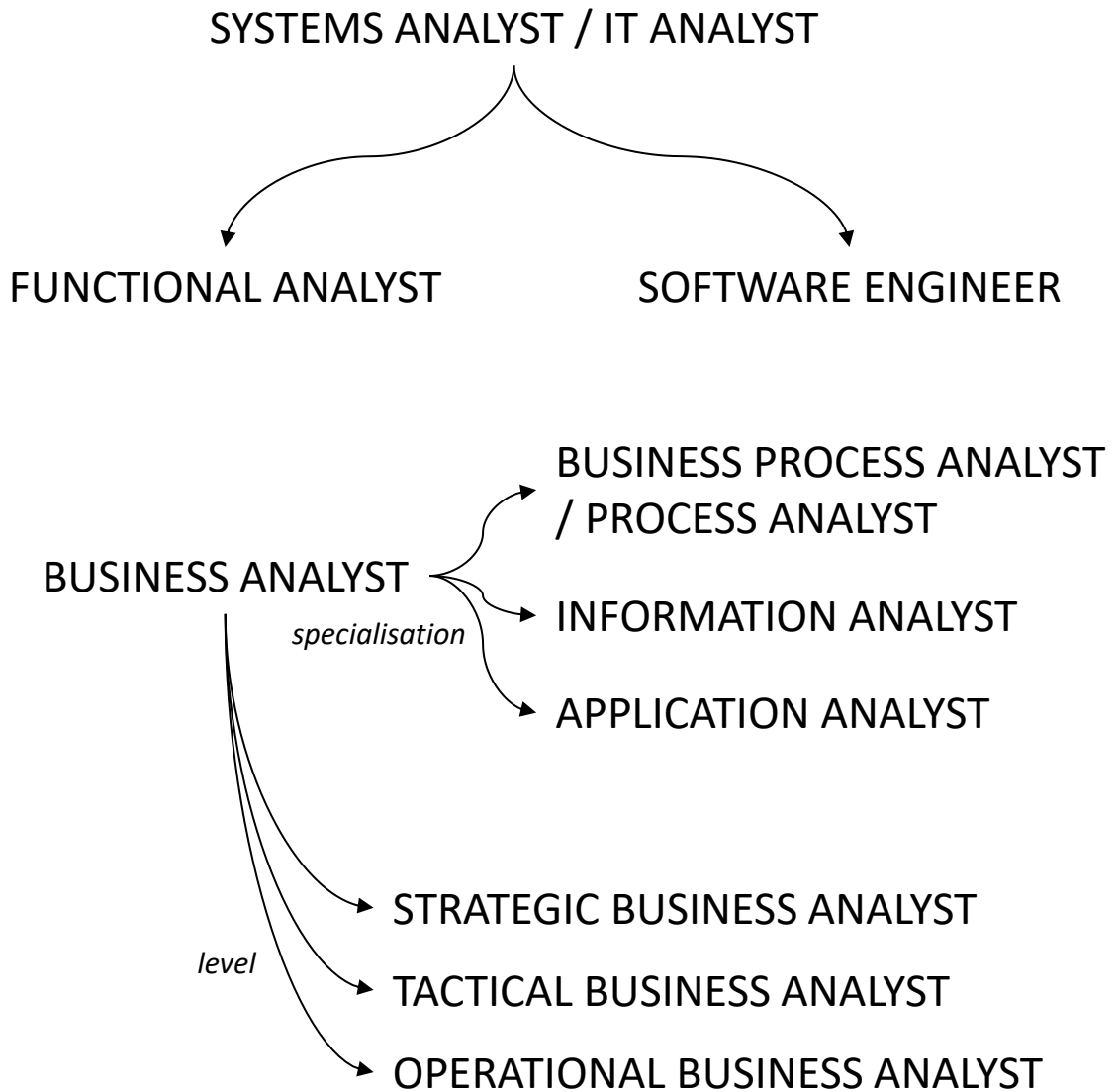
- to familiarise with the broader subject
- to learn from wrap ups and conclusions
- to solve specific issues
- or as a matter of training

Key question: Who can contribute significantly to a meeting/workshop or who can learn a lot of it?

- Avoid absences of people that can contribute or who could learn from it.
- Avoid the presence of people who can't contribute or who has little benefit from knowing what is/will be discussed. (Possibly, they may be informed by a conclusion or overview that will soon be given).



THE ANALYST



Notes:

The role of “Systems Analyst” emerged from a need. It evolved as needs were better understood and the awareness grew of better ways to use this discipline in different areas. Different areas lead to specialisations of the function.

Objectionable Drivers and Guiding Principles

30/10/2018

Goals, beliefs and principles to be avoided:

- to “satisfy the business demand”
- to limit the work to the mission and scope
- to have the requirements implemented
- to deliver a product that “meets the requirements”
- to “build and deliver the product, solution, software features as defined in the project”
- to “implement a set of needed software features”
- to build a software solution
- to create customer/stakeholder satisfaction
- to meet or exceed the customer/stakeholder expectations



These goals, beliefs and principles **LIMIT** the Analyst. They **LIMIT** the thinking and **CONTRIBUTION**.

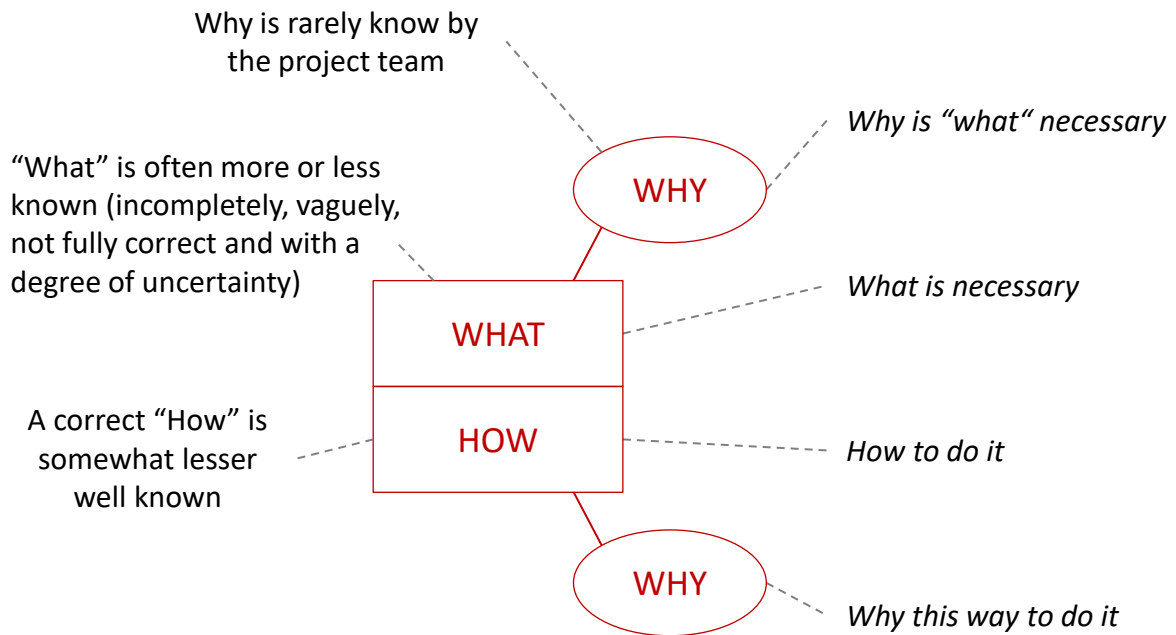
Business objectives, enterprise objectives (survival of the system) and broader outcomes are the drivers and objectives of initiatives.

Notes:

The analyst is often reduced to order taking. Then: The potential of systems analysis not exploited. And the analyst has little responsibility. The role of Order Taking Analyst makes real and good analysts to run away.

Professional vs Layperson

18/12/2018



The WHY something is necessary and WHY a certain approach, techniques, activities, steps and techniques are rarely known.

A professional differ from a layperson in that the professional knows the right answer to the WHAT, HOW and WHY's.

Avoid blindly rely on
WHAT PEOPLE ASK, WANT OR EXPECT

INSTEAD

**DO WHAT IS
RIGHT, WHAT IS NECESSARY, WHAT
MATTERS AND HAS VALUE,
WHAT IS REQUIRED
TO REACH GOALS
and what contributes to
the company's LONGEVITY and
THRIVING**

FIND OUT WHAT IS NECESSARY !!!

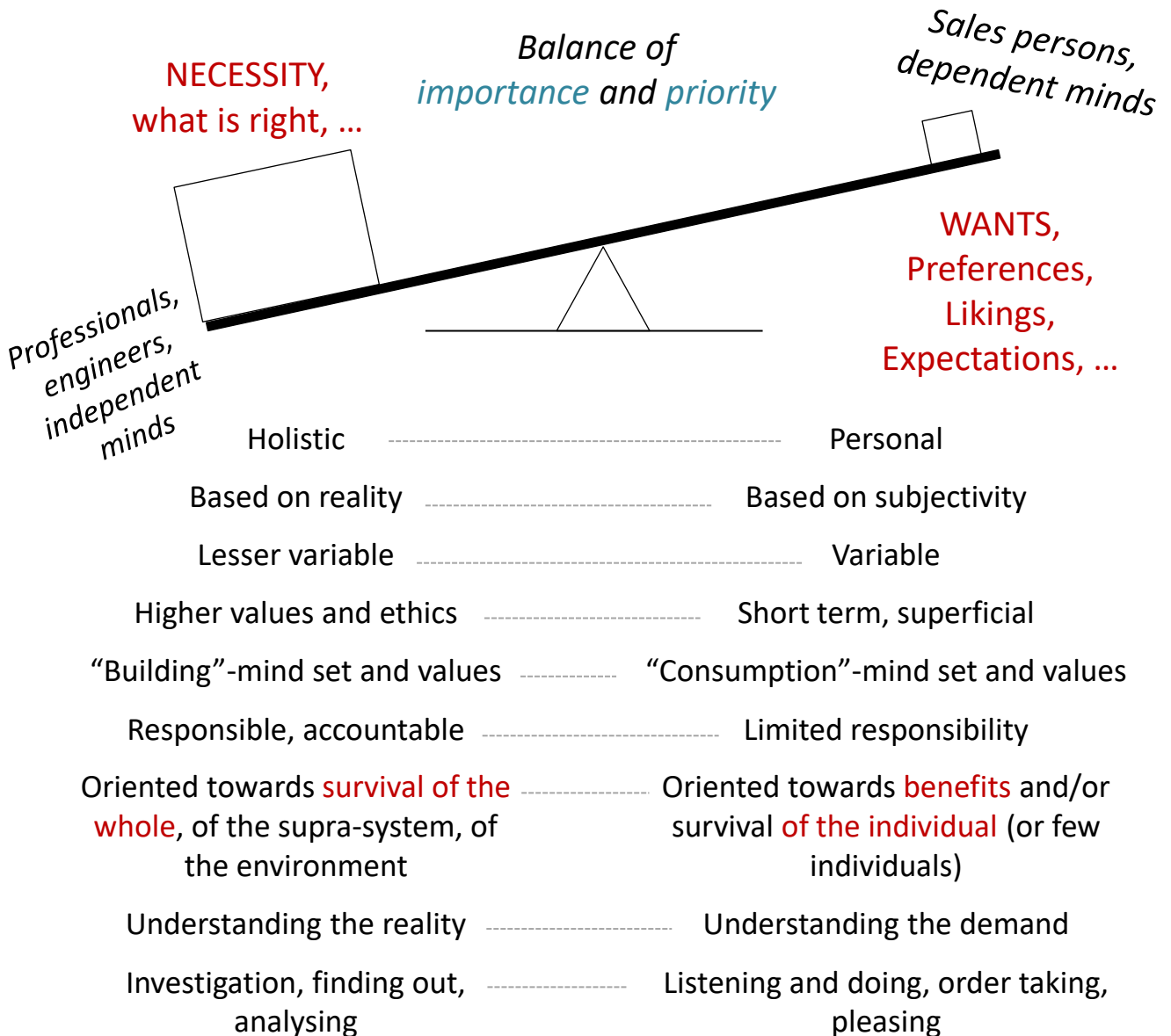
What the client says he wants is not always what
is necessary.

“The client doesn't know what (s)he wants.”

We may not base approaches on “what people want”

Necessity or Wants

15/02/2019



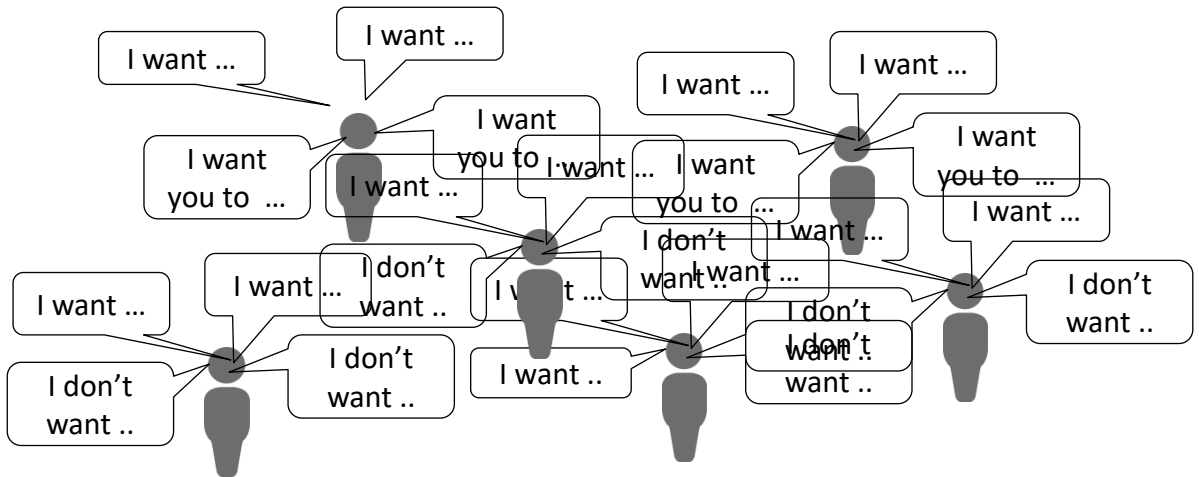
THE COMPONENT, THE SYSTEM, THE INDIVIDUAL ARE PART OF A GREATER WHOLE. THEY ARE PART OF AN ENVIRONMENT. THEY DEPEND OF THEIR ENVIRONMENT. EVEN THEIR OWN SURVIVAL DEPENDS OF IT.

Examples

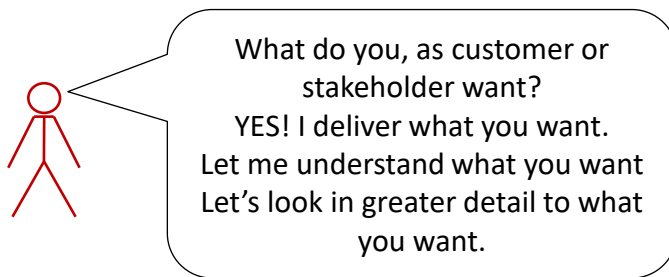
- In consumer: Delivering what the consumer wants may endanger the consumer (the consumer is not always aware of dangers of some products, like unsafe cars, fast food, chemical products, ...)
- In ecology: Doing what we wants, destroys “the planet” (nature, our life environment)
- In politics: Even doing what the majority wants, may destroy the system.

The Wants of the Stakeholders

15/03/2019



Motto: "What the Customer Wants" "Order Taking" – Solving Wants



- Customer doesn't know what he wants
- What he wants doesn't solve the problem or doesn't allow him to reach the objective
- What he wants has changed
- Although he received what he wants, he is still not satisfied.
- Unaware of the missed opportunity of creating so much better solutions
- No innovation possible (or unlikely, only minor innovations, ...)
- Wrong problem or wrong solution and find out that much time and resources have been wasted.



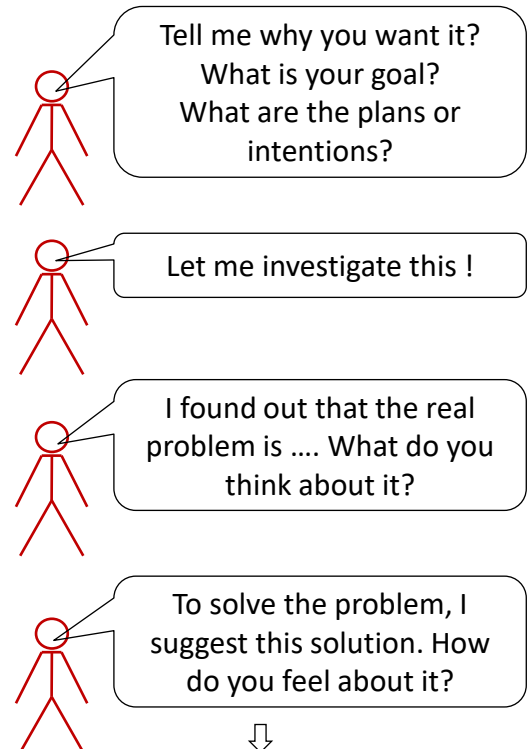
People ASSUME they know what they want



Customer Satisfaction ???

The key to failure is trying to please everybody

The Real Analyst Solving Problems, Achieving Goals



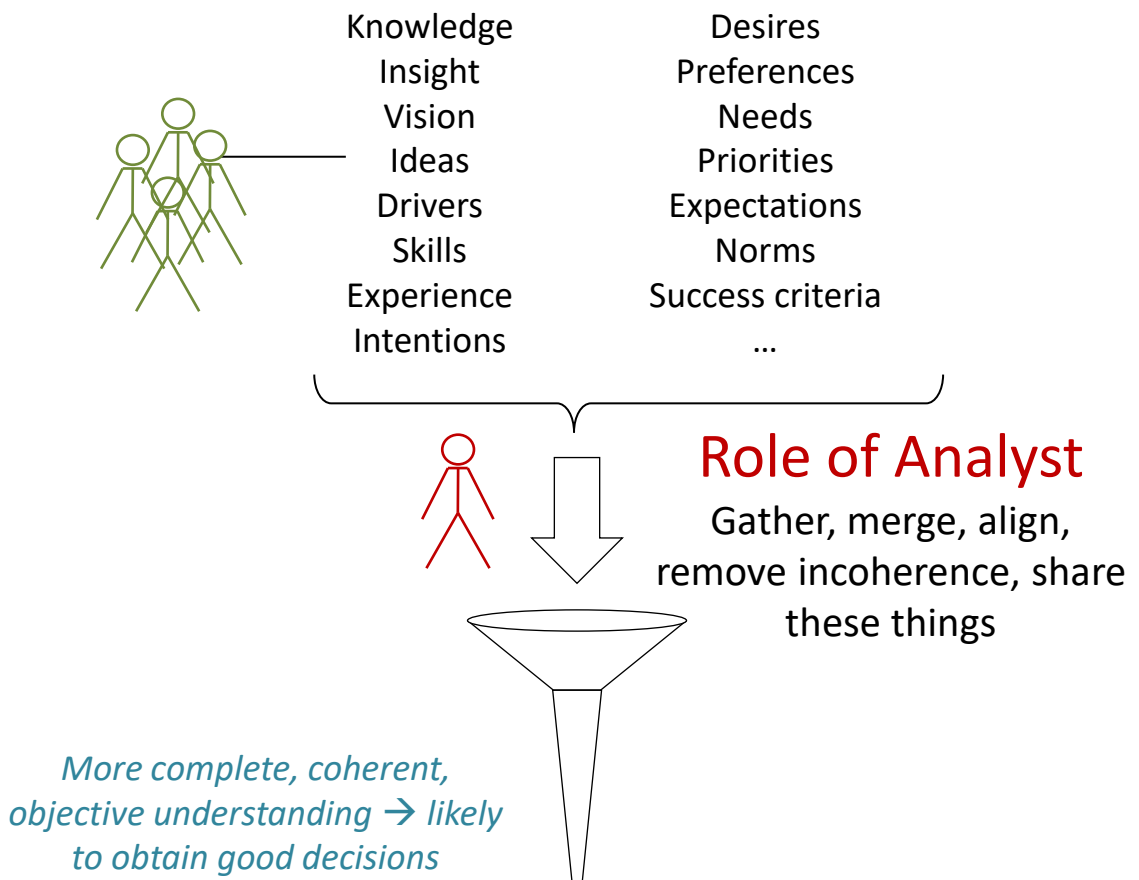
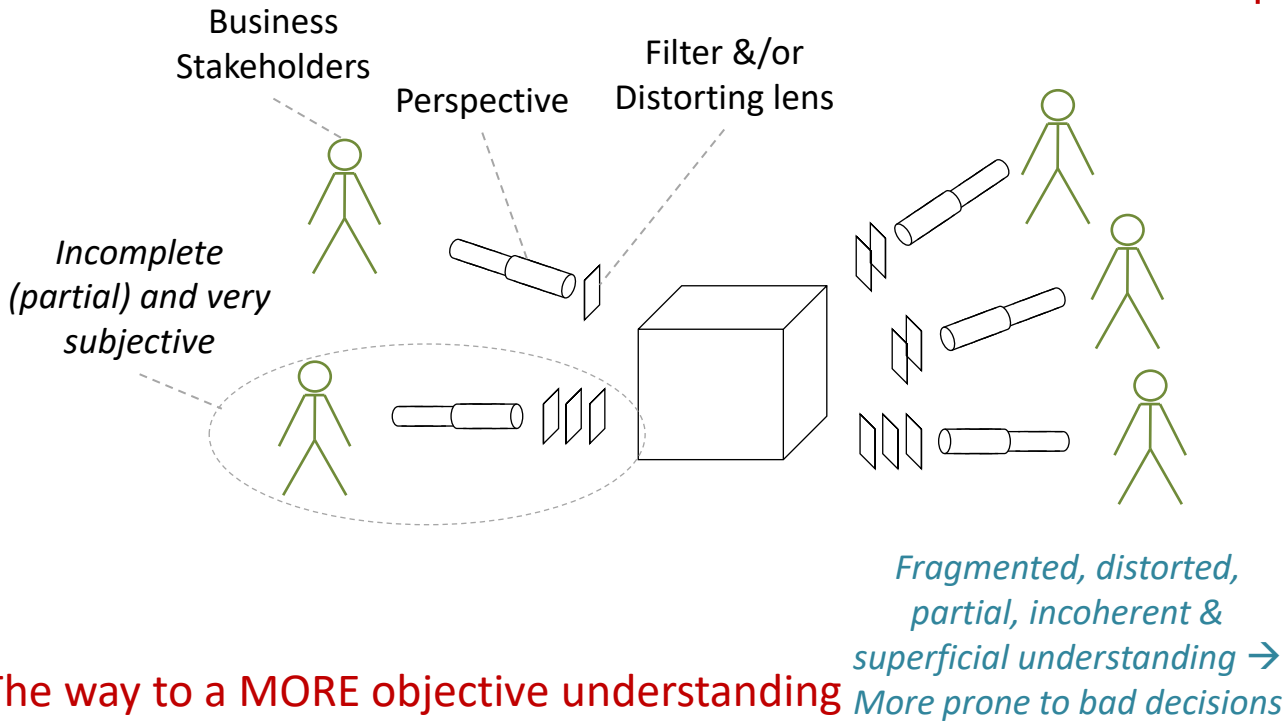
Building the right solution solving the right problem; well-designed; using best possible concepts of software; w/o creating new problems; possibly innovative;



Customer Satisfaction 682

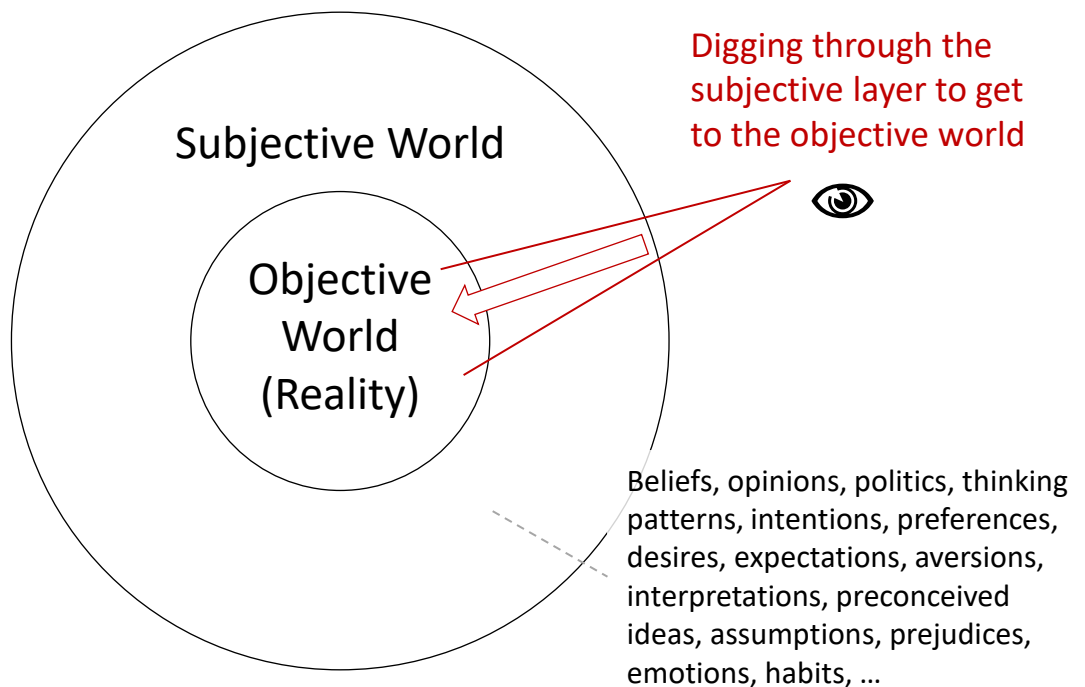
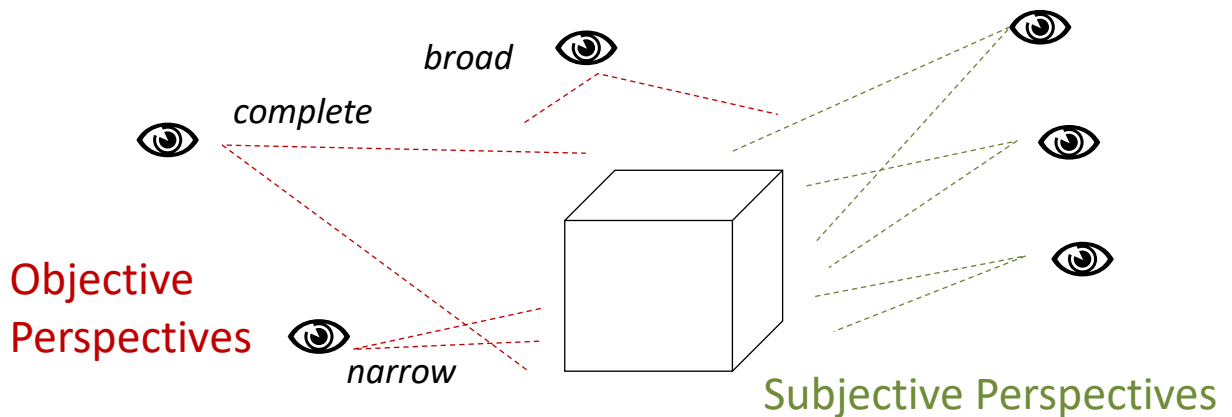
Perspectives and Filters

15/03/2019



Objective and Subjective World

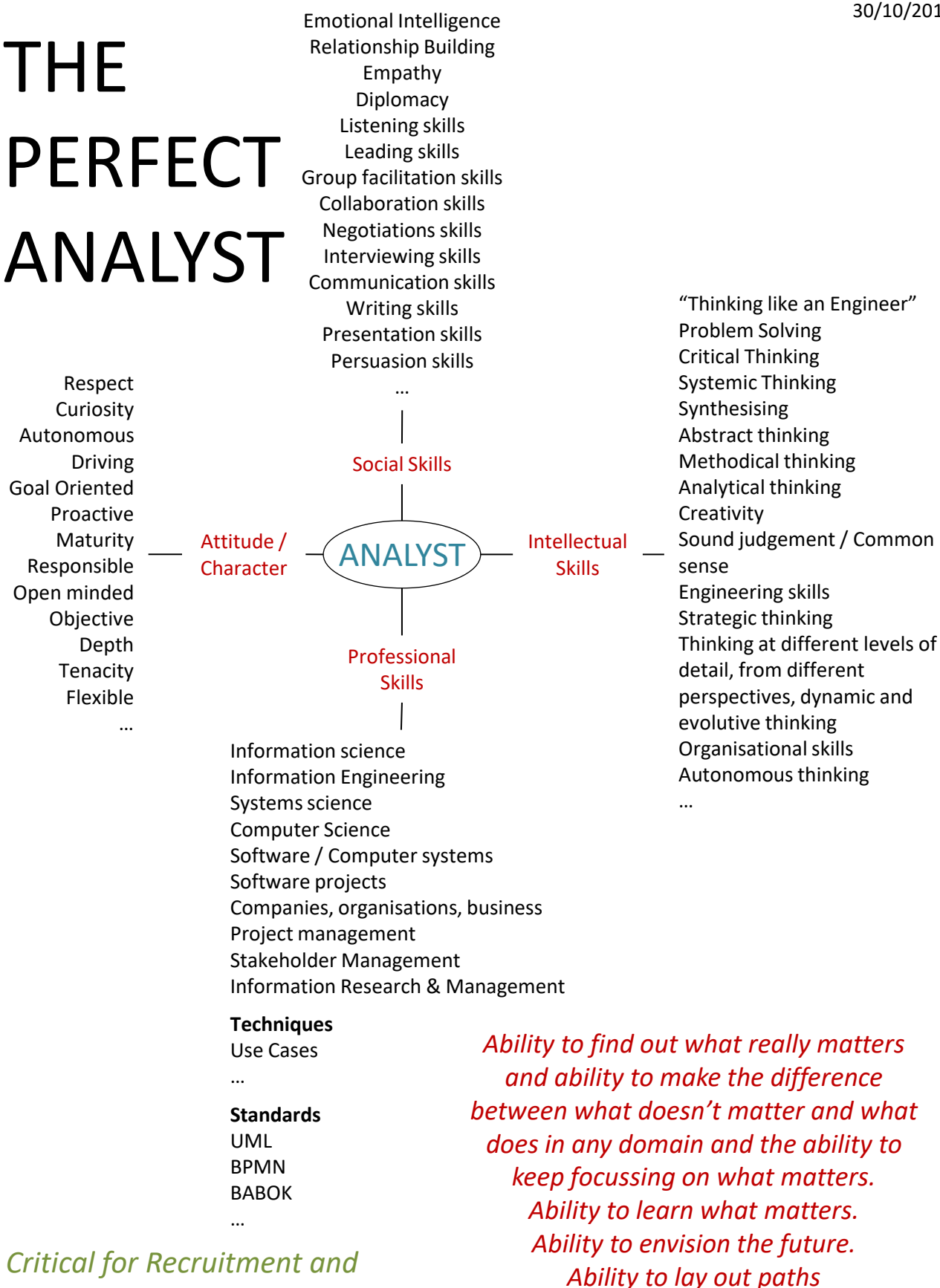
15/04/2019



Often the analyst needs to get through the subjective world to perceive the objective world, the reality, the real situation, as it is. The analyst has to be able to make the difference between the objective world and subjective aspects.

(S)He has to take the subjective aspects into account and deal with it (manage it).

THE PERFECT ANALYST



“Problem Solving”

Including all sub-skills like

- Critical thinking
- Abstract thinking,
- Systems thinking
- Holistic & Systemic thinking
- Sound judgment
- ...

Must be super-developed

>1000 x stronger than average problem solving skills

Critical !!

One can't be a (competent) analyst, engineer, architect, modeller, designer without this skill. (S)He may do more harm than good.

KEY TECHNIQUE: **QUESTIONING**

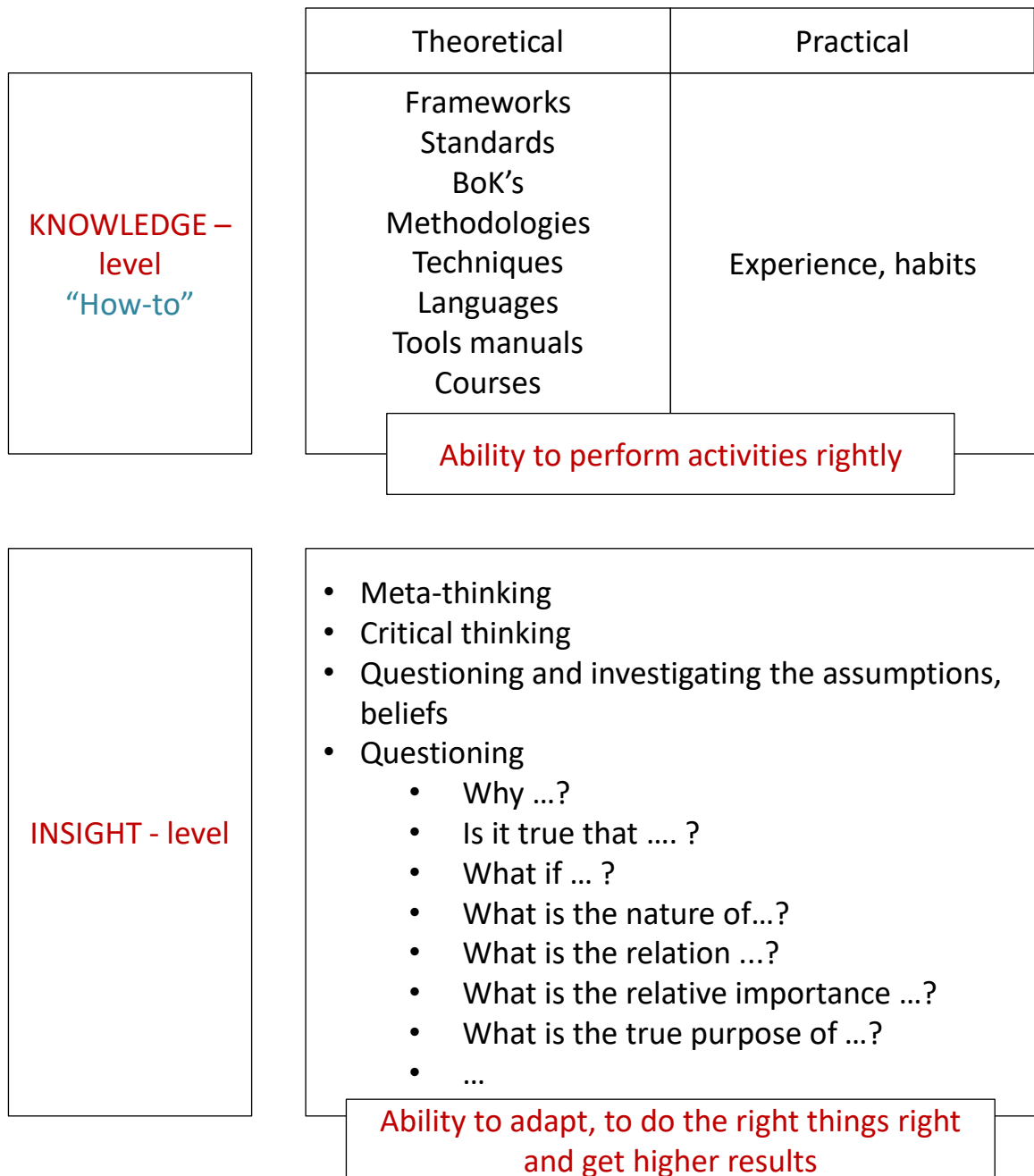
Analysts to Avoid

30/10/2018

- Seeking to please people
- Expecting to be told what to do
- Liking to be told what to do
- Seeking to comply
- Doing what is expected
- Accepting everything at face value
- Thinking in terms of black or white, all or nothing, 0 or 100%, yes or no
- Having very quickly make up their mind
- Making a lot of assumptions and being unaware of them (it's fine to make assumptions deliberately)
- Believing the majority is always right
- Confusing knowledge with understanding
- Seeing what is happening, not why things happen
- Thinking locally, short-term
- Interpreting and confusing terms and statements
- Following the rules, procedures, methodology (That's safe. Avoiding being blamed)
- Copying colleagues and habits (spread of ignorance and bad habits)
- 9-to-5 mentality
- Not caring about the product and result. Low norms: "It's fine if it works".
- Not seeking to understand the discipline
- 'Doing' is more important than 'Thinking'
- Can't work autonomously, need continuous help
- Fearing problems and complexity
- Fearing too big change
- Lacking of focus and direction
- Unable to take decisions (some hesitation is certainly fine!)
- Systematically consulting others for their opinions about problems. Fully relying on them for solutions.
- Spending a greater importance on opinions rather than on facts
- Going along with the simple but wrong answers
- Being glad with the first answer or solution that comes to the mind
- Systematically buying or copying solutions as approach to solve problems
- Calling everything that is related a 'solution' even if it doesn't solve anything or creates more other problems
- Solving symptoms
- Preferring simple or quick 'solution' (design debt)
- Doing more of the same, give up or throw away the 'old' and replace it (don't fix it, don't change the thinking, let alone the beliefs)
- Not having all the necessary social AND thinking skills

Competency Development

30/10/2018

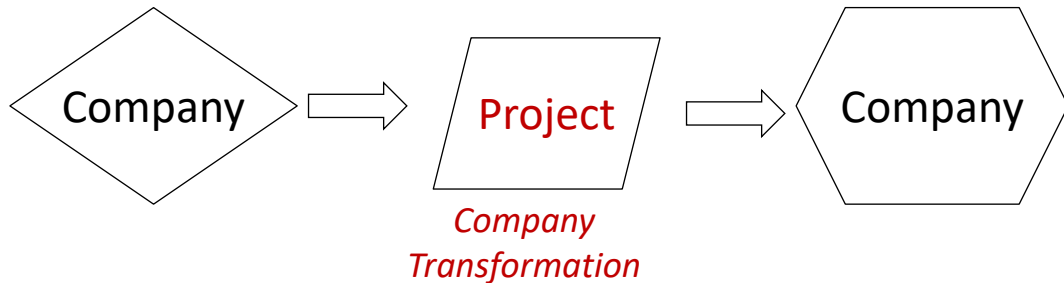


The development of the “insight”-level is the most difficult but also the most critical

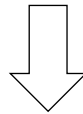
Most of the standards, frameworks, ... majorly are how-to-guides.
 “Follow the guide / the majority / the present tendency !” – mentality.
 Knowledge level only doesn't suffice. It leads to “cargo-cult”.



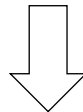
PROJECTS



Projects are a way **to build and to transform the company.**
Projects are a way **to execute strategies.**



Projects are **COMPLEX, UNCERTAIN & RISKY**
Projects are **CRITICAL** to the COMPANY



Projects skills and/or engineering skills are critical

A project is not an activity that can be done by just any group of employees. Neither is it a social gathering of nice people. It is a group of collaborating experts ... at least if experts have been recruited and/or assigned to the project.

Skills, insights, Methods and Actions help to **deal with complexity** and to **decrease uncertainty and risks.**

A Project ?

18/12/2018

PROJECT MANAGEMENT'S PERSPECTIVE

General definition

- Endeavour, A process
- Producing a unique product
- Progressive elaboration
- Executed by people
- Temporary (has begin and end)
- Uncertainty and Risks are inherent

A project is more than a process. A project is :

- **a temporary organisation**
- functioning in an (work) **environment**
- a **process** to produce a **unique agreed outcome**
- using **resources**
- **as efficient and effective** as possible and **limiting risks**
- **maximising the benefits**
- an initiative that must be **set up, planned, guided and managed.**

COMPANY'S PERSPECTIVE

(IT) Projects are

- A mean to execute (parts of) a **strategy**
- A way to **develop** new **products and services**
- A way to **create assets**
- A way to **innovate**
- A way to **build (a part of) the company / organisation**
 - to strengthen, improve and transform / adapt / change the company/organisation.
- An **INVESTMENT!!**

Conclusion:

The capability of managing and executing projects is **VITAL**, a matter of life and death; a matter of disappearing, surviving or flourishing for the company.

Stakeholders have

- Intentions, Objectives and Plans
 - Needs
 - Issues, problems, obstacles, complaints, opportunities
 - Necessities
 - Information, knowledge and insight (information sources)
 - Expectations
 - Limitations
 - Resources
-
- They **affect or are affected** by the project, product or by the product's outcome.
 - They provide **input to the project**
 - They may bring **opportunities** and **solutions**.
 - Trust and the right(!) **relation** has to be created.
 - **Collaboration** must be put in place.
 - Decision power: They will have to **take decisions**.
 - They will evaluate the project, project's product or the realised outcome.
 - They constitute **risks**. (They may make a project to fail!)

STAKEHOLDER	POSSIBLE OBJECTIVES
Company	To survive, to grow, to evolve, to live accordingly to its purpose and mission, to serve the society, ...
Management	Reaching its objectives, increasing benefits, lowering the costs, executing a plan, ...
Business community	Conducting business, reaching objectives, ...
Informatics Department	Providing information solutions to the business, creating value for the business, responsibility and capability to manage and maintain the implemented Informatics solutions and to make them evolve, implementing a strategy, innovating, ...
End-Users	Reaching his/her objectives, having a good evaluation, avoid annoying work, offering a better service to the clients, release pressure on his/her own work, ...
Sponsor	Financial benefit, business objectives, ...

Criteria

Criteria		Criteria	
On Scope	<input type="checkbox"/>	According to requirements	<input type="checkbox"/>
On Time	<input type="checkbox"/>	According to specifications	<input type="checkbox"/>
On Budget	<input type="checkbox"/>	Passed the Acceptance Test	<input type="checkbox"/>
According to plan	<input type="checkbox"/>		

Taking into account : Taken risks, leaving a deteriorated situation behind (e.g. technical debt, documentation debt, chaos, burn-out, people leaving, ...), ...

Does a project fails when it misses one criteria ? Can the same criteria be used to define failure? (not success = failure?)

Duration of Satisfaction: Project closure, acceptance of product, longer period?

But....

QUESTIONS

Is a project executed to deliver on time; within budget; respecting the scope, demand and specifications and meeting expectations?

or

Is it executed because the product, solution or change will be beneficial to the company and to its clients? (investment to build < value produced by the product, solution or change)

Is delivering within the iron triangle more important than having a product creating a maximum of value and limiting risks?

TIP: Criteria to be agreed upon and to be recorded in the Project Charter.

CRITERIA

A PRODUCT / SOLUTION is SUCCESFULL if ...

- Working software
- Meeting the business demand
- Meeting the business stakeholders expectations
Which ones? Have the expectations (not requirements, but real expectations) been recorded and validated before the project started?
- Improving the activities of the business/organisation
Seamlessness, flexibility, efficiency, effectiveness, safety, respect of rules and laws, ... and a decent quality.
- Allowing the department to reach its objectives
- Allowing the company to reach its objectives
- Improving the value and exploitability of information
- Creating value for the company
- Creating value for the company's client
- Fully exploiting the opportunity
- Strengthening the company

Meeting the different criteria require a different level of thinking, mind-set, norms and values, skills, collaboration, approach, methodology, set of objectives, ...

Notes:

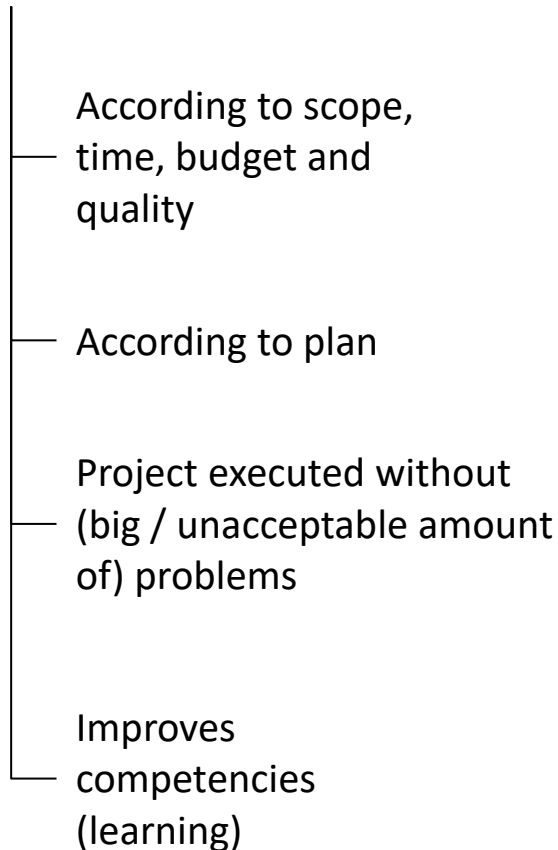
Can the same criteria be used to define product failure?

Does it solve a problem or need? Does it contribute to the sustainability of the company?

Project & Product Success

30/10/2018

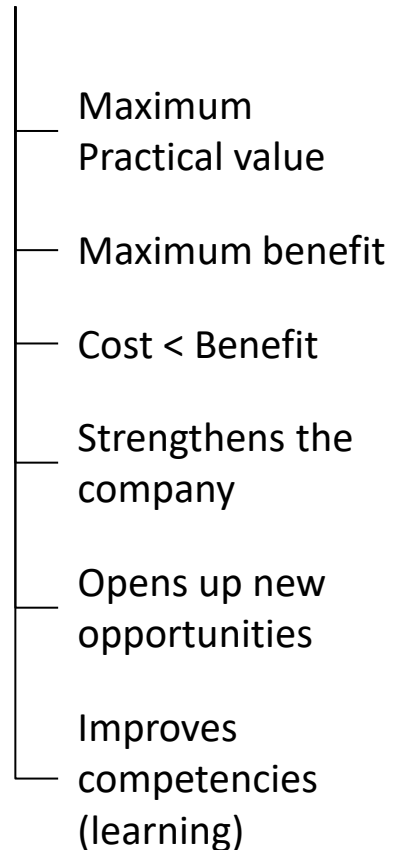
PROJECT success



Measured in duration: several months, few years

Measured at the end of the project, sometimes based on estimates made early in the project

PRODUCT success



Measured in duration: over several years

Measured over a long period of time (once operational)

Notes:

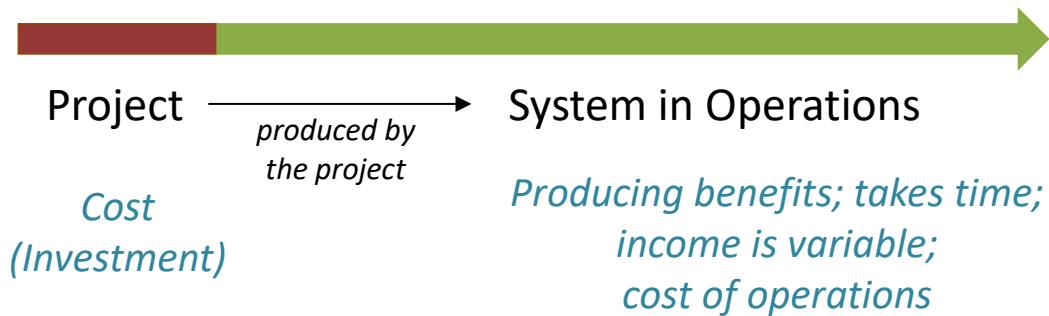
Defining the right success criteria is important. Measuring correctly.

Learn the right lessons. Put it in perspective: project success = more important than product success? On what type of success does the company/stakeholder focus most?

Investment, Cost, Benefit of Product

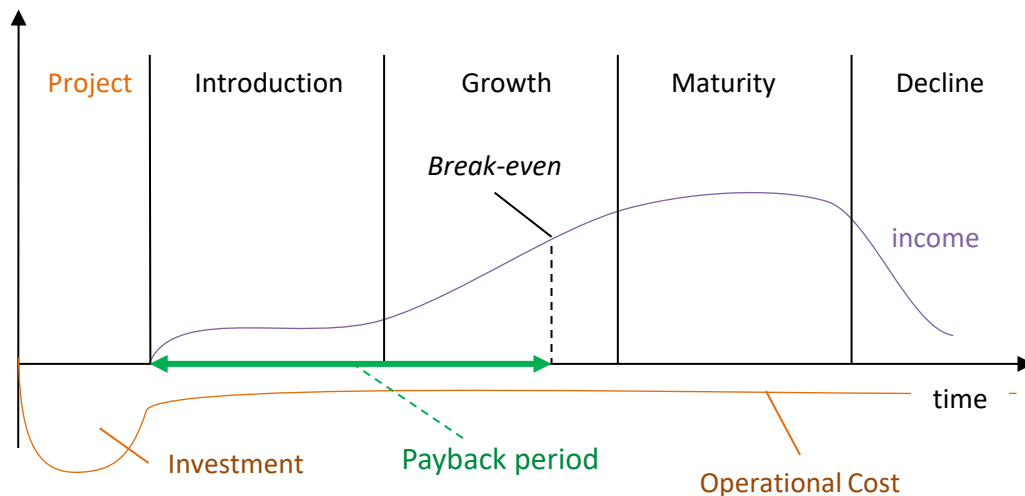
30/10/2018

Product Success



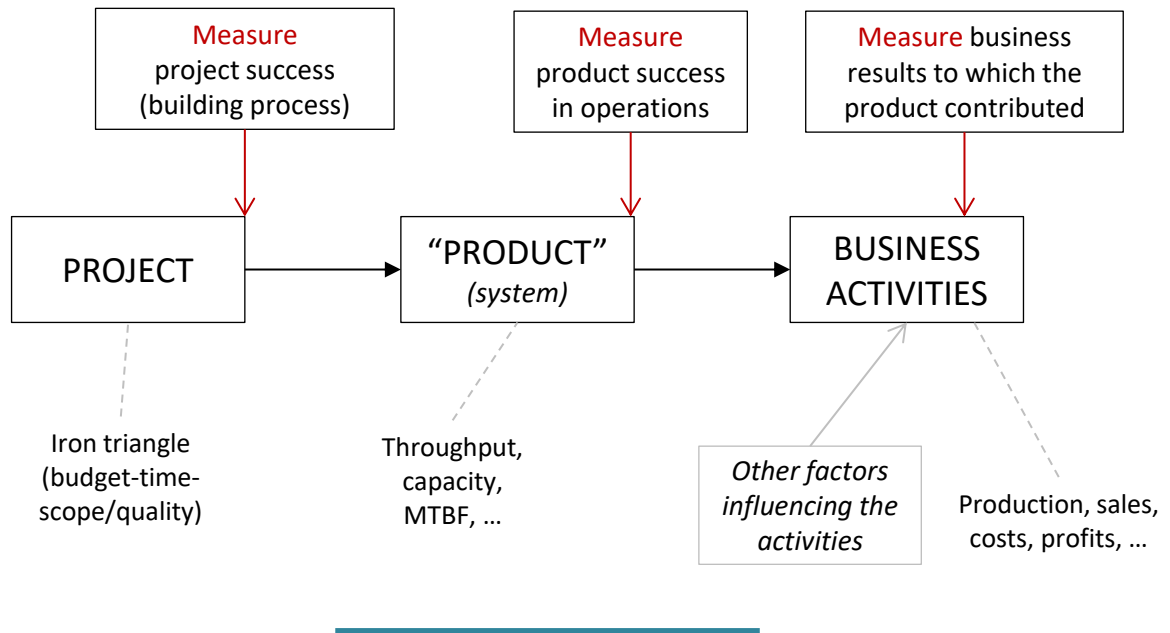
Product is Profitable? **Total Cost < Produced Benefits**

Simplified System's Lifecycle (financial perspective)



Remarks:

- 1) During the life changes may be implemented. Changes have a cost, an increase of benefits (or lowering of cost) and a payback period
- 2) Each innovation may bring in an increase of income in diagram of the system, or, for example postpone its decline.



"Product": A project produces a product, which is a part of the enterprise, such as a capability, a solution, a system, an adaptation of an existing system, a set of processes, ...

This "product" allows to execute business activities.

The success of each (Project, Product, Activities) can be influenced by other, sometimes external, factors, such as

- **Project:** decisions to be taken by business, quality of received, ...
- **Product (System) :** Training of discipline of the users, pressure
- **Business Activities:** Products, Services and actions of competitors, ...

Notes

When evaluating the contribution of the solution/system, other factors that influenced the business results have to be taken into account

Project Intentions → Project Driver

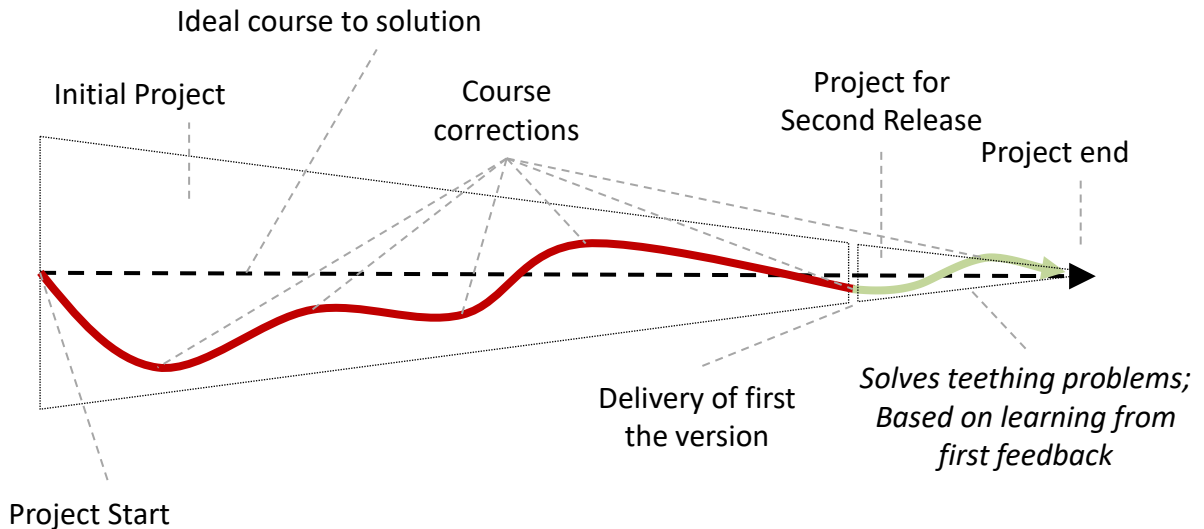
10/01/2020

What are common intentions for informatics projects ?

1. [] A Vision to implement
2. [] A Strategy or Plan to execute
3. [] Legal constraints to respect
4. [] An Architecture to implement
5. [] An Objective to meet
6. [] New products and services to launch
7. [] New capabilities to develop
8. [] An amount of additions and improvements to implement
9. [] To keep workers busy
10. [] Abundance of Time
11. [] Excess of money. It has to be spent.

Answer is 1 to 8 :	Objective ("Scope"-related)	} unlikely
Answer 9 :	Budget, Cost (& Resource)	
Answer 10, 11 :	Time (& Resources)	

- Commonly, a projects is started and systems are built in order to obtain a result, to achieve a goal.
- The **Scope is the main natural driver** of projects, and time and budget are dependent.
- Without profitable **scope or goal**, there would a time and budget.
- Budget and resource allocations are assigned based on a justification of a beneficial product, outcome or goal.
- A deadline can be established and scope can be adapted to meet the deadline. But without scope there wouldn't be a deadline.



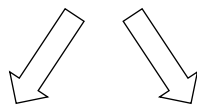
- Course corrections may imply scope changes, adaptations of project plans, budgets, schedule, delivery date, ...
- Analysis reduces the likelihood and the amplitude of deviations from the ideal course.
- Course corrections shouldn't be caused by "changes of mind" or preferences.
- After Analysis the likelihood on changes should be severely reduced. It should be even more reduced after the design.

Projects are Semi-structured / CASE-based Processes Initiatives

Implications:

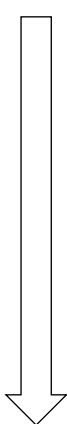
It is senseless to deal with a project as if it was a strongly structured process that can be defined fully upfront and then executed as written.

A process or plan devised upfront can not be considered as prescriptive.
(see page about un-/semi-/structured processes)



Any **Project Plan**
defined upfront
will **have to be adapted**
during the execution

Mastery of the **Discipline**,
in-depth understanding
and common sense
are **critical**

1. Produce the product described in the customer's demand
 2. Meet the requirements
 3. Meet the expectations
 4. Produce a specific product
- 
5. Improve company's KPI's or other measures
 6. Contribute to a pre-determined business objective and allow to reach it
 7. Drive business through optimisation, improvement, innovation, new ways of working and new products and services

Allowing only a specific or a limited way to contribute



Allowing to contribute in a broader range of ways;
Offering more freedom;
better use of possibilities and skills;
More can be learned

ASSIGN/LINK HIGHER OBJECTIVES TO PROJECTS

RESULT DELIVERY rather than **PRODUCT DELIVERY**

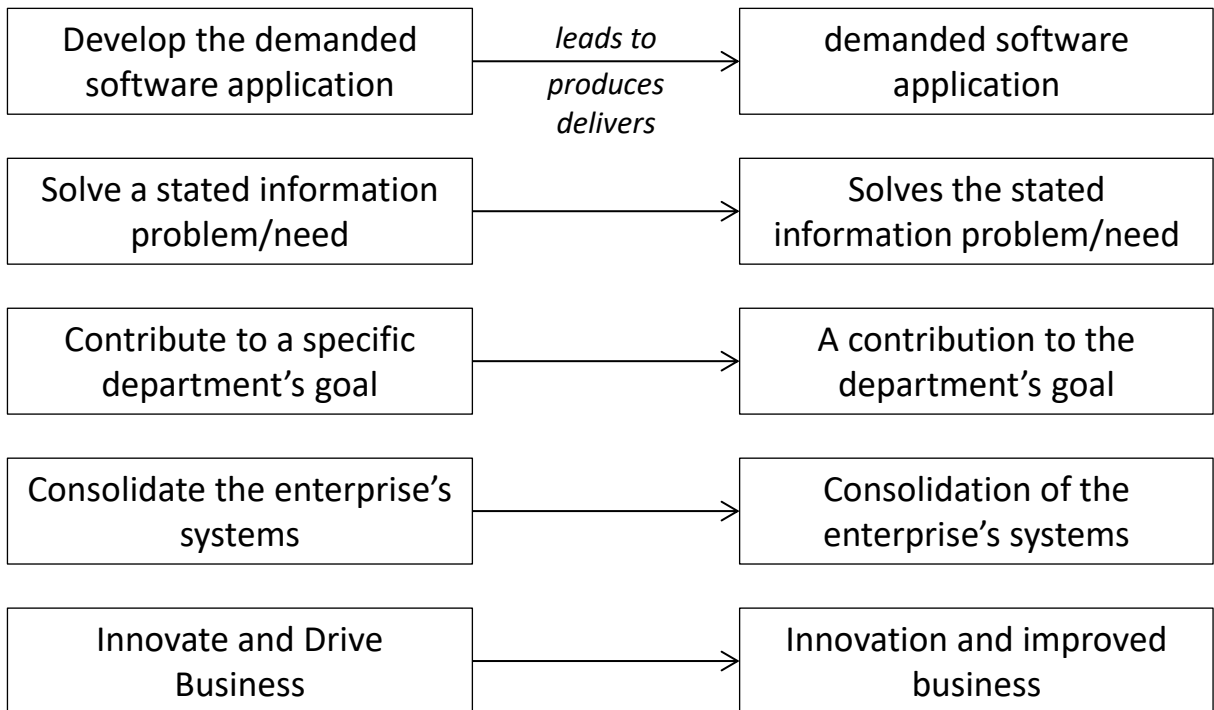
(organisation, human resources and collaboration have to be adapted)

Project Objectives

18/12/2018

OBJECTIVE
“mission”

RESULT
“what you will produce/get”



Low objectives leads to low results

High objectives leads to higher results

The objective assigned to the project
determines
the result produced by the project.

The stated **objective** is habitually the maximum that the project will deliver.

Notes:

It is better to turn Software Development projects at least into Information Systems Development Projects.

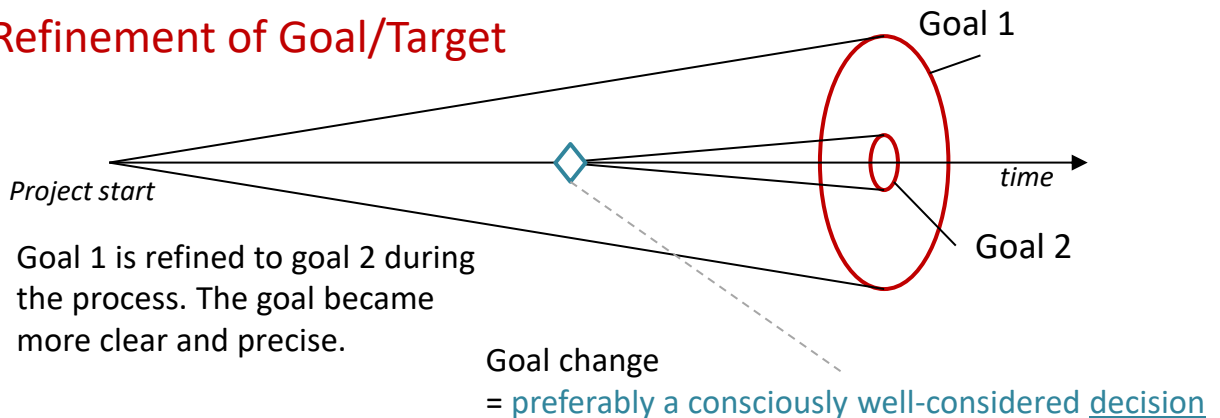
ASSIGN COMPANY's GOALS

To ISE/IT PROJECTS,
and also
to ARCHITECT's jobs,
to ANALYSTS's jobs,

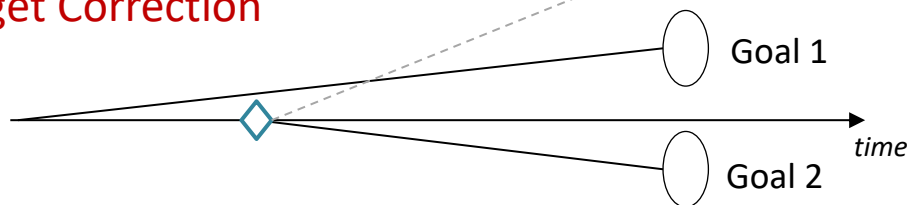
In order to get more creativity,
more innovation
and much better solutions and benefits

Give target and context and let people do their job.

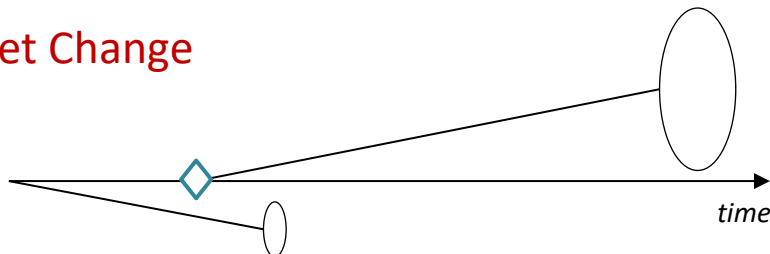
Refinement of Goal/Target



Goal/Target Correction



Goal/Target Change



Goals are tight to Purpose

Know the (real) purpose(s) and you know the goal

Notes:

A goal provides either an objective, the idea of the pursued final desired outcome, or a direction. It brings focus. It allows to align all the means and efforts in one direction. It brings focus, stability, effectiveness and allows efficiency.

A goal avoids wasting resources and efforts in different and/or opposite directions or on opportunities of lower interest. Frequently changing the goal(and the direction) is the same as having no goal.

However, fixing the goal, cutting it in stone, is also a misuse of the concept. It makes no sense to continue to pursue a goal that has been found being useless. Small goal changes or goal refinements are normal. Frequent important goal changes are a symptom on an underlying problem. 180° change of objective can sometimes be necessary, but should happen very rarely.

Goal and strategy (or other plan) shouldn't be confused.

A Project – A Few Reflections

10/01/2020

Objective

- Reaching a Business Objective
- Implementing a part of the Strategy or tactic
- or Strengthening or improving the Enterprise

Purpose

- Adapting the company or the company's product and services

Investment

- Intent to have a favourable return on investment
- A company doesn't invest in initiatives, blindly, without knowing what they hope, expect or will get. It is not a carte blanche.

Difficult, Complex and Critical

- Since it strengthens and transforms the company, the most skilled people should be assigned on projects.

Risky

- Implies the project may be lesser successful or even failing. We have to accept this eventuality.
- Need to minimise (mitigate) risks

Life after the Project goes on

- When the project is done, we need to leave the company in a better state (people's morale, servers, documentation, technical debt, ...)
- Product of the Project has a Lifecycle

This has to be respected when defining the objectives, norms, approaches, products, when taking decisions, ...

Concepts Defining the Project^{10/01/2020}

Project Mission

General statement describing what the project aims to achieve.

Project Goals / Objective(s)

Objectives the project seeks to achieve. Note that the project's product has also a goal or objective.

Project Drivers

Reasons, motives, why the project is undertaken.

Project Boundaries

Boundary delineating the domain/area that can be investigated and/or which can potentially be adapted or directly impacted by the project.

Project Dependencies

Projects are dependent of external factors of which the project depends. They rely on them. Therefore, they affect the project execution and their outcome. They may pose a risk to the project.

Project Assumptions

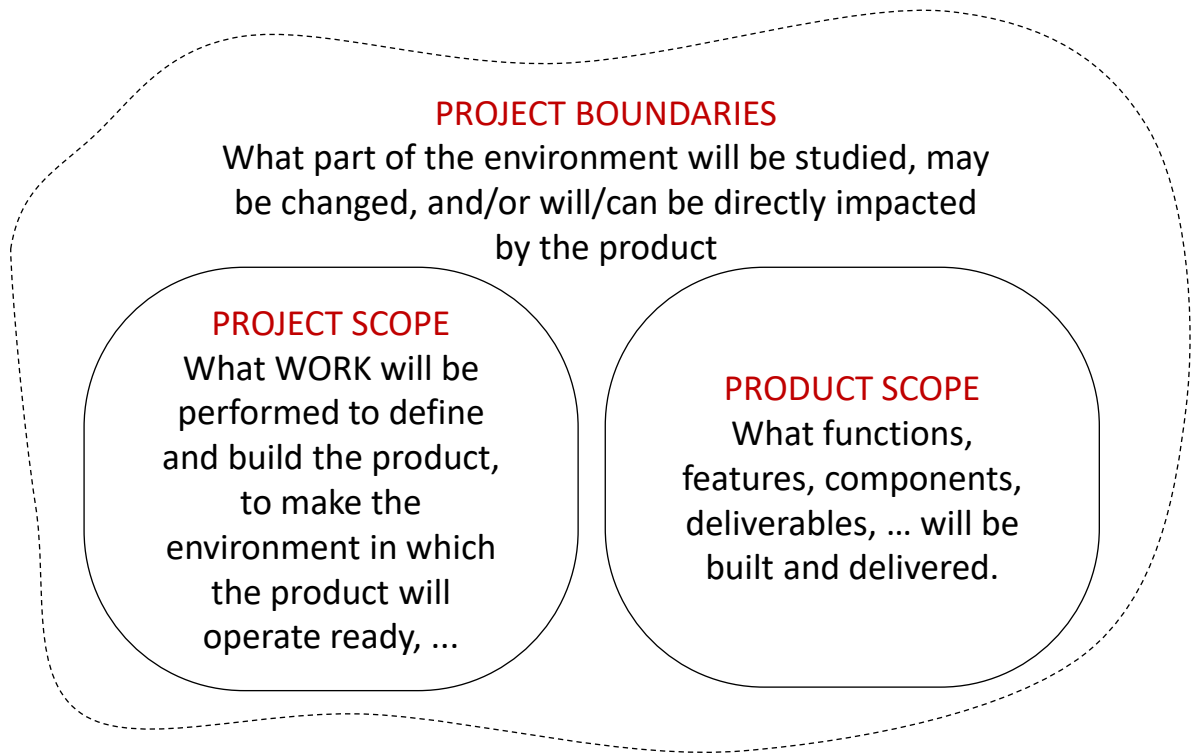
Hypothesis upon which the project is based and which may significantly affect the project execution, its outcome or the value of the outcome. They have to be clarified if possible and when possible. Vigilance is required.

Project Constraints

Project constraints are restrictions or obligations imposed on the project. (Similar to [Product Constraints](#)).

Product Scope, Project Scope, Project Boundaries

14/02/2020



These elements define limits of the project. **They DELIMIT !**

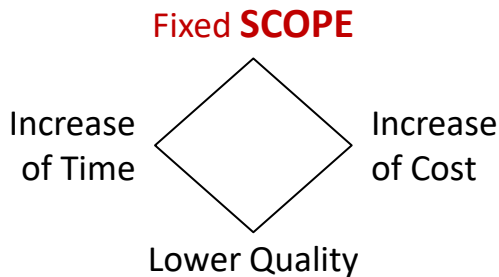
However ...

1. The business environment may change during the course of the project.
2. New insights, issues and ideas may surface at any time.

Therefore

When it makes sense, when it is beneficial, they should be able to be adapted.

**Plans are
adaptable and dynamic artefacts**



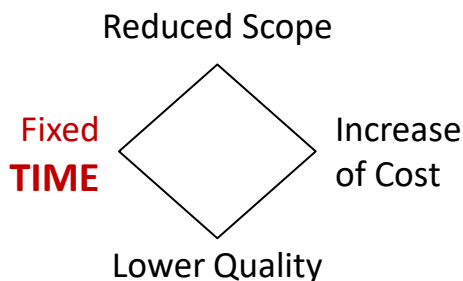
Companies work based on a vision, on strategy, on goals and targets, on production of value, on limiting risks and costs, on increase of benefits and on regulations.

In a natural way, this leads to the submission of project proposals. The product or outcome is the primary reason why projects are executed.

It is obvious that scopes define the missions of projects. Time and cost are deduced from the product scope.

What happens if a project is running late and the scope is fixed?

Time will increase. Cost will increase. The quality can be lowered, sometimes in an hidden way. This creates debt. Or it can be a combination of these.



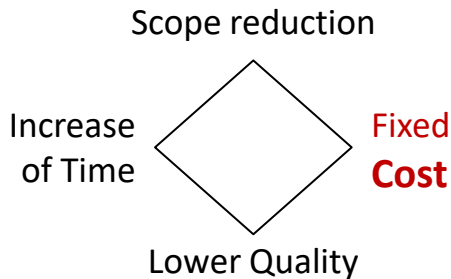
Time is never the trigger and driver for the start of a project. It can be an imposed criteria (legal obligation, celebration day, ...). A client is rarely ready to invest in a project of which (s)he doesn't know what (s)he will get, and thus what will be the return. Making some vague estimations is always possible. (see predictability of constraints)

What happens if a project is running late and the delivery date is fixed?

The scope can be reduced. The cost may increase. The quality can be lowered, sometimes in an hidden way. Consequences of a lower quality will come later. Or it can be a combination of these.

Note:

- When something is "fixed", this means that it is the main constraint. It is the last one that should be adapted. And in some case, it won't be possible to adapt it, although those cases are rare.
- In most cases, adaptations of the constraints is not desirable unless it is really beneficial. 709

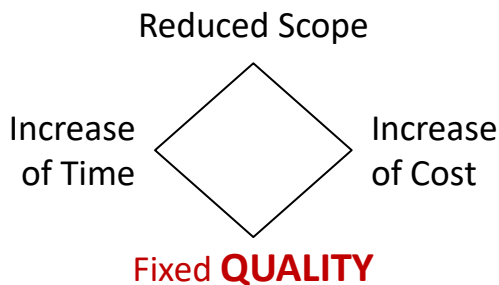


Cost is never the trigger and driver for the start of a project. Sometimes it is possible to find additional budget. This depends of the expected benefits of the project. A client is rarely ready to invest in a project of which (s)he doesn't know what (s)he will get. Making some vague estimations is always possible. (see predictability of constraints)

It is obvious that scopes define the missions of projects. Time and cost are deduced from the product scope.

What happens if a project is running late and the COST is fixed?

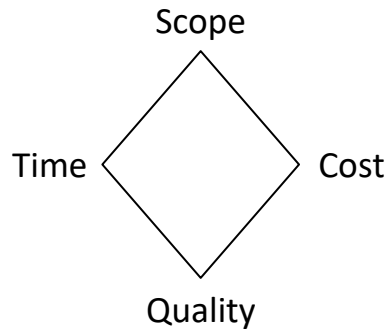
Scope can be reduced. Time may increase. The quality can be lowered, sometimes in an hidden way. This creates debt. Consequences will come later. Or it can be a combination of these.



Quality is never the trigger and driver for the start of a project. However, quality can be a non-negotiable constraint. A client is rarely ready to invest in a project of which (s)he doesn't know what (s)he will get, and thus what will be the return.

What happens if a project is running late and the quality is fixed?

The scope can be reduced. The cost or time may increase. Or it can be a combination of these.



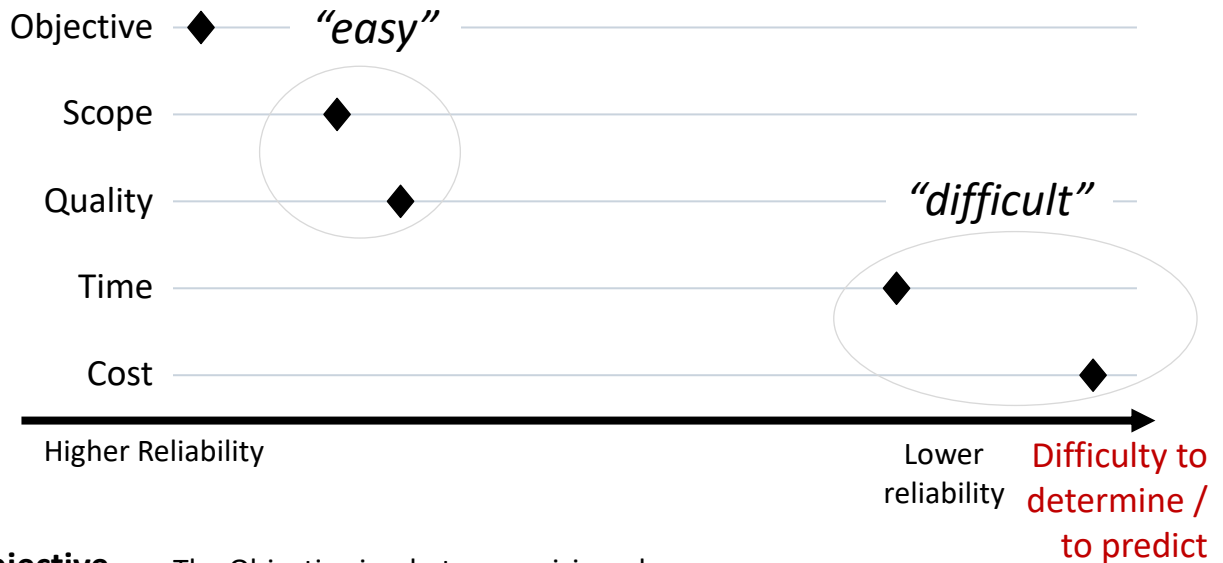
We can negotiate on 5 aspects.

So, there are SIX questions

1. Can we adapt the scope? For example, by postponing some parts.
 2. Can we simplify the solution while keeping the scope?
 3. Can we reduce the project duration?
 4. Can the cost be reduced?
 5. Can we be satisfied with reduced quality and where can the quality be lowered and until what point?
 6. Or can any of the answers to the question above be combined to have a satisfactory result?
- It is advisable to try to do a good analysis and to estimate better in the first place and to reduce everything that may pose delays.
 - A project runs out of scope, time or budget for a reason: inefficient analysis, bad estimations, bad leadership, bad planning, work environment, insufficient collaboration, received information, slow or bad decision making, ...
- Maybe first find and deal with the causes rather than trying to solve the consequences. Only then we can learn valuable lessons !!
- What if we could assign lesser importance to the iron triangle and more focussing on the product?

Difficulty of Predictability of Project's 5 Main Parameters

14/02/2020



Objective The Objective is what we envisioned.

Scope The Scope can be determined (if we focus on what is necessary, rather than on wants and if SASD is done decently)

Quality Required Quality is relatively easy to be determined.

Time Time = mainly $f(\text{Scope}, \text{Quality})$
+ non-linearity + uncertainties + quality of organisation, collaboration, information, ...

Cost Cost = mainly $f(\text{Scope}, \text{Quality}, \text{Time})$

The product, as defined by the objective, the scope and the quality create satisfaction. They create the value.

Time and Cost can be constraints. They are inherent parameters of the execution of the initiative.

General principle: Time and Cost are always dependent of the Scope and Quality.

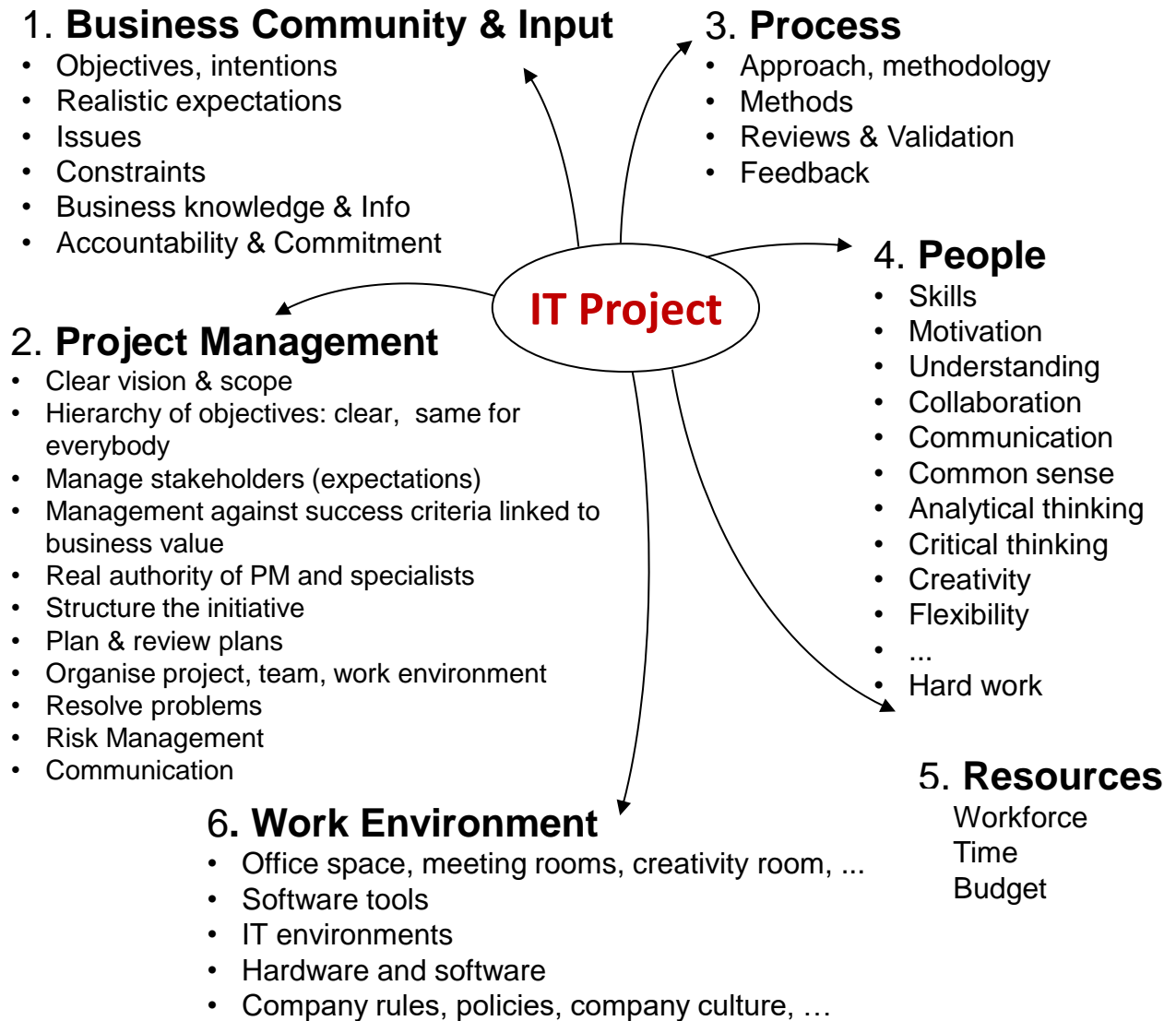
Scope and Quality may be adapted during the project execution. Still then Time and Cost are a function of Scope and Quality.

Factors influencing the progress and success of a project:
(some are external to the project):

- Availability of stakeholders
- Conflicting goals and/or visions among stakeholders
- Time to align visions and get agreements among the stakeholders
- Pace of decision making and quality of decisions
- Frequently changing priorities and changes
- Hidden agenda's
- Collaboration with stakeholders
- Pace of receiving information & quality of received information
- Size of the problem area & solution
- Skeletons in the closets
- Degree of knowledge of systems and organisation/company
- Quality of existing source code
- Collaboration within the project team
- Work environment
- Skills of the team
- ...

"Six Factors Model+1"

Contributing to Project's Success



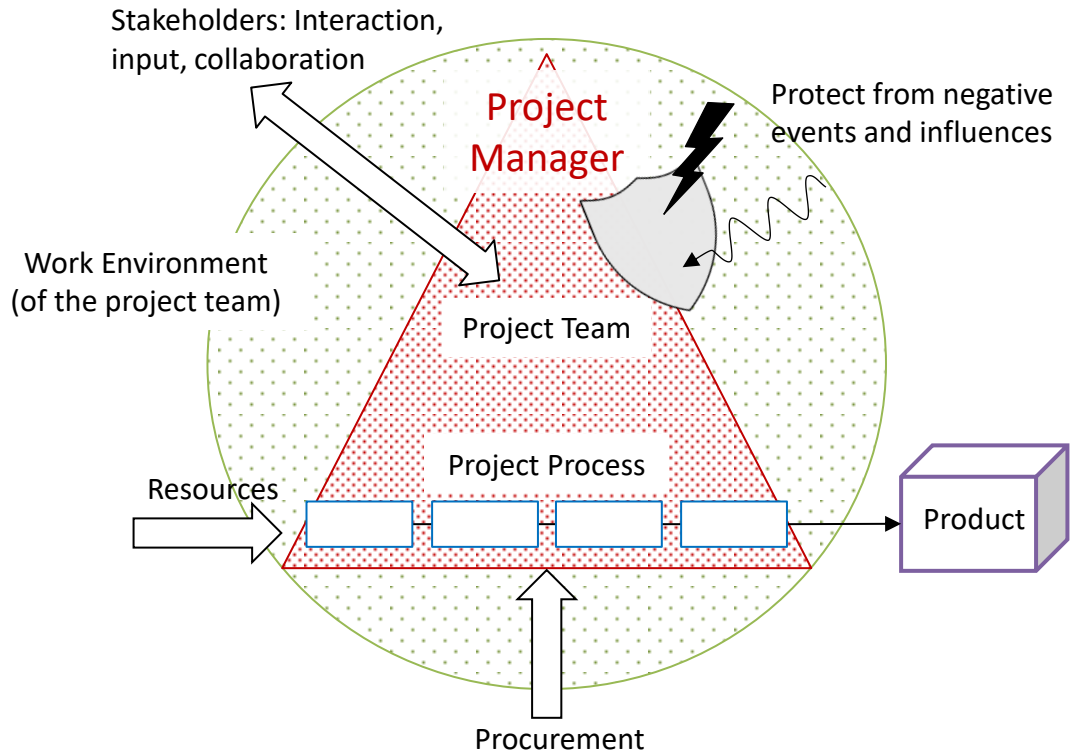
The "+1"

The project **MUST**
respect the **FUNDAMENTAL PRINCIPLES**

of the involved disciplines like

- information systems engineering,
- Information management,
- project management,
- change management, ...

Main Aspects Managed by the Project Manager



The Project Manager

- 1) sets up this **temporary organisation** and **work environment**
- 2) ensures the project fulfils the mission by reaching the objectives (within certain conditions).
- 3) obtains, creates, organises, plans and manages the initiative
- 4) manages Risks, Stakeholders, Communication, Budget and Resources.
- 5) protects the initiative (shield, filter, ...) and remove all negativity in it

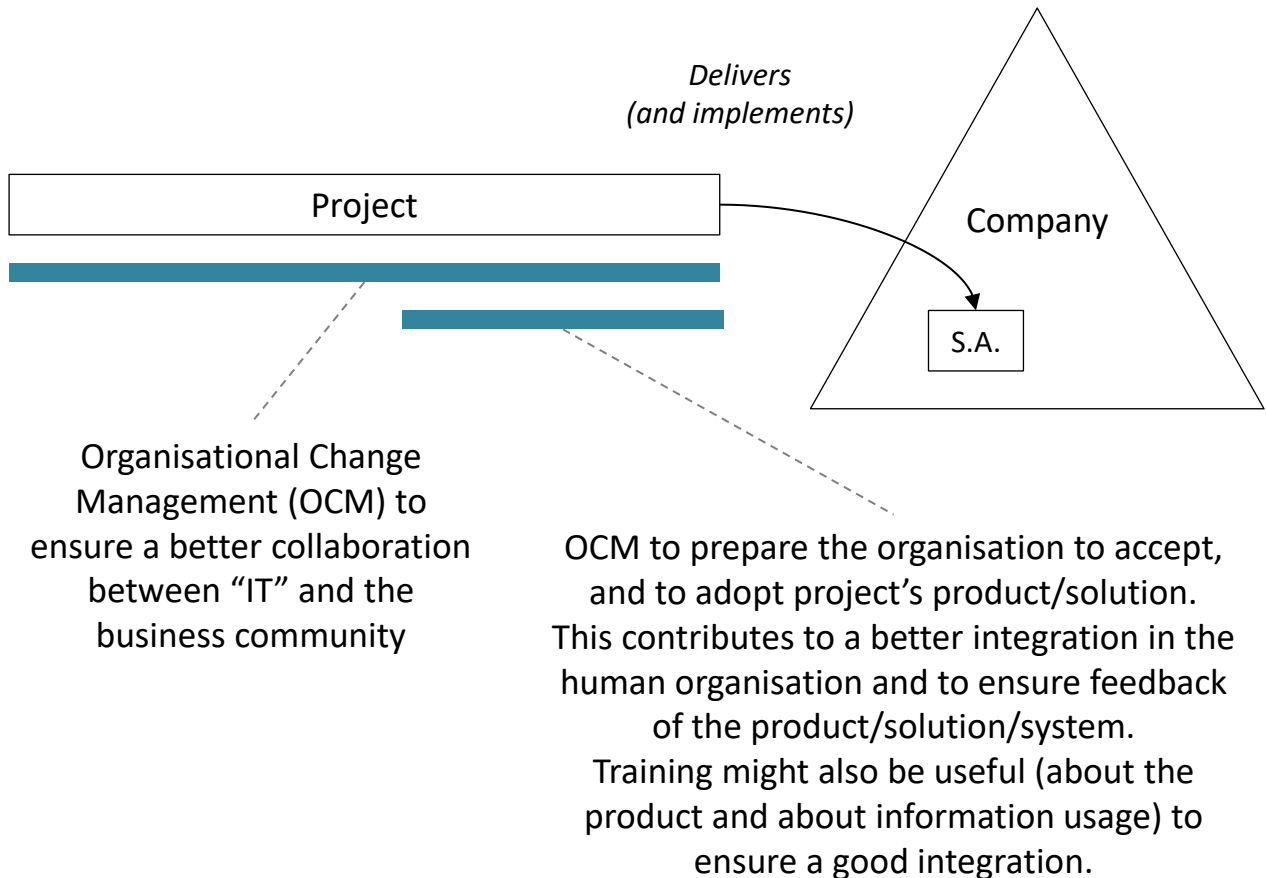
Probably, one of the most important responsibilities and tasks of the Project Manager is to create an organised and safe work environment that suits the project team to work and that allow them to work efficiently.

“What do you (team) need to be able to work ?
And what hinders you?”

Organisational Change Management

18/12/2018

Software Development Projects should be more than implementing and delivering a software application.

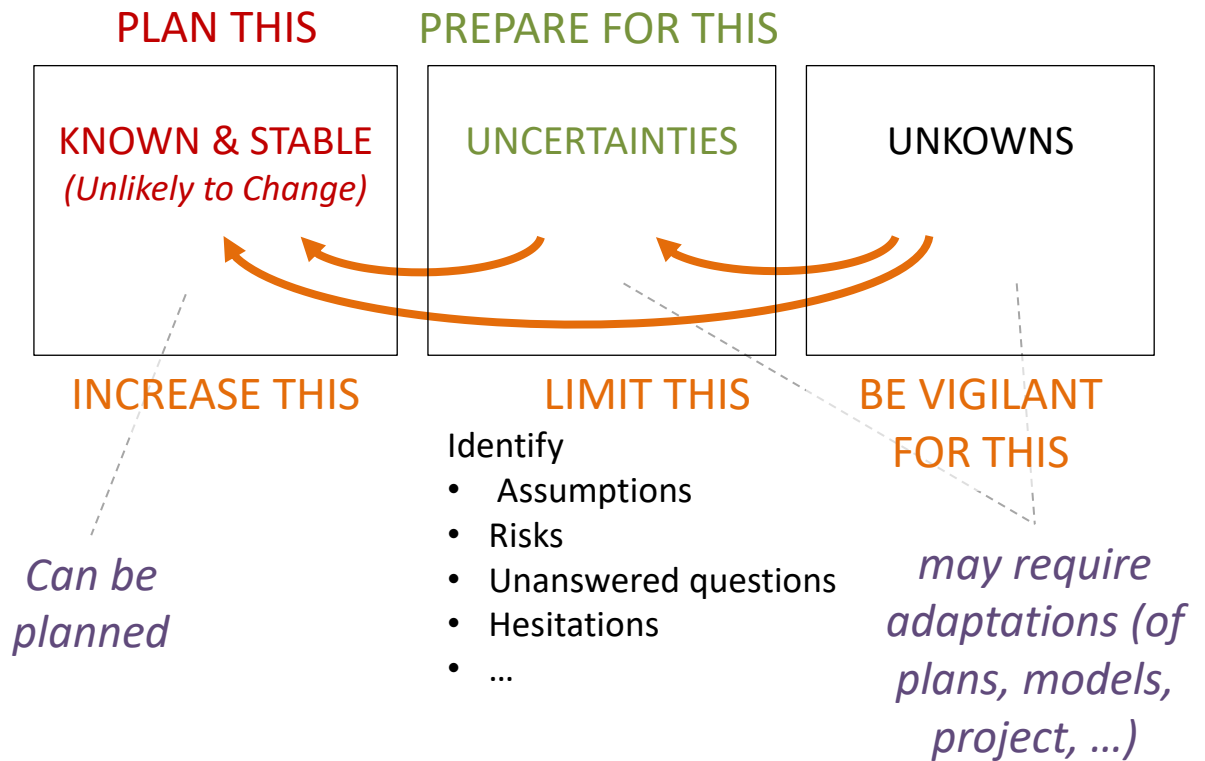


Notes:

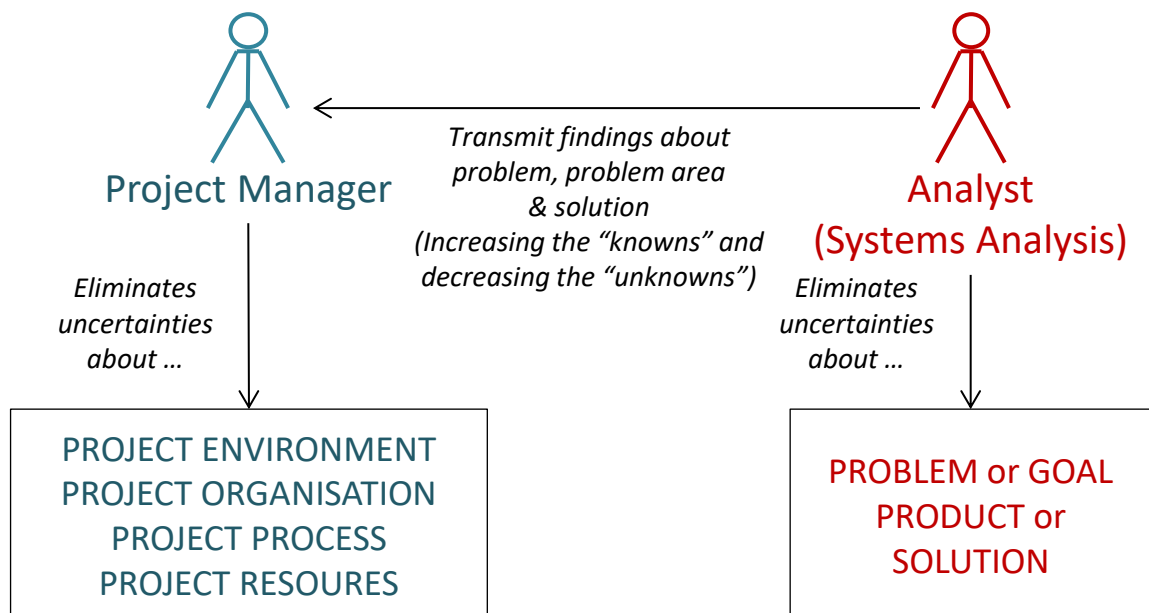
This view should influence the way the project mission and scope are defined. It may also influence the role of the Project Manager and of the Analysts, Architects, ...

Planning & Managing Projects

10/01/2020



Try to turn Unknowns and Uncertainties into Known and Stable elements and then to take the best possible decisions.



Three Main Profiles



Business Leader(s)

Responsible for the goals, for the business vision, for the business logic, ...
Can be manager(s), business architects, SME's,



Solution/System Leader(s)

for the solution and/or system for the product.
(perspective of information exploitation)



Project Manager

Responsible for the project as a temporary organisation, work environment, work and resources.

They must defend the goal and mission; have the motivation, the authority and competencies to ensure intended solution will solve the problem or need or will produce benefits inline with the goals; and they must be very present and accessible.

Other Key Profiles



Security Expert



Technology Expert



....

Note:

This does not imply that these persons must take all the decisions. They may guide, coach and support. They may give freedom to a team or to a few persons. They may ensure a democratic process in a team. Etc.

In the end, they are responsible for the final outcome.

Leadership – Team Management

10/01/2020

	Project Manager	Analysts / Architects
1. Visionary Leadership		
2. Transformational Leadership		
3. Servant Leadership		
4. Coaching Leadership		
5. Pacesetter Leadership		
6. Transactional Leadership		
7. Laissez-faire / Hands-off Leadership		
8. Bureaucratic Leadership		
9. Democratic Leadership		
10. Autocratic Leadership or Authoritarian Style of Leadership		

Style of leadership of the Project Manager can be different of that of an Analyst/Architect since both have a fundamentally different role.

Styles of leadership can be combined depending of the situation.

Leadership – Team Management

10/01/2020

Manager-led Team ≠ Self-managed Team ≠ Self-organised Team ≠ Self-governing Team

	Manager-led Team	Self - managing Team	Self - organising Team	Self - governing Team
Setting overall direction, purpose & composition of team members				
Defining or modifying work processes and policies				
Monitoring and evaluating the work process & monitoring progress				
Executing discrete tasks that make up the delivery of some valuable product / service				

Manager

Team

What if ...

- a manager delegates decision making to the team and coaches and guides the team in their decision making process ?
- the team defines the work process (approach, methodology) and policies but a manager manages its execution ?
- the composition of the team is defined by the team, but the team is managed by a manager ?
- some important directions and purposes can be defined by a team, but not all ?
- etc ...

PM – Analyst Collaboration

30/10/2018

PROJECT MANAGER

ANALYSTS - ARCHITECTS

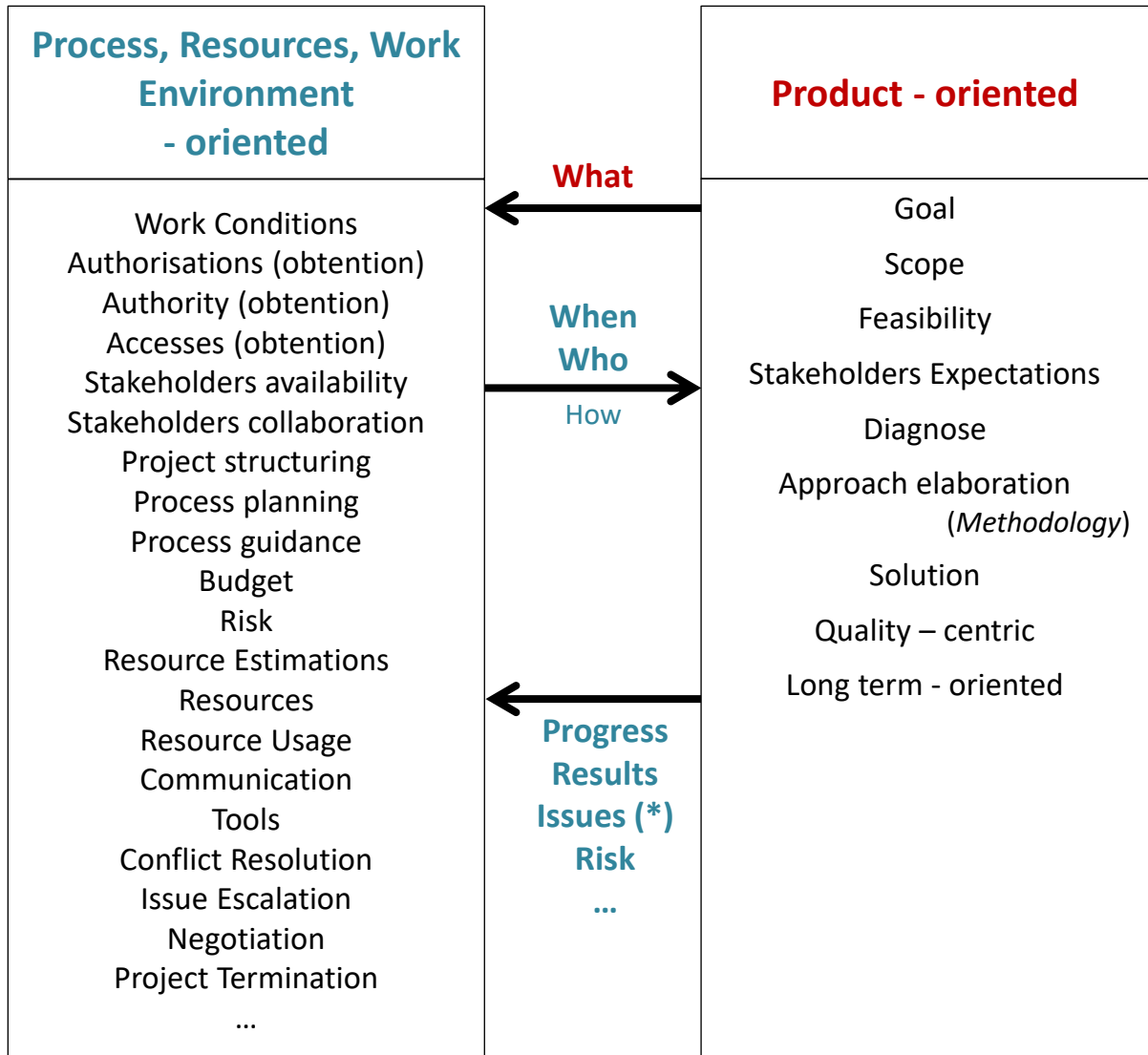


Diagram represents the general idea

Notes:

Some tasks, like Scope definition, may, originally been defined prior to the project start.

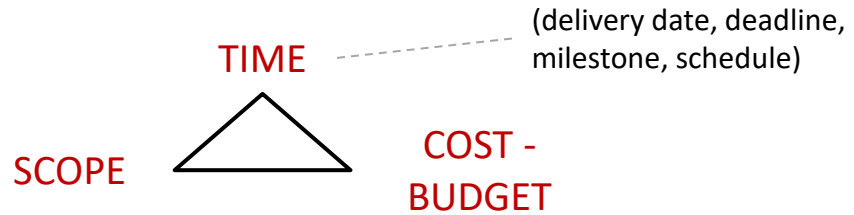
Examples: The Analyst notices that a scope change might be advantageous. This affects the agreements, plans, required resources, ... The Analyst has to discuss this with the PM. The stakeholders will also be involved in a discussion.

The Analyst detects a risk. This has to be discussed with the PM to see what measures can be taken.

*: The issues the Analyst discusses with the PM are essentially those which impacts project management. The Analyst may inform the PM about some problems related to the logic and product, yet they have to be solved by the Analyst. A PM doesn't solve an issue in process modelling or information modelling.

Iron Triangle

17/09/2019



MEANING:

- A project delivers a product/solution accordingly to the scope. To produce this product/solution time and budget (cost) is required.
- These 3 constraints are related to each other
- A change in one of the three changes, affects the other two parameters.

It does

- not imply that the scope is fixed
- not define the priority among the constraints
- not imply that all three constraints are fixed (frozen, can't be changed)
- Not imply that delivering accordingly to these constraints a solution is delivered that solves the problem, that this solution is beneficial or that the best solution has been delivered.

CRITIC:

The aspect of created quality and risk avoidance are not included.

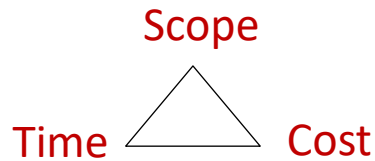
A better alternative:



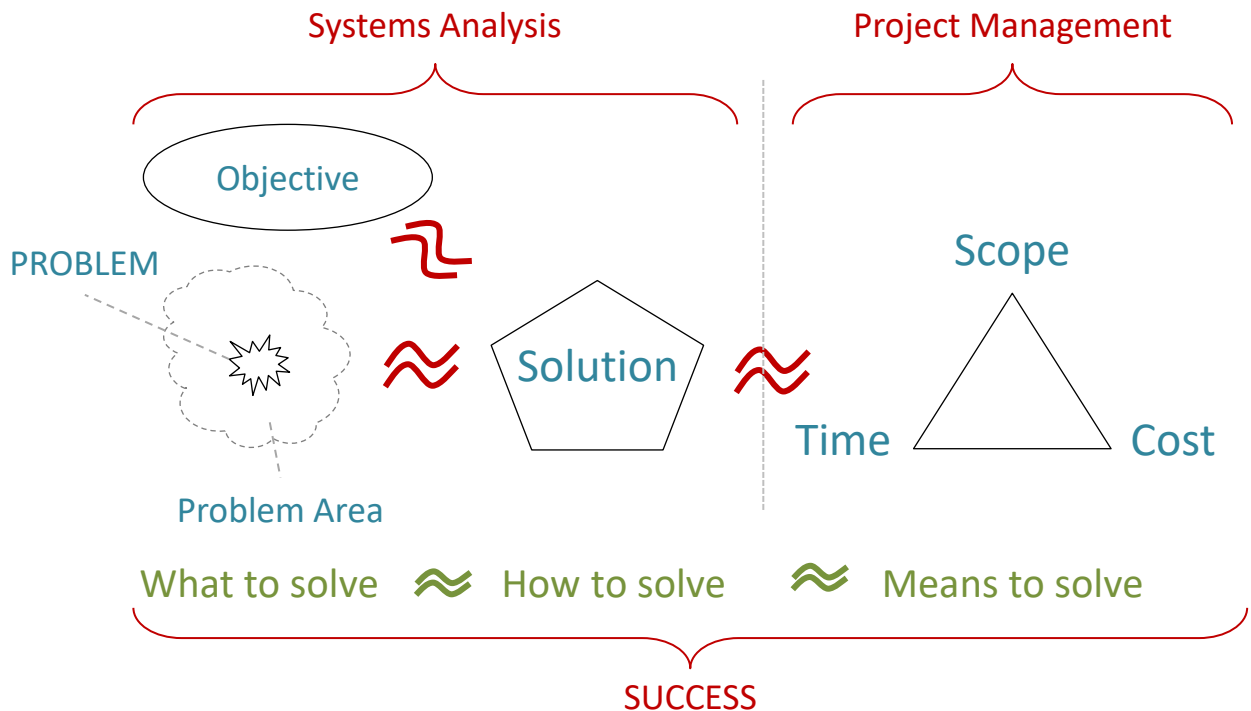
A deadline can be the result of an agreement based on estimations, on preferences
Or it can be an imposed date.

Iron Triangle

10/01/2020



Broader Picture



- The **Objective** and/or **Problem** (including the Problem Area),
- the **Solution**
- and the Iron **Triangle**

must be

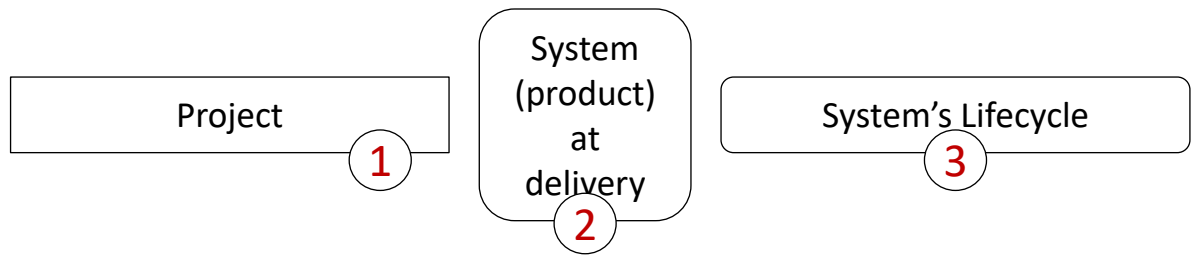
- **correct and/or feasible**
- and have to **correspond** with each other

during the **WHOLE COURSE** of the project.

if Objective/Problem and Solution are wrong → Iron Triangle = worthless

Notes:

Note that the scope is not the solution or the demand. The Scope only delimits the problem and the solution.



Iron triangle gives only a partial view on this

1) How well has the project been executed?

Delivering according to scope and within time and budget.
Other criteria/indicators can be found.

2) How well is the product at delivery?

Does it matches the expectations?
Does it meet the demand?
Does it correspond with the requirements and specifications?
Does it integrates well?
Does it work ?
...

3) How well is system performing during its lifecycle (lifetime)?

Does it create value?
Is it able to deal easily with many unforeseen events?
Is it flexible?
Is it reliable?
Is it manageable?
Is the quality maintained?
Are the performances stable, irregular or declining?
Is the system under control?
Is the system easily adaptable?
What is the Total Cost of Ownership?
What is the created value ?
...

Priorities among the constraints of triangle depend of the product and situation.

The priorities are INDEPENDENT of the chosen approach.

ALWAYS Prioritise the Triangle

Always have at least 1 adaptable constraint

Highest Priority	Explanation
Value - driven Goal, Quality	Minimal value to be delivered (at least this or more) Creating value has the highest priority. Cost and time are adjustable to create the maximum of value. If critical applications, long term, sustainable, has to evolve, to be built on top, to be integrated, ...
Time-driven	Delivery at deadline or earlier. For events, one shot
Cost-driven	Deliver accordingly to budget (or cheaper). One shot, (really) temporary, "cheap" is expensive.

Project Manager, Business Stakeholders, Architects and Analysts should always decide together at the start of the project :

- What are the priorities? What is the most important?
- If one had to change, which one would it be?
- Main stakeholder may need advice about the priorities.
- The stakeholder's perspective is only one perspective. (S)he may not understand all the factors, aspects, risks and implications of a choice.
- If an opportunity or issue occur, decisions will always be taken against these four aspects.
- Although constraints and their priorities should be defined as well as possible to avoid changes, if it appears that it is advantageous to bring in change, then this should be done (being awareness of the benefits and consequences and adapting the project to it).

Fix all Three (or Four) constraints is a Recipe for Failure !!!

QUESTION 1: Heart surgery is necessary. The patient is offered for the **SAME** operation two options:

- a) operation of 30' and Cost 500 €
- b) operation of 1h30' and Cost 1000 €

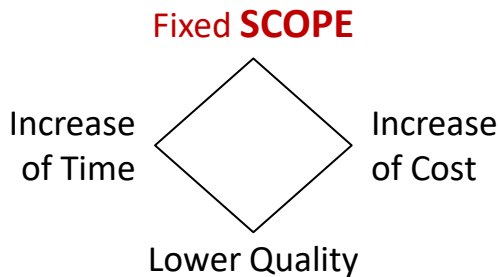
Difference: in (a) the surgeon works very fast. In (b) (s)he has the time to verify, to do things right, ...

What would you chose if you were the patient (provided you can afford the 1000€)? Why? What is the lesson?

QUESTION 2: A team gets a software development mission.

1. It delivered a valuable, high quality, innovative, software system after 10 months.
2. It built an awkward system with questionable purpose and value, delivered on time and on budget but with a reduced scope.
3. The team could, together, overcome unexpected challenges, had the opportunity to do things they didn't think they could do, to learn and to conceive and create a valuable good system without having to worry too much on scope, time and budget.

In which case would the team have the most satisfaction and pride? Why? In what order? Why? What is the lesson?



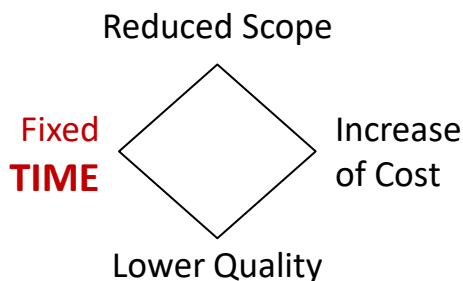
Companies work based on a vision, on strategy, on goals and targets, on production of value, on limiting risks and costs, on increase of benefits and on regulations.

In a natural way, this leads to the submission of project proposals. The product or outcome is the primary reason why projects are executed.

It is obvious that scopes define the missions of projects. Time and cost are deduced from the product scope.

What happens if a project is running late and the scope is fixed?

Time will increase. Cost will increase. The quality can be lowered, sometimes in an hidden way. This creates debt. Or it can be a combination of these.



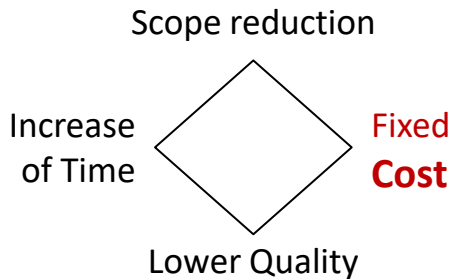
Time is never the trigger and driver for the start of a project. It can be an imposed criteria (legal obligation, celebration day, ...). A client is rarely ready to invest in a project of which (s)he doesn't know what (s)he will get, and thus what will be the return. Making some vague estimations is always possible. (see predictability of constraints)

What happens if a project is running late and the delivery date is fixed?

The scope can be reduced. The cost may increase. The quality can be lowered, sometimes in an hidden way. Consequences of a lower quality will come later. Or it can be a combination of these.

Note:

- When something is "fixed", this means that it is the main constraint. It is the last one that should be adapted. And in some case, it won't be possible to adapt it, although those cases are rare.
- In most cases, adaptations of the constraints is not desirable unless it is really beneficial. 727

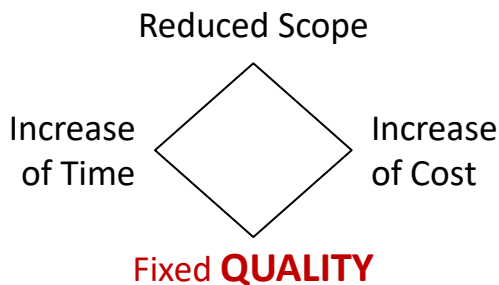


Cost is never the trigger and driver for the start of a project. Sometimes it is possible to find additional budget. This depends of the expected benefits of the project. A client is rarely ready to invest in a project of which (s)he doesn't know what (s)he will get. Making some vague estimations is always possible. (see predictability of constraints)

It is obvious that scopes define the missions of projects. Time and cost are deduced from the product scope.

What happens if a project is running late and the COST is fixed?

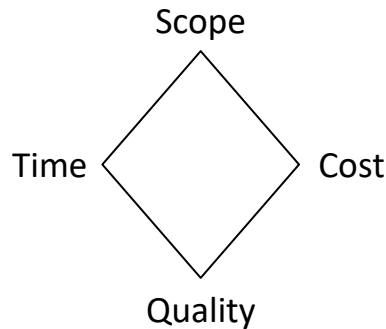
Scope can be reduced. Time may increase. The quality can be lowered, sometimes in an hidden way. This creates debt. Consequences will come later. Or it can be a combination of these.



Quality is never the trigger and driver for the start of a project. However, quality can be a non-negotiable constraint. A client is rarely ready to invest in a project of which (s)he doesn't know what (s)he will get, and thus what will be the return.

What happens if a project is running late and the quality is fixed?

The scope can be reduced. The cost or time may increase. Or it can be a combination of these.



We can negotiate on 5 aspects.

So, there are SIX questions

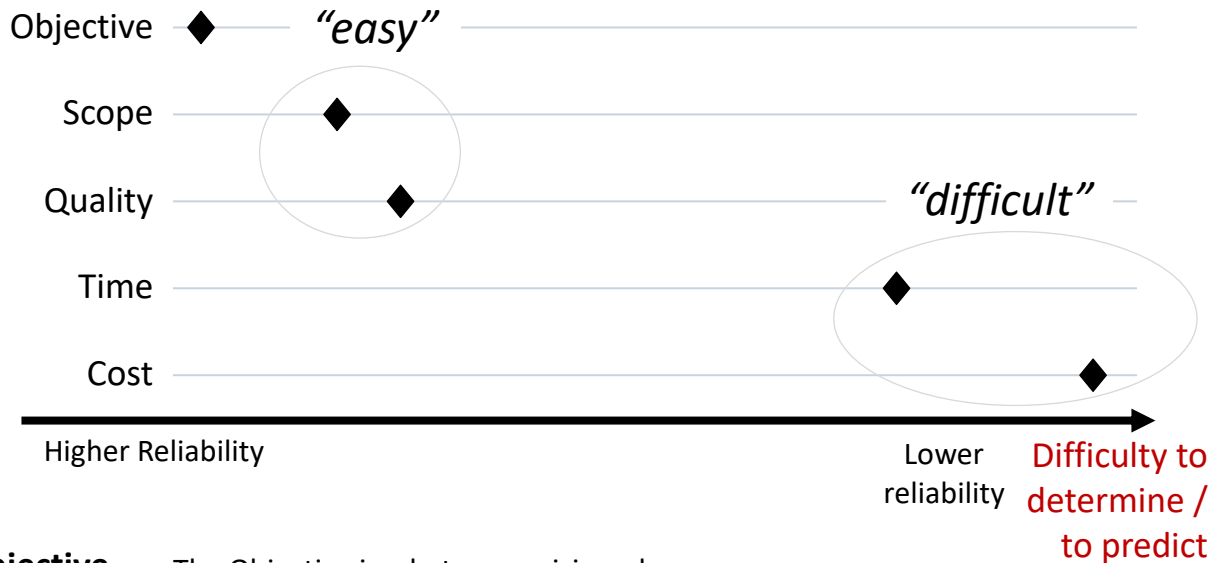
1. Can we adapt the scope? For example, by postponing some parts.
 2. Can we simplify the solution while keeping the scope?
 3. Can we reduce the project duration?
 4. Can the cost be reduced?
 5. Can we be satisfied with reduced quality and where can the quality be lowered and until what point?
 6. Or can any of the answers to the question above be combined to have a satisfactory result?
- It is advisable to try to do a good analysis and to estimate better in the first place and to reduce everything that may pose delays.
 - A project runs out of scope, time or budget for a reason: inefficient analysis, bad estimations, bad leadership, bad planning, work environment, insufficient collaboration, received information, slow or bad decision making, ...

Maybe first find and deal with the causes rather than trying to solve the consequences. Only then we can learn valuable lessons !!

- What if we could assign lesser importance to the iron triangle and more focussing on the product?

Difficulty of Predictability of Project's 5 Main Parameters

14/02/2020



Objective The Objective is what we envisioned.

Scope The Scope can be determined (if we focus on what is necessary, rather than on wants and if SASD is done decently)

Quality Required Quality is relatively easy to be determined.

Time Time = mainly $f(\text{Scope}, \text{Quality})$
+ non-linearity + uncertainties + quality of organisation, collaboration, information, ...

Cost Cost = mainly $f(\text{Scope}, \text{Quality}, \text{Time})$

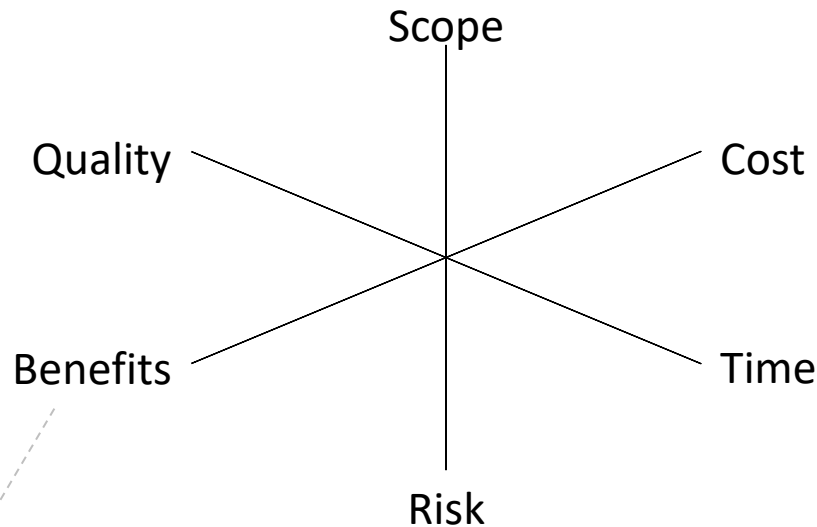
The product, as defined by the objective, the scope and the quality create satisfaction. They create the value.

Time and Cost can be constraints. They are inherent parameters of the execution of the initiative.

General principle: Time and Cost are always dependent of the Scope and Quality.

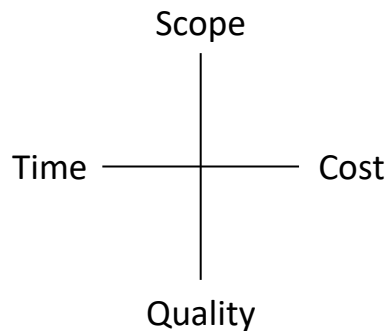
Scope and Quality may be adapted during the project execution. Still then Time and Cost are a function of Scope and Quality.

Actually, A project continuously, from before it begins until it ends, balances 6 main constraints.



Benefits:

- What is the alignment with goals and strategies?
- What value does the product or outcome create?
- What advantages and new opportunities are created?
- What new hindrances and limitations are created?



We can negotiate on 5 aspects.

So, there are SIX questions

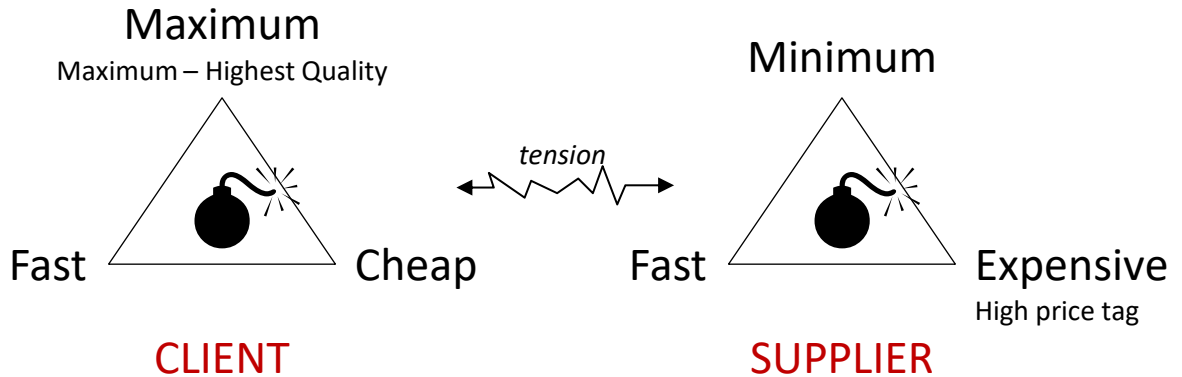
1. Can we adapt the scope? For example, by postponing some parts.
 2. Can we simplify the solution while keeping the scope?
 3. Can we reduce the project duration?
 4. Can the cost be reduced?
 5. Can we be satisfied with reduced quality and where can the quality be lowered and until what point?
 6. Or can any of the answers to the question above be combined to have a satisfactory result?
- Better is to try to do a good analysis, better estimations in the first place and to reduce everything that may pose delays.
 - A project runs out of scope, time or budget for a reason: inefficient analysis, bad estimations, bad leadership, bad planning, work environment, insufficient collaboration, received information, slow or bad decision making, ...

Maybe first find and deal with the causes rather than trying to solve the consequences. Then we can learn valuable lessons !!

- What if we assigned lesser importance to the iron triangle and more focussing on the product?

Tensions between Triangles

10/01/2020



Client's Triangle (drivers)

Clients desire appropriate, effective and efficient well-designed systems. They desire solutions that solve their problems and which create benefit for their organisation.

But they want it fast and cheap. Clients usually don't have solid competencies in software engineering. They are unable to estimate what a near optimum solution is and what it takes to build it. As a result, they may easily underestimate it.

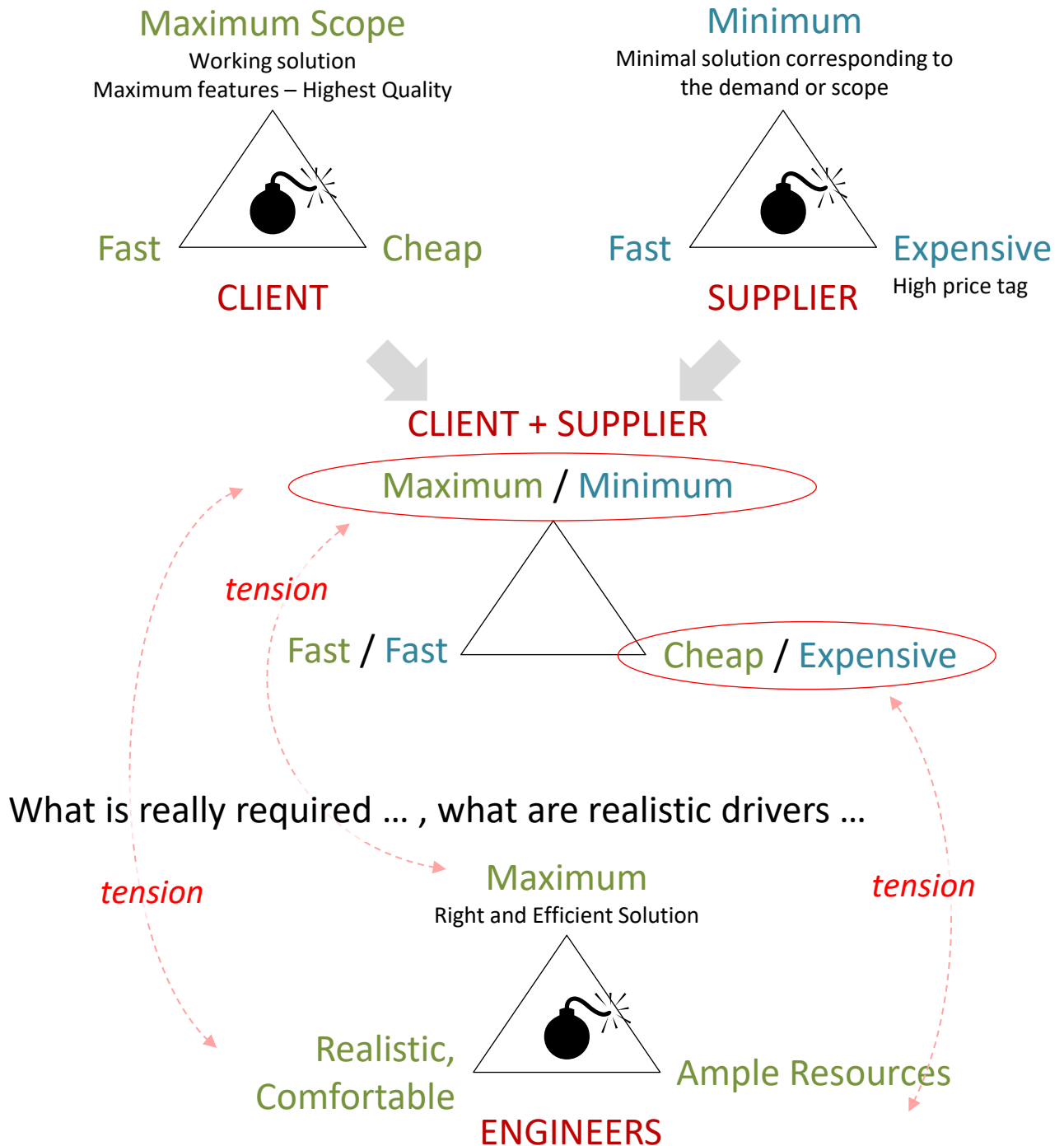
The desire to have a result quickly and at low price will then undermine the conditions required to build a good system and a real advantageous solution. This is particularly true when they see the project as a cost.

Supplier's Triangle

Suppliers seek to sell a project at the highest price. They will seek to minimise cost and delay. A shorter delay means that more projects can be executed. As a consequence, they may be tempted to deliver the minimum that still complies with the demand. Some suppliers may genuinely want to help the client and deliver better and truly valuable solutions.

Tensions between Triangles

10/01/2020



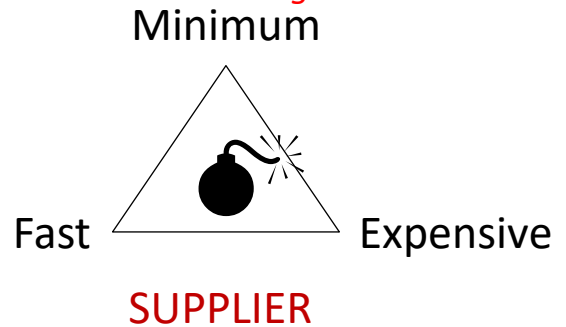
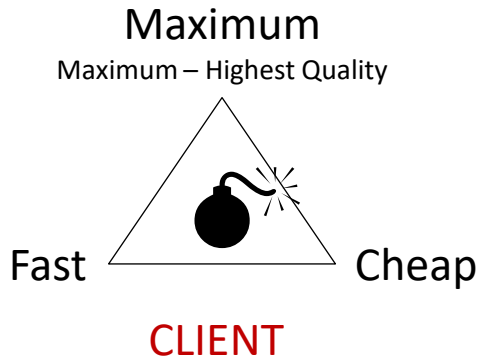
What is really required ... , what are realistic drivers ...

The Client's differences between the Supplier's objectives create obvious tensions. But even if they agree, their agreement may still differ from the objectives and needs of the systems analysts, engineers and builders. This agreement then constitute a threat. It is not favourable to create the right work conditions (pressure, lack of resources, feasibility, ...) and constitute a risk for the engineers and for the whole project.

Tensions between Triangles

10/01/2020

Page to be reviewed



Delivering a maximum at a cheap price won't satisfy the supplier and the engineers. Supplying a minimum at a high price may lead to a product that doesn't solve the problem and the client may not be glad about the price.

Both, client and supplier, may agree to execute the project quickly ("FAST")

Reduction

Reduction of cost, effort, time is a receipt for disasters. In a short time, it may seem to save money or time. Soon this may turn out to be much more expensive.

Short-cuts

Looking for short-cuts to speed up the development process is a losing strategy

Diluted version

Faking, imitating, cargo cult, pretending: Often artefacts, products, documents are produced without doing the real work, without understanding the true purpose of it, its meaning and value, without the real substance.

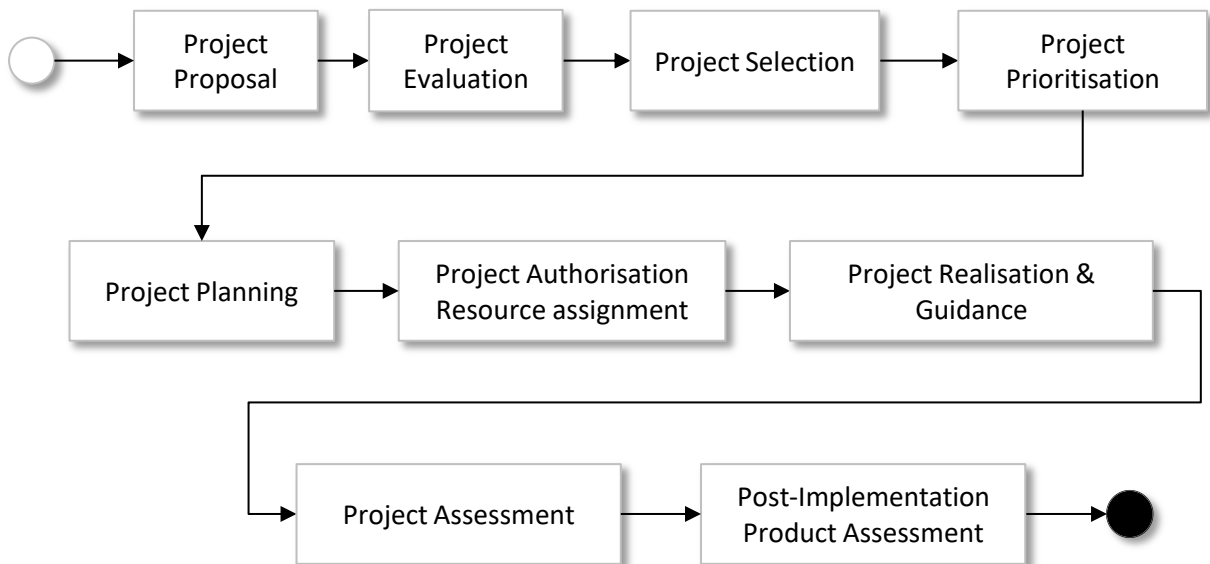
Low Quality

The required level of quality is often underestimated. High quality is not wanted because it is expensive and because it is confused with luxury.

Notes:

There are true techniques that facilitate or increase development speed. Some ways, like CBD, increase the speed in the longer term. Real professionals can make a difference between short-cuts and true techniques.

Project Overall Process



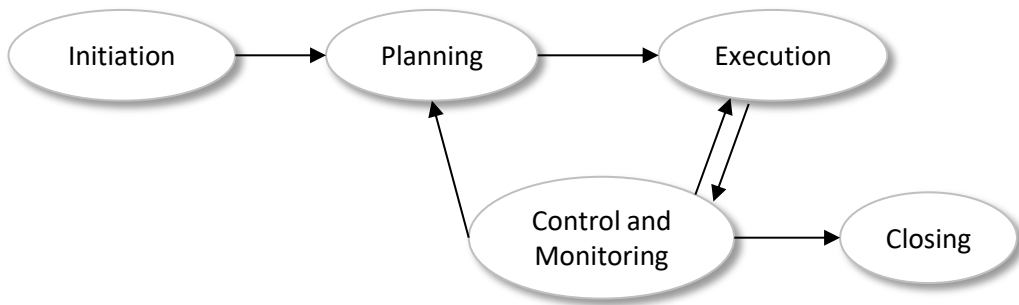
Executing a Project?

30/10/2018



Notes:

Other aspects to be considered: strategic fit, criticality, priority related to other projects



1. Project Management **Integration**
2. Project **Scope** Management
3. Project **Risk** Management
4. Project **Time** Management
5. Project **Human Resource** Management
6. Project **Procurement** Management
7. Project **Cost** Management
8. Project **Quality** Management
9. Project **Communication** Management

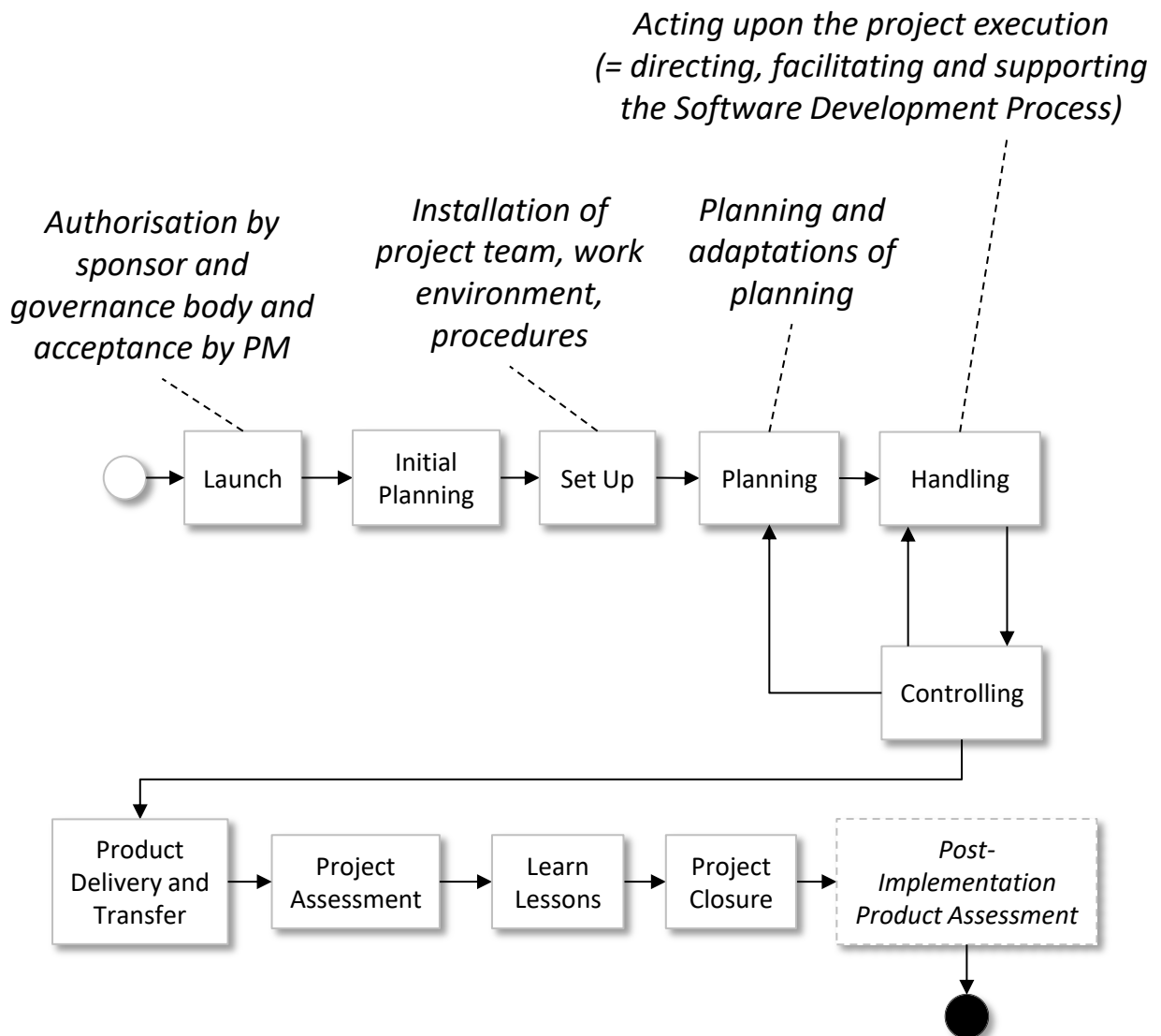
Source: Project Management Institute : Guide to the Project Management Body of Knowledge

Project Management in a Nutshell 30/10/2018

Initiation	<ul style="list-style-type: none"> Check mission, objectives Check management support Check feasibility and risks Identify stakeholders (role, alignment of objectives, ...) Obtain authority and resources Develop the Project Charter
Planning	<ul style="list-style-type: none"> Structure the project Define the WBS Develop a schedule Estimate resources Estimate costs and plan budget Plan quality Plan human resources Plan communication Plan risk management Plan procurement Review and adapt plans Align plans
Execution	<ul style="list-style-type: none"> Acquire, develop and manage project team Direct the set up of the work environment Conduct procurements Manage the work environment Manage stakeholders and their expectations Perform quality assurance activities Manage changes (eg inspect and negotiate changes) <p>Execute the project (analysis, design, software development, testing, migration, implementation)</p>
Monitoring	<ul style="list-style-type: none"> Control scope and progress Monitor and control resource usage Monitor the work environment and the project team Identify, monitor and control risks Identify need for change Perform quality control
Closing	<ul style="list-style-type: none"> Learn lessons Close the project

Project Management Process

30/10/2018



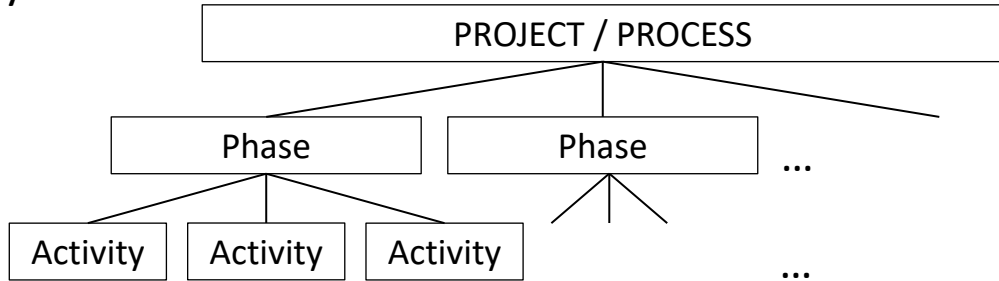
Project Phase & Activities

30/10/2018

Traditional Representation in Methodologies



Analysis:



In Software Development ...

Project Phases

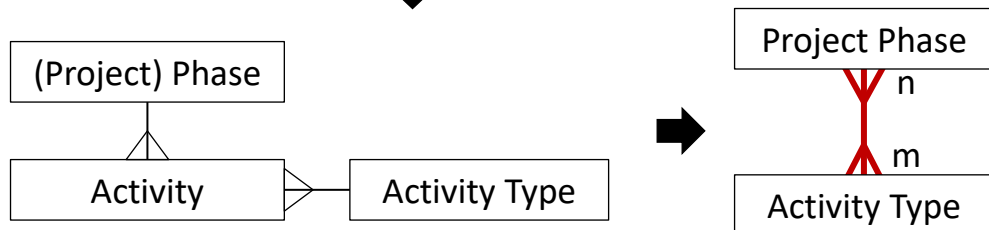
Analysis
Design
Programming
Testing
Implementing

?

=

Activity Types

Analysis
Design
Programming
Testing
Implementing



- Design activities can be performed during an Analysis phase
- Analysis activities can be performed during a Design phase
- Programming can be done during a testing phase (for example, to correct bugs)
- Analysis can be done during a Testing phase to know how to correct a bug in the higher logic.
- And so on ...

Project Phases do / are

- Indication of the main **FOCUS** and **LEVEL of THINKING** now required
- Indication of the **MAIN TYPE of ACTIVITIES** currently being executed
- Indication of **TYPE of OUTPUT** to be expected soon
- Indication of the **PROGRESS: WORK** (and **TYPE of WORK**) that has been done and what is being done
- Facilitate the **MANAGEMENT** of projects
- Result from a Project Management Decisions

Project phases do NOT/ are NOT

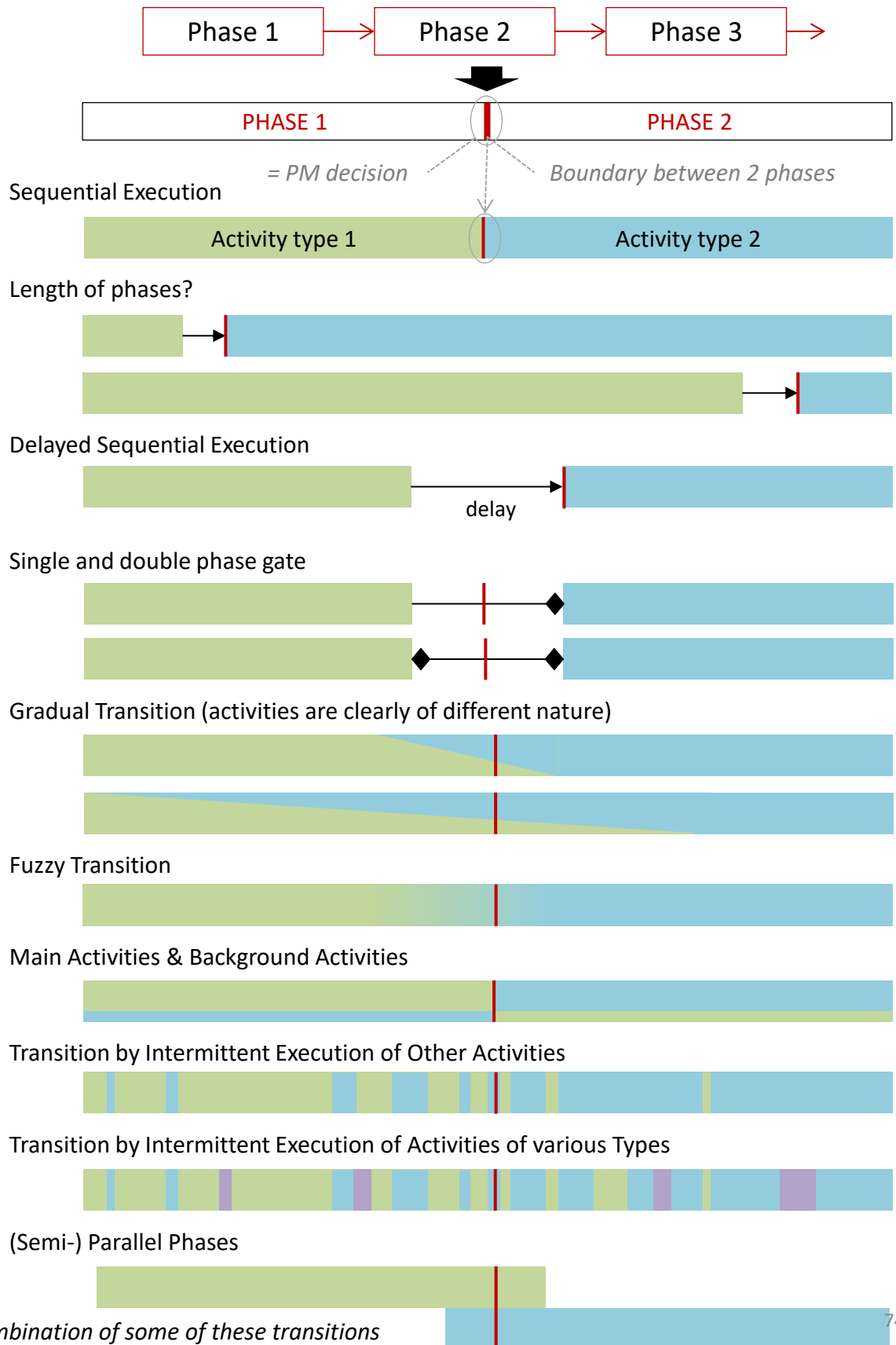
- Do not limit the type of activities being done
- Not meant to provide a precise measure of progress (in time, cost or effort). Other measures do.

A **PHASE** is also a **BUILDING BLOCK** that helps to structure the software development approach. Each phase has its **purpose** and **goal** and can be applied on a subject, component, issue. A phase can also be **executed at a certain level of detail** or from a **certain perspective**. The phases can be defined and assembled in many ways.

Phases offer flexibility and scalability to projects

Phase Transitions & Activities

30/10/2018



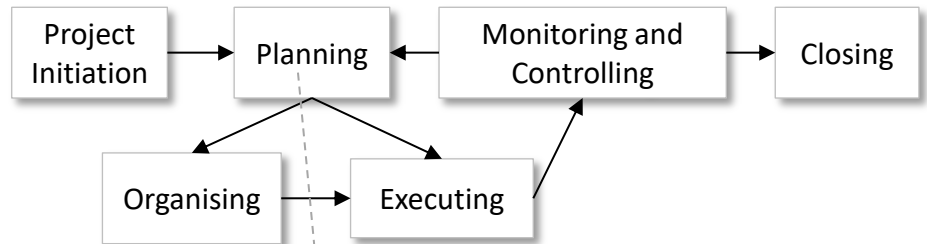
Project Phases ?

30/10/2018

Project Management Methodology

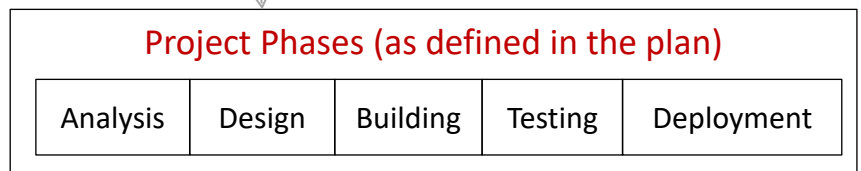
(simplified framework)

"How to manage a project?"



Project Management Vision on the Project's Process

*How PM sees, organises and plan the project.
(example: Deciding to move the project to next project phase)*



PM Perspective: PM Purpose, PM decisions, PM Needs

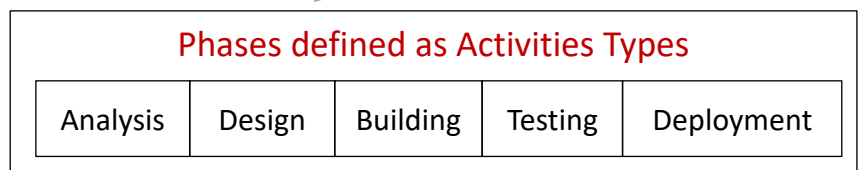


IS / SW Dev. Perspective

= some correspondence

Software Development Perspective on the development process

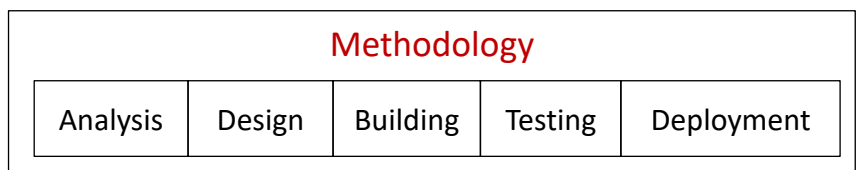
*What project members do.
Nature of activities, focus,*



Project team's perspective:

What is necessary to be done, what types of activities are executed now.

Standard Methodology



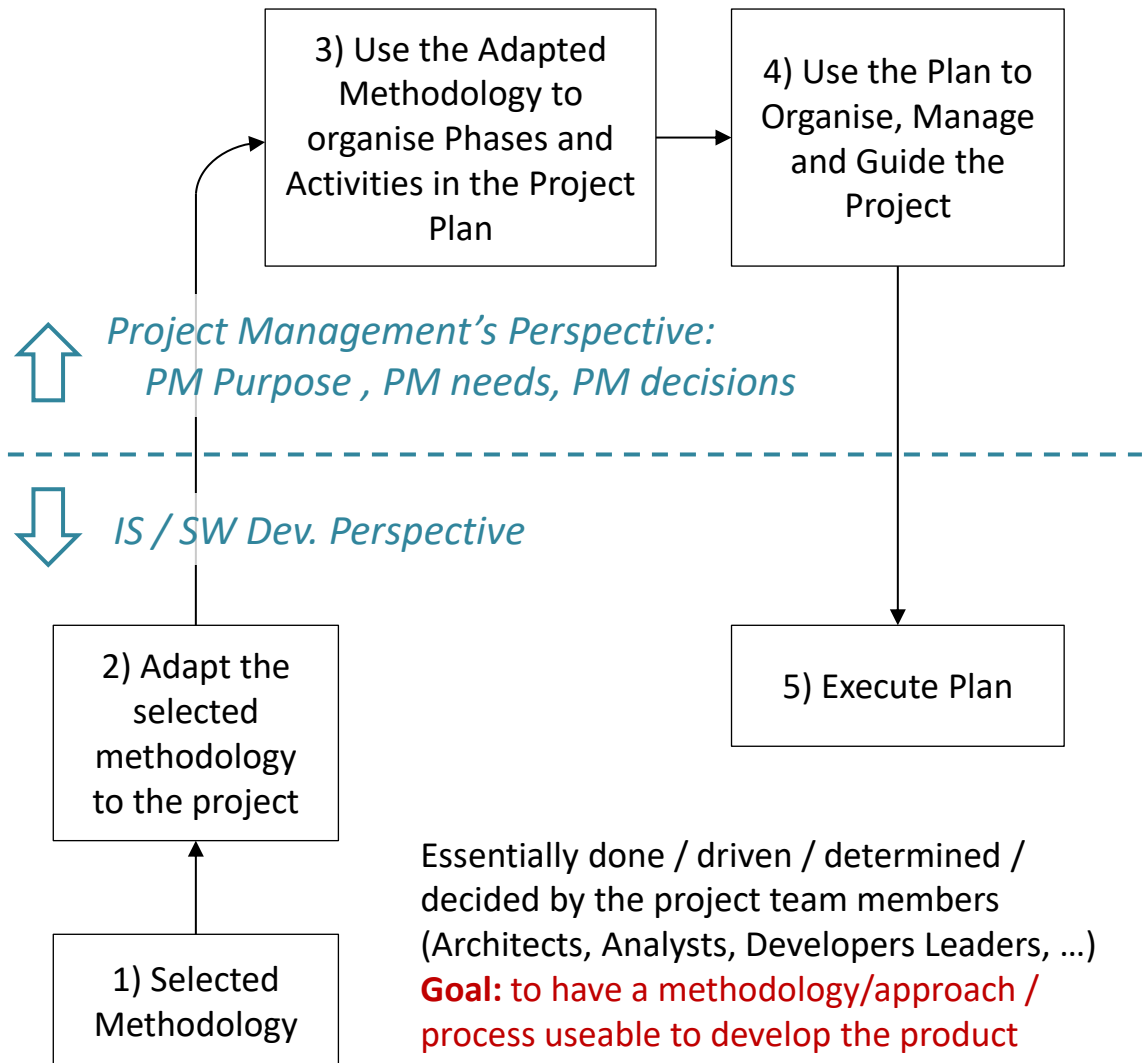
The standard, the template, the theory, the whole set of artefacts proposed as a basis for IS/software development endeavours

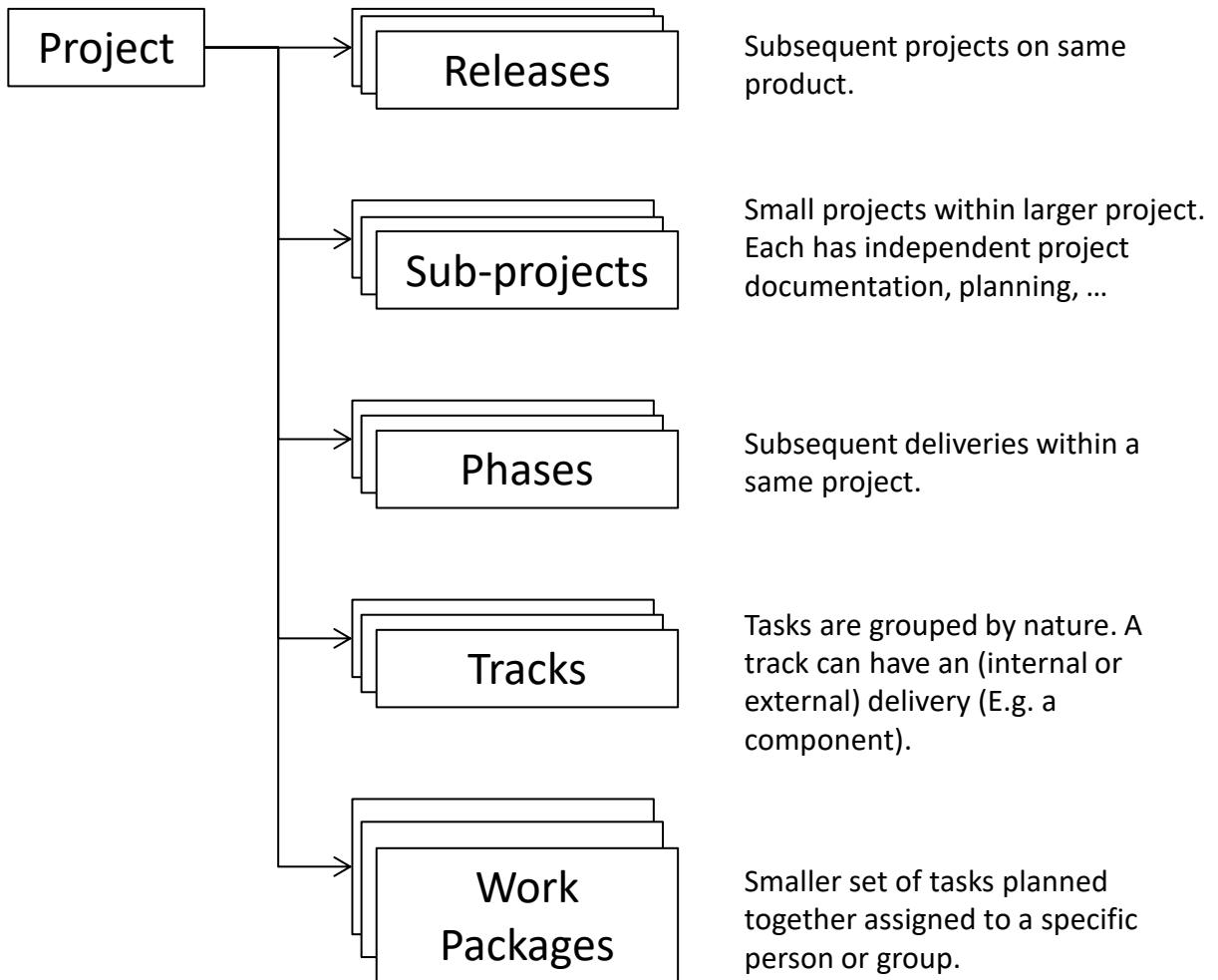
Project Phases ?

30/10/2018

Lead by the Project Manager
(with team input and collaboration)

Goal: to have a plan useable to manage the project





**STRUCTURING A PROJECT
MAKES
THE PROJECT SCALABLE**

Notes:

Structure is the key to scalability.

Without clear structure, no scalability is possible.

Structuring the Project - WBS ^{30/10/2018}

The **WBS** (Work Break-down Structure) is a decomposition of all the work to be done by the project. It is used to estimate the necessary resources and time and then to schedule and staff it.

The **WBS** can be **organised in different ways:**

- per **sub-project**, track, ...
- **component**-based
- by development **phase**
- by **nature of work – output / deliverable**-based

Example: All test activities are gathered together in one branch of the WBS. All Analysis activities are gathered in another branch. All this, regardless of the phase in which the work is executed.

- by **executor**
- ...
- Or a mix

Remark:

The WBS may or may not reflect the work really executed in a project phase.

An alternative is a structure formed by the decomposition of the product into components.

Choice to be discussed:

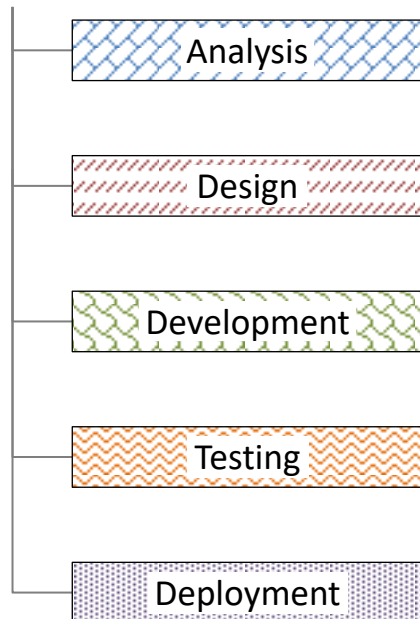
- What suits the project the best ?
- What is the easiest to manage the project?
- Impact reporting !

WBS – Gantt - Phases - Execution

10/01/2020

Activities gathered by type in the
Work Break-down Structure

Project's WBS



Used to make
(staffed) schedule
(Gantt-chart)



Activities as they are executed



Project Phases



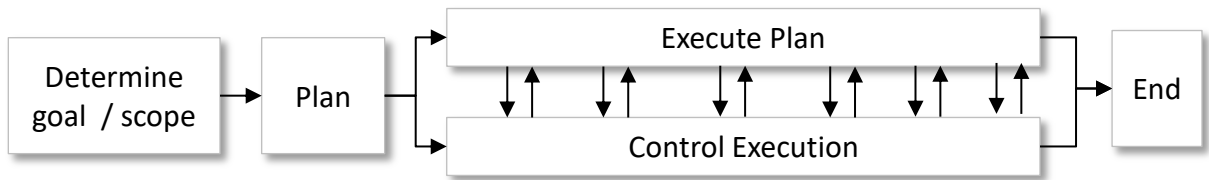
Analysis

Design

Development

Integration

Deployment



Why this doesn't work ?

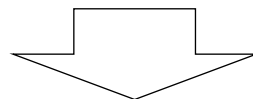
Assumes :

- All information necessary for planning can be gathered and taken into account
- The planner excels in planning
- The plan is always right.
- The execution of the project (the reality) can and has to be adapted to the plan.
- People's mind (thinking, creativity,) works at a linear pace
- People work and collaborate at a linear pace
- No skeleton in the closet or other difficulties, obstacles, setbacks
- The objective (or obligation) is to follow the plan
- Executing "as planned" leads to the achievement of the objective

Ignores :

- Uncertainties and risks are greatest at the beginning of the project (exactly when the project plan is defined)
- Limited understanding of the initiative in the beginning of the project
- Limited skills of the planner
- Unforeseen events may happen

PLAN = OBLIGATION



PLAN = PRISON

REAL Cause of Death March

How to prevent this?

Why do Planning Fail?

- We underestimate the work to be done (size, complexity, ..)
- We overestimate ourselves (competencies, knowledge, insight, abilities, ...)

How to execute a project successfully?

Multiply the estimates by 10 (or by 2)

Then ..

- Why to make a plan?
- Will it solve the estimation abilities, the abilities to manager buffers and the project management skills ?
- = symptomatic problem solving

Wrong ways of using a plan:

- Not using a plan. Leave it on the disk or in the drawer.
- Freeze the plan and stick to the plan.

RECIPE FOR FAILURE :

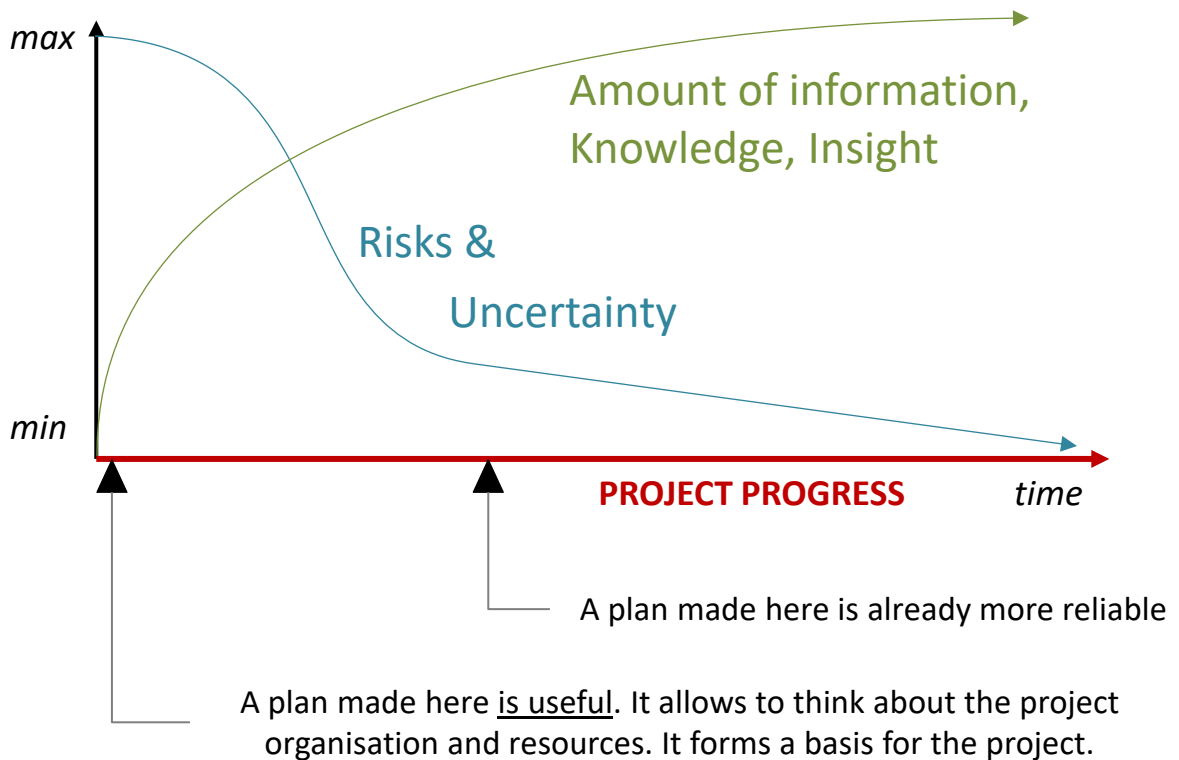


based on a lack of understanding the fundamentals and essentials of planning.

"An idiot with a plan can beat a genius without a plan" (Warren Buffett)

provided the idiot's plan isn't too bad and (s)he knows how to use plan correctly

Why NOT FREEZING a plan made early in the project?



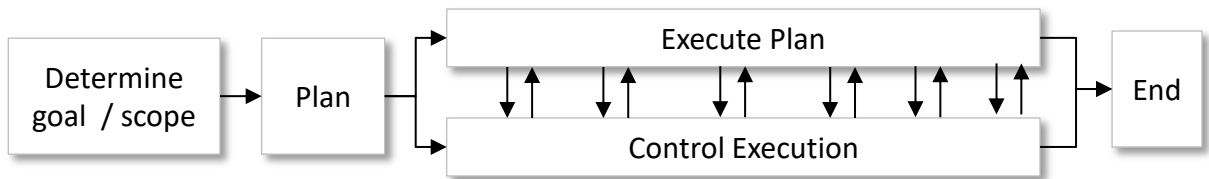
A Plan is **PREDICTIVE AND ADAPTIVE**

Any plan made in a situation with reduced information; a lot of uncertainties, unknowns, risks, unsettled issues and/or for an endeavour and environment that isn't under full control, is very likely to change. It will have to be adapted as more is known, as more is being settled, as the situation unfolds. It has to evolve to continue to be as realistic, feasible as possible and suitable to the situation.



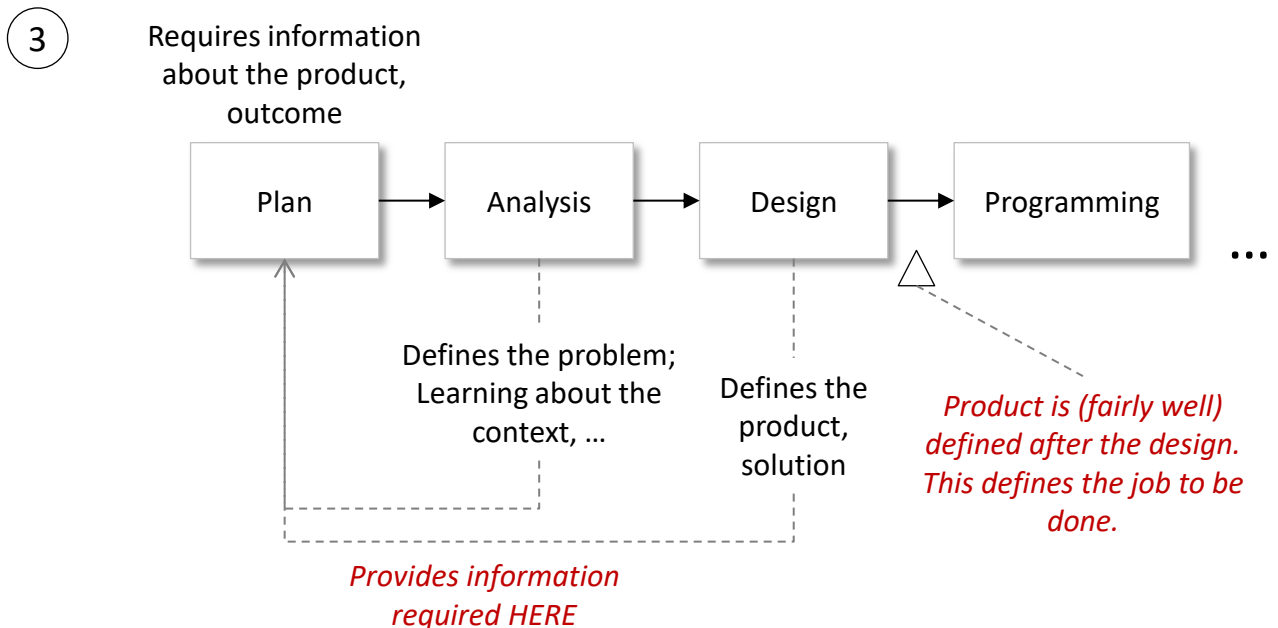
Notes:

Re-emphasising: “**Validation**” is different from “**Freezing**” and does not imply it. A plan can be validated. However, a plan is a **dynamic, adaptive, evolving** tool.



Why this doesn't work ?

- 1 **RIGID Planning** of hard to estimate activities in dynamic environment
- 2 Intellectual activities and collaboration are absolutely **non-linear activities**.



Analysis and design provide information required for planning. But these activities happen **after the planning phase**.

How to solve this?

A part of the solution is to do a HL pre-analysis & HL pre-design (before planning). Then a HL-plan can be established (coarse-grained) and a more low level-plan for the analysis activities. As the project progress, more HL-planned activities are planned in greater detail. (Rolling Wave) Uncertainty is decreased, but a lot of it still remains.

A plan is not static. It has to evolve, to be corrected, to be refined, to be adapted in order to converge to the goal. **Plan Dynamically**.

A project is a one-time (unique), uncertain and risky initiative producing a unique product usually using scarce resources. There is no established approach for them.

Need:

- To achieve the goal of the project
- To be effective and efficient
- To reduce risks
- To organise and control the project

How? Establishing a plan

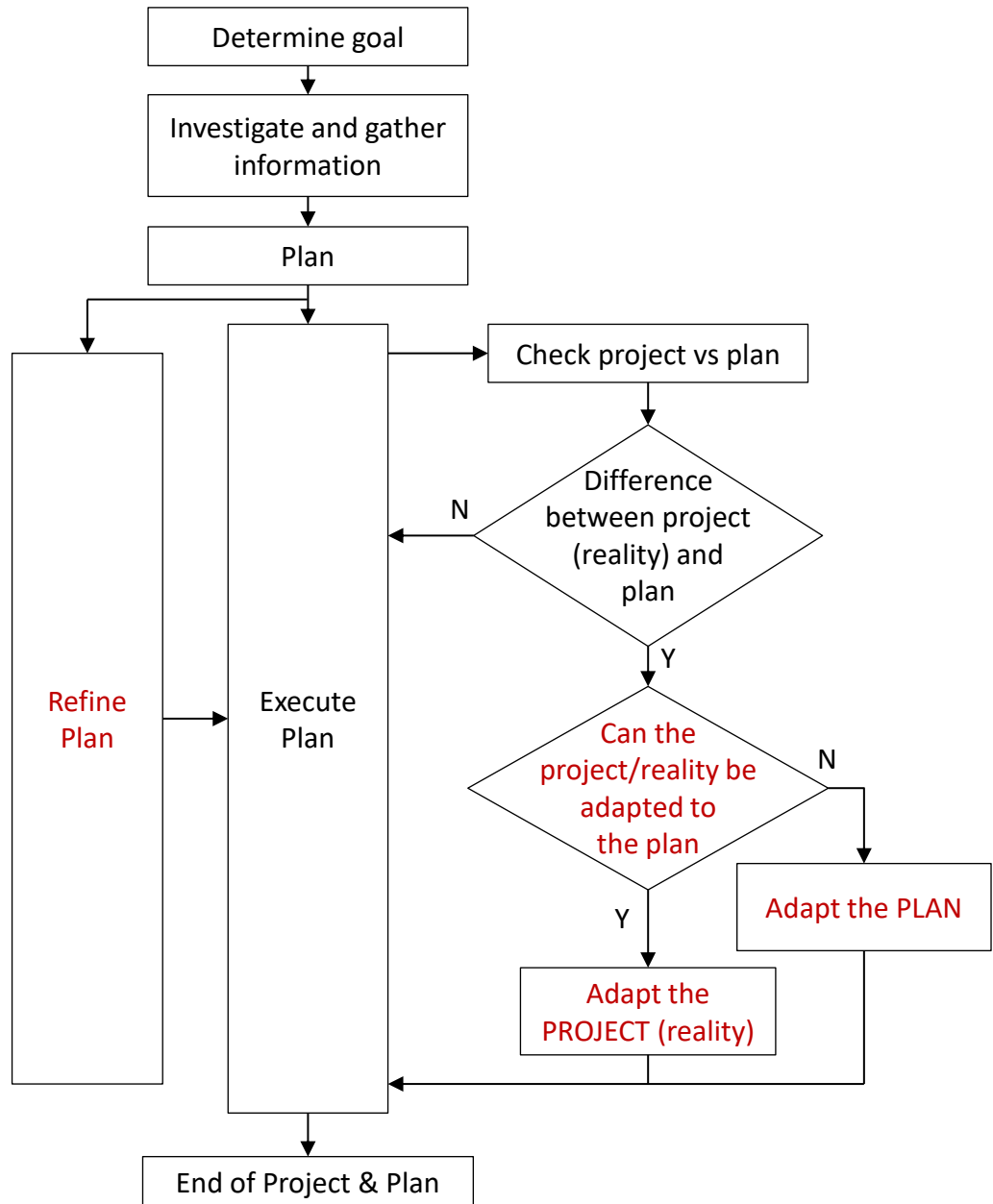
The **purpose of planning** is to execute the project effectively and efficiently while controlling and/or reducing the risks and to move swiftly and steadily towards the goal. It increases the chance for success to achieve the goal.

It's all about the activity of planning. Establishing the plan forces the planner to investigate, to think, to foresee, to structure and to organise. It must be used actively. A plan is an aid, not a self-constructed dictator.

Therefore we need to know things like:

- What process is required to execute the project
- What and how much resources are needed
- When are these resources needed
- We need estimates (we need estimate skills)

Warning: Planning aims to save time and resources. But planning takes time. The value of the plan and planning should be greater than the effort spent in planning. Spending too much effort in planning may kill the benefits.



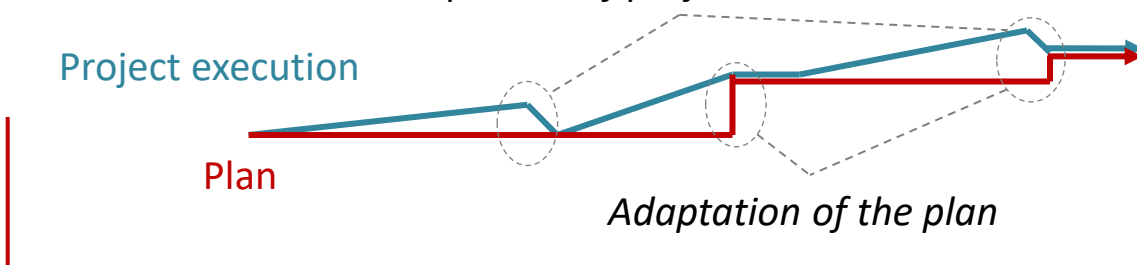
Plan versus execution

Adaptation of project execution

Project execution

Plan

Adaptation of the plan

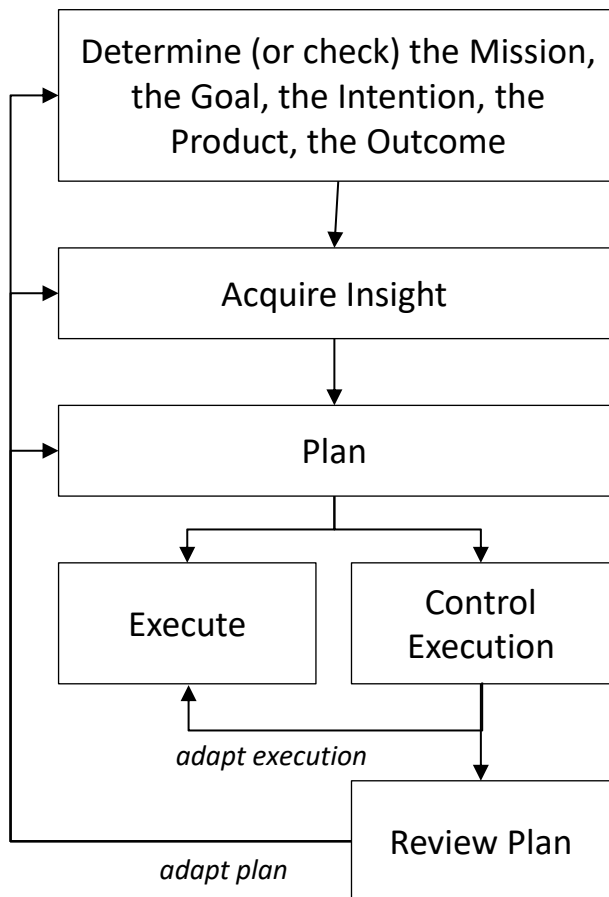


Mistake 1: Ignoring the purpose of planning

The goal of planning is to think about how an initiative can be organised, what it requires and other aspects necessary to obtain an efficient and effective execution; and thinking about critical aspect that may undermine the project. The aim of planning is to INCREASE the chances of success.

All action must contribute to this goal. Their execution must be based on the circumstances and in line with the goal.

Mistakes



- Wrong mission, goal, intention or product (often partial problems, local problems or consequences are solved)
- Acquiring insufficient or unreliable knowledge and insight. Worse: guessing or decreeing based on ignorance.
- Planning in too great details
- Planning for all eventualities
- All kinds of underestimations (time, resources, cost, knowledge, skills, norms, difficulty, ...)
- Critical aspects are missing in the plan (and haven't been considered)
- Plan is not used (in a drawer, not communicated, not followed)
- Plan and estimates are never questioned
- Plan is never reviewed and adapted.

Notes:

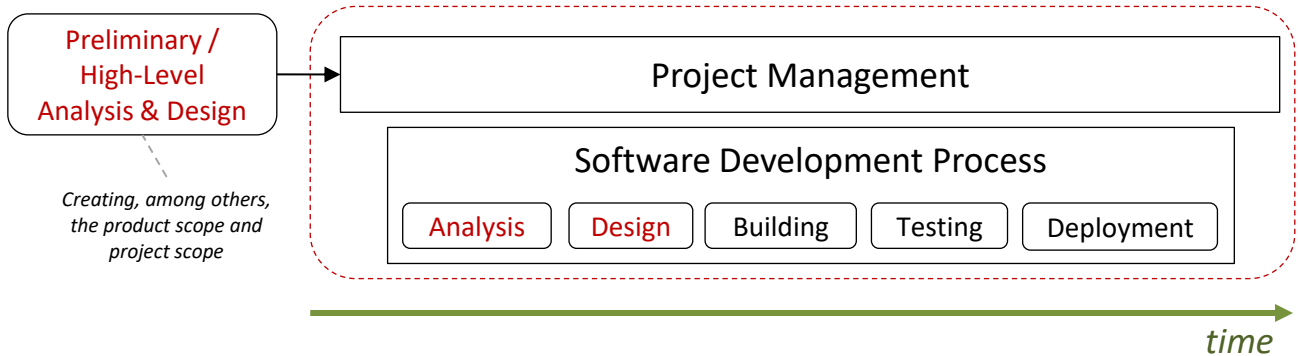
The described planning process provides a general idea. In practice, the process is NOT as strict as in the model.

Product Scope & Project Scope

10/01/2020

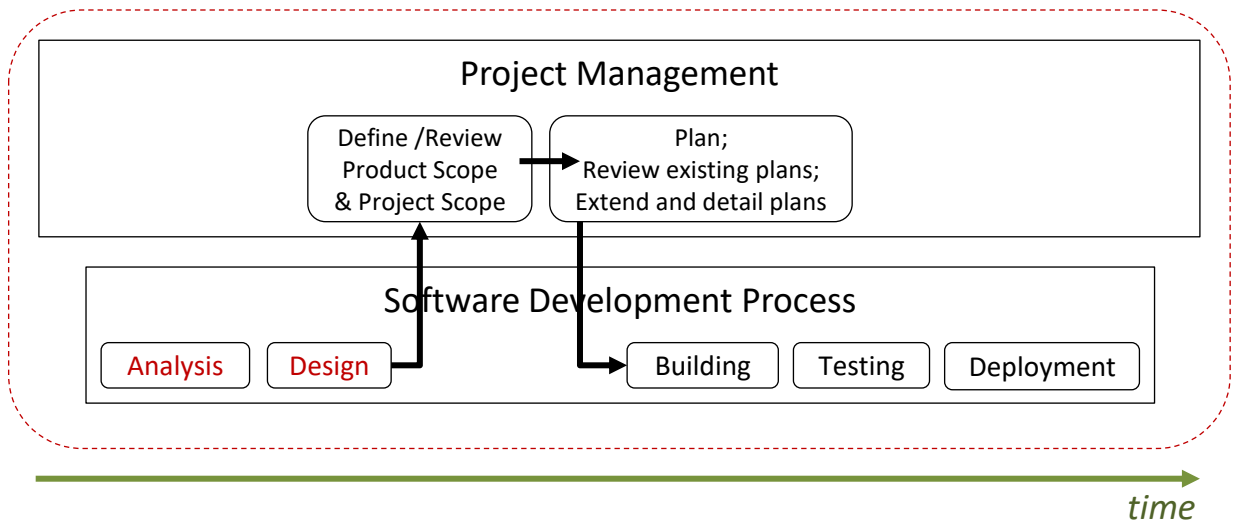
The Product Scope and Project Scope are the result of Systems Analysis and Design (in a broad sense; it includes Architecture).

A solution can be:



Product Scope and Project Scope are subject to revision during the project.

Another solution can be:



- General (undetailed) project plans can be made at the project start
- Or, only Analysis can be (loosely) planned. Later Design can be (loosely) planned. Only then the project can be planned a bit more in detail.

An Opportunity of a Change occurs during the project

- Is it obligatory ?
- Is it feasible ?
- Is it beneficial (Cost / Benefit / Risks / New Opportunities / New Limitations) ?
- What is the impact on the project ?

Possible Decisions

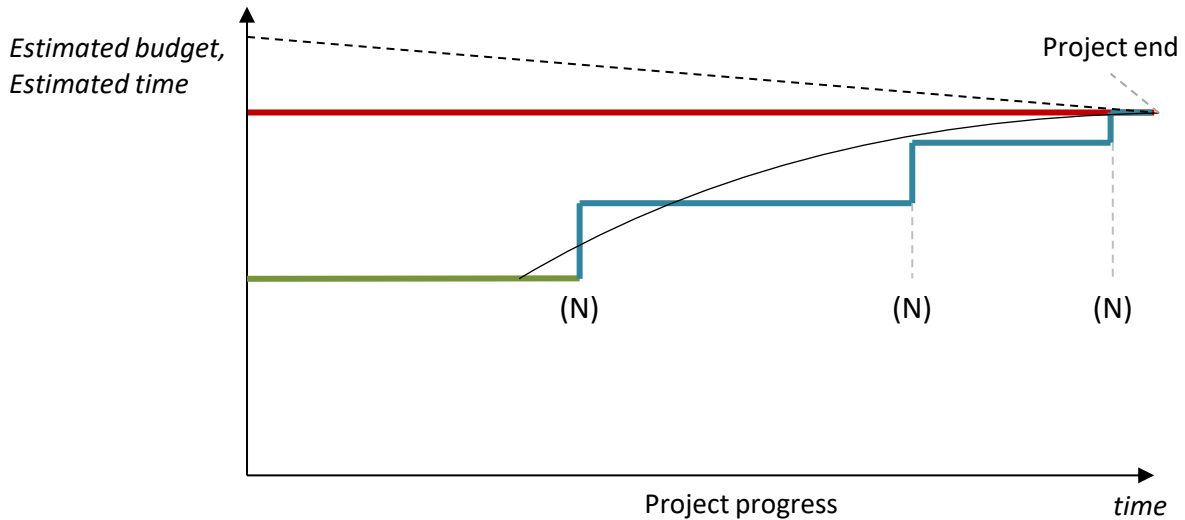
- Accept the Proposed Change
 - Implement as it is defined
 - Postpose the change
 - Adapt the change
 - a simplified or adapted version
 - Phased Implementation
 - Implementing the change partially now, other parts later.
 - Prepare the present solution to allow an easier implementation later
 - Let another party implement the change
 - A mix of the above
- Refuse the Change

Distribution of Effort

10/01/2020

Phase / Types of Activities	Effort
Pre-Analysis (Requirements)	5-10%
Analysis	15%
Design	10%
Construction + Unit Testing	25-30%
Integration + Integration Tests	10%
Systems Tests	15-20%
Acceptance Tests	5%
Deployment	5%
Documentation	5%
Training	

- Limiting the early phases will reduce the insight. A part of the learning shifts then downstream to later phases causing late investigation additional rework due to changes (corrections due to bad decisions).
- Project estimations are often too optimistic or too many impediments will appeared. Therefore the global estimate is often multiplied by a factor of 1.5 to 2.5.
- Seasoned project managers know not everything is foreseeable. They use time and money buffers to deal with it. These buffers can be used to regulate pressure. They, and no one else, manage them.
- Bad practice: Reducing Testing to meet the delivery milestone.
- Bad practice: Having a “Requirements” phase. Requirements is a technique or an outcome of Analysis activities.



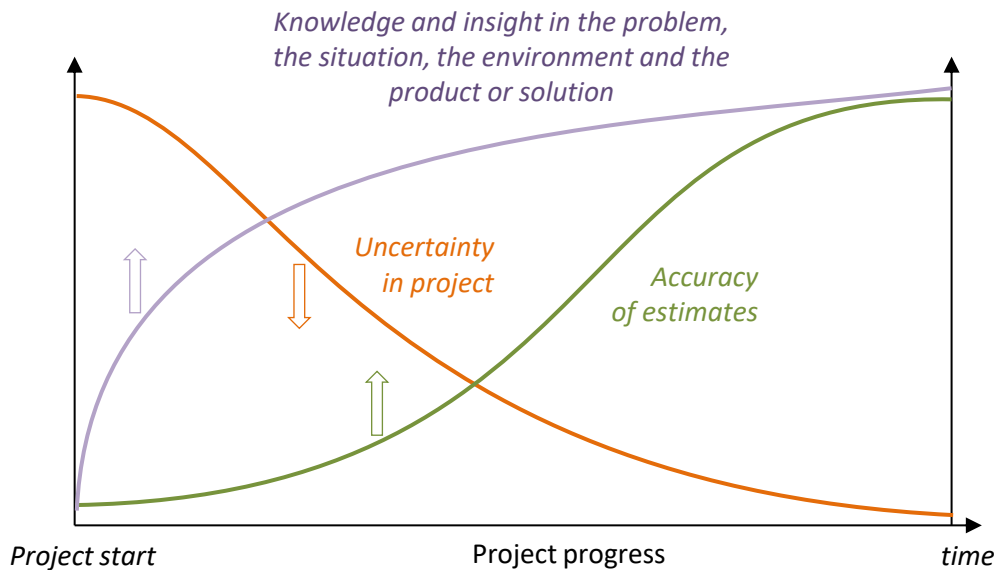
- Final duration and real expenses of the project (represented linearly for the sake of simplicity)
- Too short estimation of time or budget. Often done by business stakeholders instead of by the IT department and/or project team. It can be negotiated, which is also bad. Creates pressure, insufficient resources, ... Too short estimates are a reason of project failures, particularly when they are frozen. A cause of “Death March”.
- Additional budget and/or time are allocated (by negotiation?) until it meets the final duration and cost. Not only the short estimates have negative effects, allocating budget and time requires also a lot of additional time and effort.
- Time and budget estimates tend towards the final real time and budget.
- Rarely time and budget are overestimated.

Reasons of project failure (related to estimates):

- **Underestimation** (Causes: lack of experience, wanting to appear as competent, over-optimism, fear of not getting a GO for the project, stakeholder’s pressure, result of compromise, ...)
- Estimations done at the beginning of the project when greatest uncertainties and most unknowns.
- Both in combination with frozen estimates.

Estimate

- **Reliability** : depends of reliability, completeness, ... of information
- **Accuracy** : - 5 days to + 10 days
- **Variability** : conditions are very stable, stable, ... very variable; high risks of unexpected adverse events, variable pace of work, ...



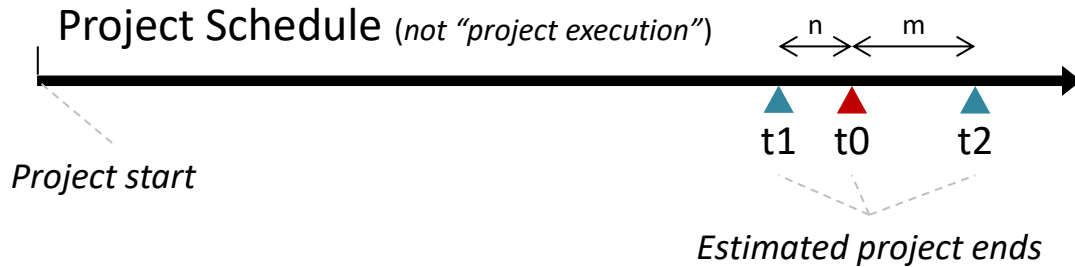
The lack of knowledge and insight creates a greater uncertainty. Both prevents from having accurate and reliable estimates in the beginning of the project.

Questions:

1. How can the uncertainty be decreased as quickly as possible and as early in the project as possible?
2. How can knowledge and insight be acquired as early as possible?
3. How can the estimates be improved? (trade-off)

Notes:

- Estimations competencies (methods, skills, experience) are necessary.
- When estimating, we can make use of assumptions. They have to be expressed together with the estimates.
- In the end, we can't always have what we want.



t_0 :

Ideal project duration estimated by Laplace's Demon

t_2 : overestimation (*team has more time*)

- More verification
- More aspects can be taken into account
- More learning
- Idea maturation
- More tests
- Experimentation, exploration, optimisation
- Time for alternatives
- Deployed later
- Risk for wasting resources

Even if $n < m$, the drawbacks and risks of t_1 are more important than for t_2 .

t_1 : underestimation

- Faster delivery
- Insufficient resources
- Higher pressure. Working faster
- More assumptions are made
- Stress, conflicts, burn out
- Lesser investigation
- Lesser verification
- More superficial thinking
- Fewer tests (risk of releasing solutions with more bugs)
- Higher risk for inappropriate, awkward solution, or even a non-solution creating more damage.

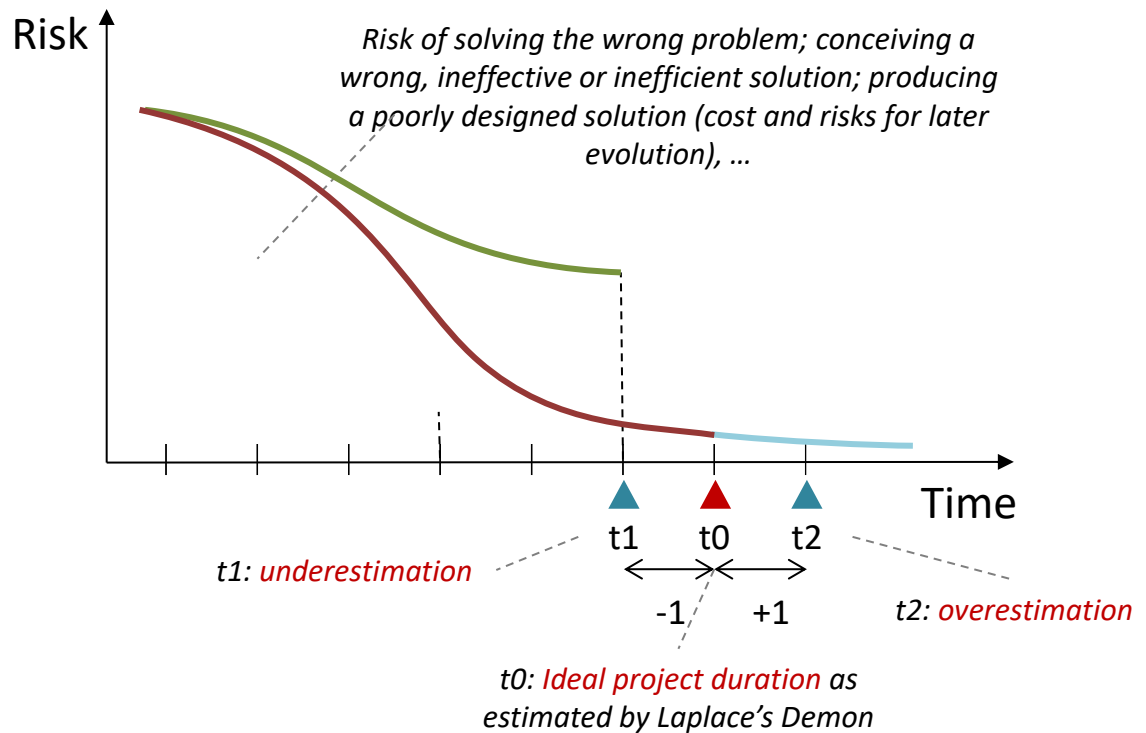
Better to overestimate, than to underestimate.

Notes:

Laplace's Demon is a superintelligence that knows and understands everything of the whole universe at all times. It can deduce everything that is deductible from this knowledge and understanding, except the truly unpredictable, like for example emergence, creativity(?),

Planning: Estimates

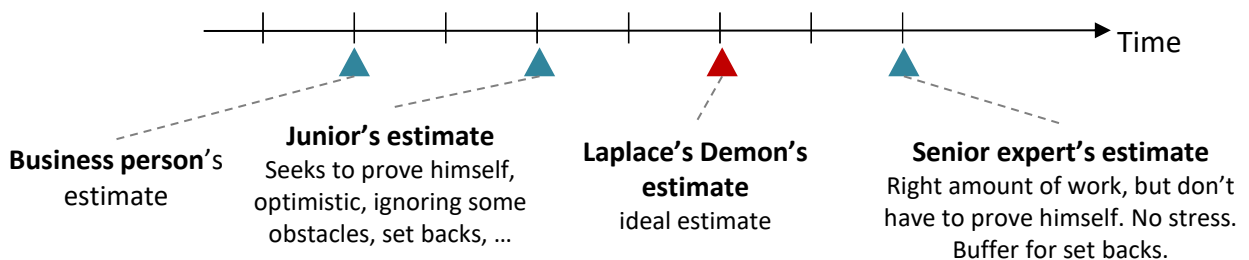
30/10/2018



- A slight underestimation may quickly pose a high pressure.
If a project's duration and budget are underestimated, the risk is likely to remain high. Right estimations may become underestimations when hindrances occur.
- Risks will quickly decrease with right estimations. Even on delivery some risks remain. It is impossible to exclude all risks.
- With overestimations, the risks may continue to be lowered (unsure). The question is whether the effort and time are worth this reduction. Beware that tasks are like a gas. It takes tends to take all the time and resources allocated to it. Overestimation doesn't necessarily lead to a timely delivery.

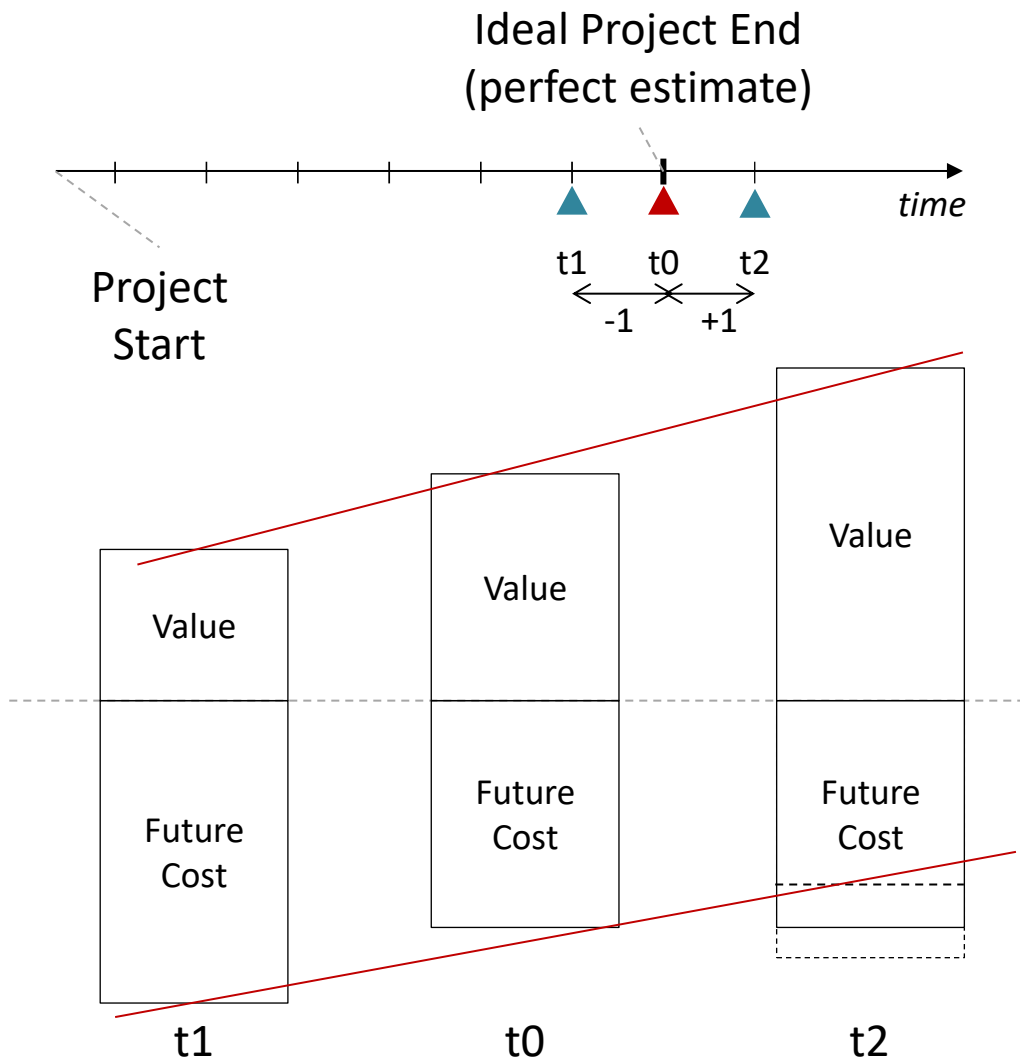
The **assumption** is made that the team works well and don't waste time.

Free time can be used for experimentation (innovation), improvement, ..



Investment, Cost, Benefit of Product

30/10/2018



Probably ...

t0: expected cost and value is created

t1: likely to have much higher cost, and much lesser value or increased risk.
Risks of getting lesser net value if seeking to save too much on project.

t2: Cost can be lower or somewhat higher, but value is more likely to be significantly higher (or risks much reduced)

Estimating too short is more risky than estimating too generously.

Project: Estimating

30/10/2018

UNDERESTIMATION

OVERESTIMATION

	Too little ←	RISKS		RISKS	Too much →
Duration		Unchecked info, superficial insight in problem and impact, weak foundation, inelegant design, lesser features, short cuts, more errors, work not done, technical debt, ...		Realistic / Acceptable ← →	Waste of time (likely)
Cost		Insufficient experience and competencies, low cost tools, furniture, ...			Wasted money
Difficulty		Oversimplification, real complexity is postponed			Overly complex (unlikely)
Quality		Lack of quality, inappropriate design			Luxury (likely)

May render the product completely useless and/or lead to serious post-project problems

Waste, but unlikely to menace the project

Devastating effects are on this side.

“Faster is Slower”

“Hasty speed is seldom good”

“Qui va piano va sano ”

“Cheaper is more expensive”

- Estimations are **ALL WRONG by definition**, but they should be useful.
- **Unrealistic estimations are worthless**. They do more harm than good.
- Estimation must help to manage the project. If they don't, something is fundamentally wrong.
The same is true for all kinds of plans, methodologies, methods, tools, ... Often the usage is wrong.
- The point is to **reduce the error margin** to get an estimation that is as optimal as possible.
- The fewer and **uncertain** information, the greater the uncertainties (beginning of the project), the more **unreliable the estimations** are.
- The **more knowledge and insight and the more things are settled** we have about the objective, the product, the situation, and so on, the **more precise** the estimation can be.
- Improve the estimation techniques, improve estimation skills and decrease uncertainty and risks.
- Don't stick with first estimation. Get the estimations reviewed and confirmed. Or, **re-estimate regularly**.
- **Estimations are, by their nature, never negotiable**. Maybe one can negotiate the scope, resources, features and other characteristics. Estimations are the deductions of these.
- Some project managers multiply the estimate by a factor 1.5 or 2. Some remove the highest and lowest estimates and calculate an average and then multiply it with a factor.
- Use buffers ! The PM should include buffers (and tell no one else). They manage and use these buffers when necessary. Is adapting plans not better?

Notes:

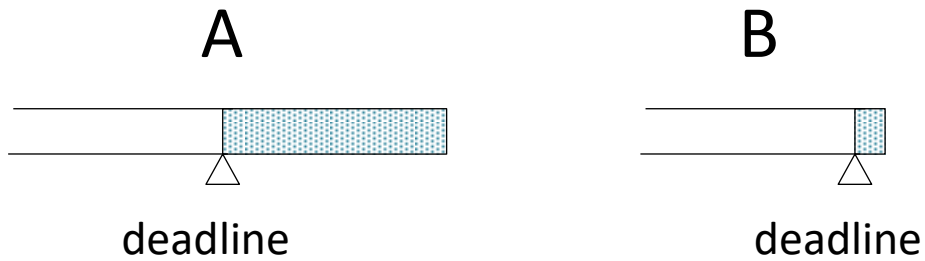
Wrong estimations is not about "being wrong".

Reviewing and re-estimating a project may lead to adapting the project plan. Yes, this is more work for the project manager. But this is part of the job.

Missed Deadline (1)

15/03/2019

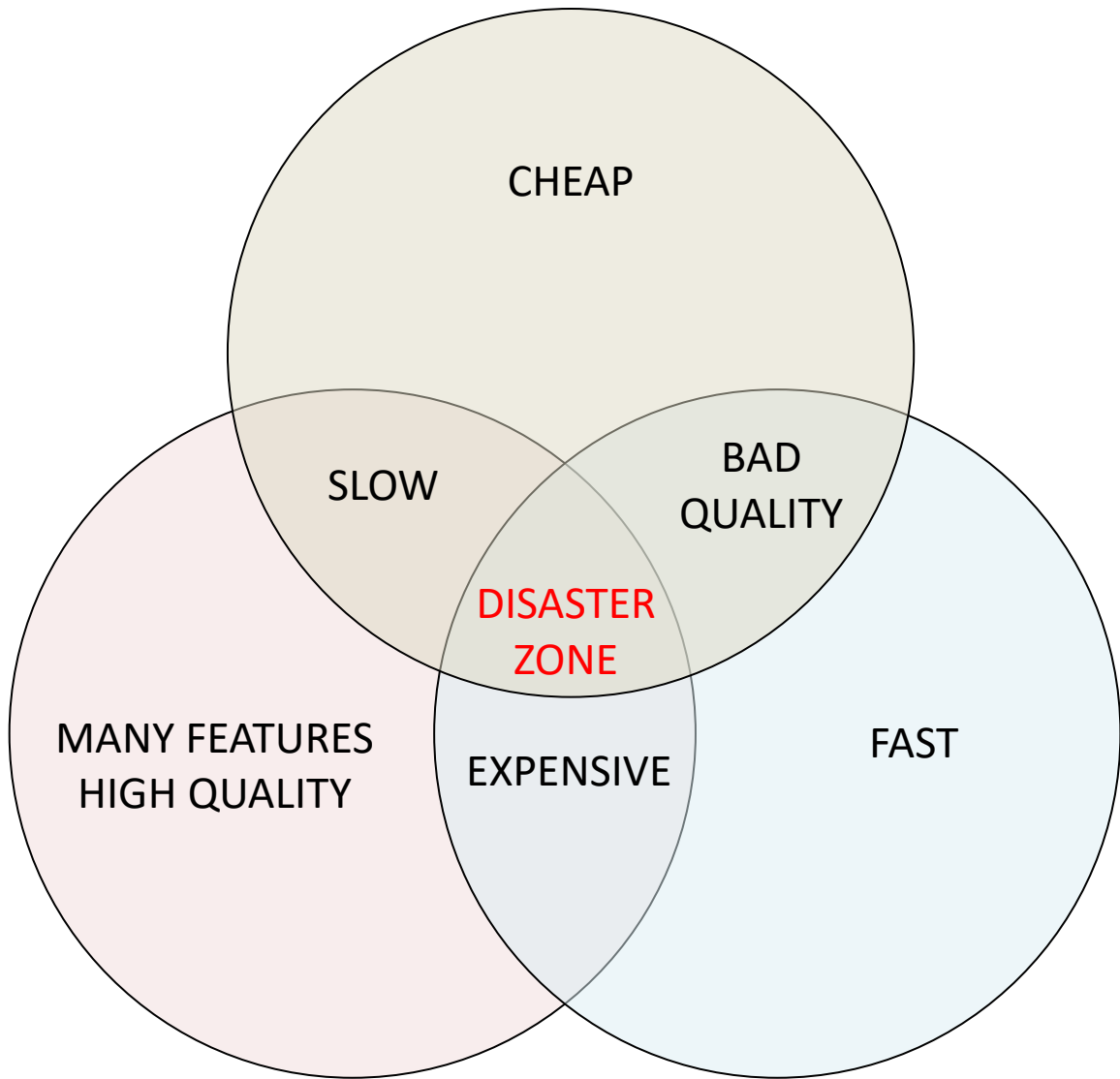
Missed Deadline



Which case is the worst? A or B ?

Project Estimations

30/10/2018



There are lower limits:

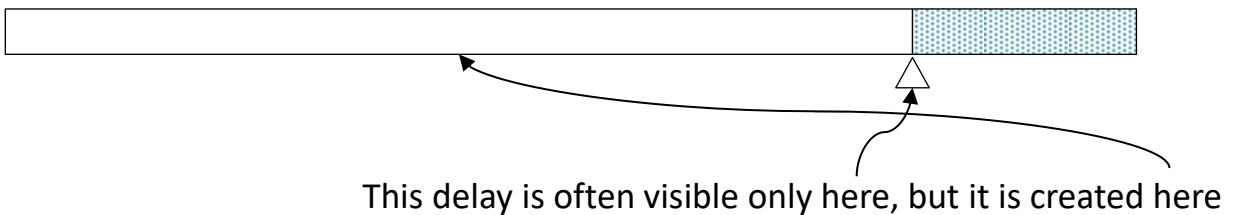
- In time: 9 women can not make a baby in one month
- In cost: if you pay peanuts, you get monkeys
- In quality: the price to pay will come later.

Fast	becomes	Slow
Cheap	becomes	Expensive
Many Features	becomes	a lack of quality

Missed Deadline

Missing a smaller deadline is not as bad as missing a big deadline. But this is misleading.

Case A

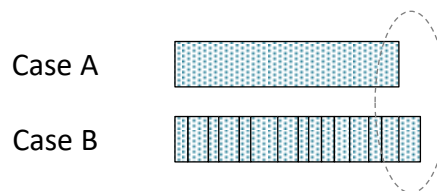


Case B



By cutting the process or time in slices (like short iterations), the deadlines can then be spread. The missed delays are smaller, but they are more frequent as well. It hides the true effect of the delays.

Small differences are not always documented (recorded, written, ...) They aren't (always) visible.



This principle exists in other situations: energy, effort, waste, costs, ...

Question: Is missing a deadline so bad? Should we ...

1. improve our estimating skills?
2. create circumstances allowing better estimates?
3. adapt the development approach to deliver within deadlines?
4. Or let the plan to vary (no deadlines, but rather expected delivery dates)?

Rigidity of Deadlines

10/01/2020

Why Do We Deal with Deadlines in a Very Rigid Way ?

Why Do We Think so Obsessively over Deadlines ?

- Estimated Deadlines (estimated! guessed! Imposed)
 - Not Delivering within a Deadline (on time) is a **FAILURE**
- Sprints have a fixed length
- Time boxing

Deadlines are used to create pressure.

Pressure is a bad substitution for motivation, for drive, for collaboration, for being interested, for attitude.

Pressure inhibits learning, curiosity, innovation and may destroy quality. It may create conflicts and burn-outs.

Does a company want/need a decent system or solution or does it want just something that can be built in a limited period of time and at a certain cost?

Do we value and check speed of development or do we value good systems?

Why does meeting deadlines matter more than conceiving and developing good systems ?

Missing a deadline may indeed require to adjust plans. But that's inherent part of dealing correctly with plans.

Negative Effects of Time Pressure

10/01/2020

Page to be reviewed

- Lesser time is dedicated to analysis. The superficial understanding increases the risk for a bad conception (design) and bad decisions.
- Much lesser is verified at any stage and at all levels. Verifications take time.
- Unconditional trust of demands, decisions, choices and information of others, even if not justified. ("We deliver or do whatever you ask for (even if the request unwise) and we perform the work the way you tell us to do").
- More assumptions will be made and blanks of knowledge will be filled creatively.
- Likely to chose for the first, simple, local short term or solutions. The chosen solution may be rudimentary and easy and fast to implement.
- Lesser or even no alternative solutions are considered. Sticking to the first simple solution.
- Lesser features are built. Features are kept unsophisticated.
- Lots of mechanisms and features necessary for the survival of the system (like controls, maintenance, ...) are not implemented.
- Cargo cult: Filling in templates, producing analysis artefacts and models; but not doing real analysis and real design
- Fewer tests are performed.
- Lesser time is spent on optimising and improvement
- Lesser time is dedicated to exploration and experimenting
- Lesser innovation
- The final solution is of lower quality and the opportunity to create a more valuable solution is destroyed.
- More likely to opt for temporary solutions. Temporary solutions may become definitive.
- Leaving out apparently lesser important deliverables
- Lesser communication
- Lesser sharing, exchanges and collaboration. Lesser mutual support
- Decreased efficiency
- More misunderstanding and conflicts
- More mistakes are made and goals aren't met.
- More corrections (rework) creating additional delays
- Increased blames and punishments.
- Tighter deadlines, obligations, blames and punishments instil fear.
- More self-protection at cost of others, of the project and of the solution. (letting others take decisions, information retention, hiding issues, taking shortcuts, ...)
- Exhaustions, burn-out
- The chosen "solution" may not even solve the problem or constitute hindrances or limitations later. It may cause problems elsewhere or may undermine the supra-system.
- Implementing (possibly unknowingly) time-bombs
- Risk for creating problems elsewhere in the system or organisation
- Meeting the demand gets a higher priority than meeting the expectations , satisfying the needs or reaching the objectives, let alone maximising the created value.
- Obtaining a solution undermining the supra-system and the

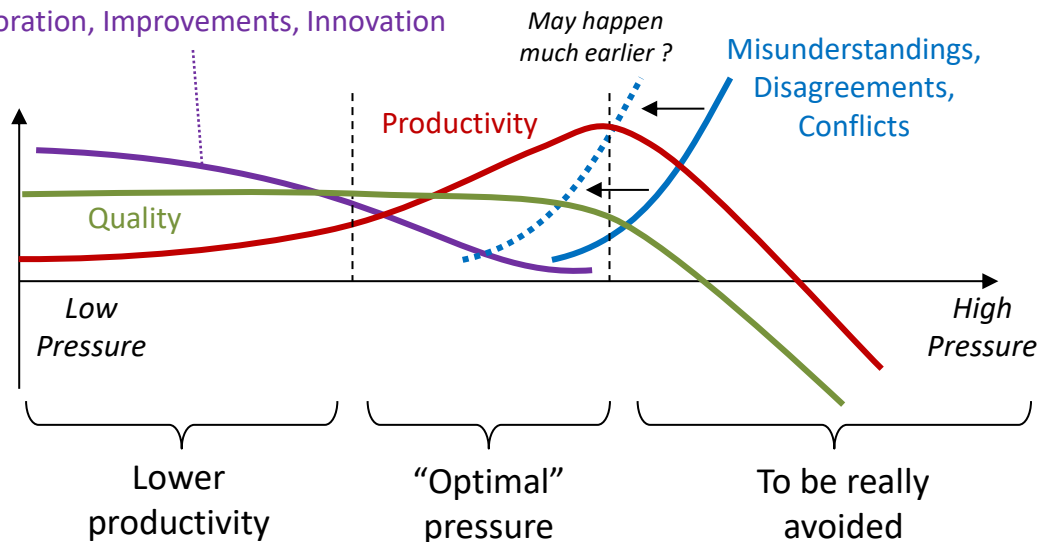
Pressure & Productivity

14/02/2020

Increase Pressure to Increase Productivity ?

Know what you gain.
Know what you lose.
And think again.

Innovation, Mutual help, Collaboration,
Exploration, Improvements, Innovation



Increased production comes at a cost.

- We can follow on the graph what happens when pressure gradually increases. Thinking: Why does it affect the other 'lines' the way is showed?
- Picture is illustrative to represent an insight. It shows what can possibly be expected.
- Not all people react the same way to pressure.
- A big quality drop may always occur. This risk is always present.
- **Pressure is not the only way** to increase productivity (work environment, work atmosphere, hiring the right employees, good leadership, tools, ...)
- **When** there is **low pressure**, some people will use the time they have to **learn**, to **explore**, to **investigate**, to **think**, to be more **creative**, to **teach** others, to **help** colleagues, to **share knowledge**, to **support** others, ... **while others** will waste time with activities not contributing to the organisation.
- People sometime do things, not measured, not asked, not visible, not known by management, maybe even not known by anybody but so much important for the company.

Why Do We Have a Very Rigid Way of Dealing with Deadlines ?

Why Do We Think so Obsessively over Deadlines ?

- Estimated Deadlines
 - Not Delivering within a Deadline (on time) is a **FAILURE**
- Sprints have a fixed length
- Time boxing

Deadlines are used to create pressure.

Pressure is a bad substitution for motivation, for drive, for being interested, for attitude. It creates stress and stress is bad for health.

Pressure inhibits learning, curiosity, innovation and may destroy quality.
It may create conflicts and burn-outs.

Does a company want/need a decent system or solution or does it want something that can be built in a period of time and at a certain cost?

Tips

- Define mission, product, release + time required
- Principle: “It’s ready when it is ready”
- Start Early - Remove all pressure – Remove all fear

DON'T USE STRICT DEADLINES AND MILESTONES

Progressing correctly and steadily is most important

Deadlines and milestones may serve as a guide, as an ideal situation.

Deadlines and milestones are (often) based on wishes or on estimates.

It is (often) better to miss a deadline and

- to do a sloppy job .. consequences and cost will come later
- to be able to implement a better solution creating much more value
- to be able to innovate.

Missing a deadline or milestone can be more beneficial for the company than meeting it.

Notes:

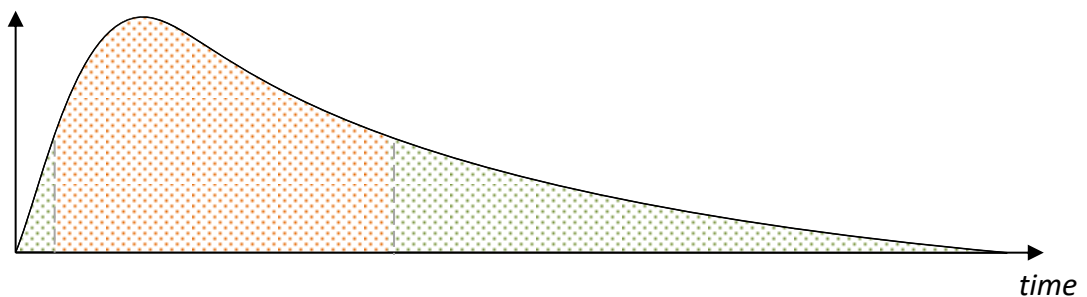
Although, there are circumstances when deadlines are critical.

Uncertainty, Difficulties and Risks

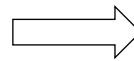
14/02/2020



If the project fails on the complex, challenging, uncertain and risky part, then the easy work performed earlier might be wasted. It's better to fail early in the project.



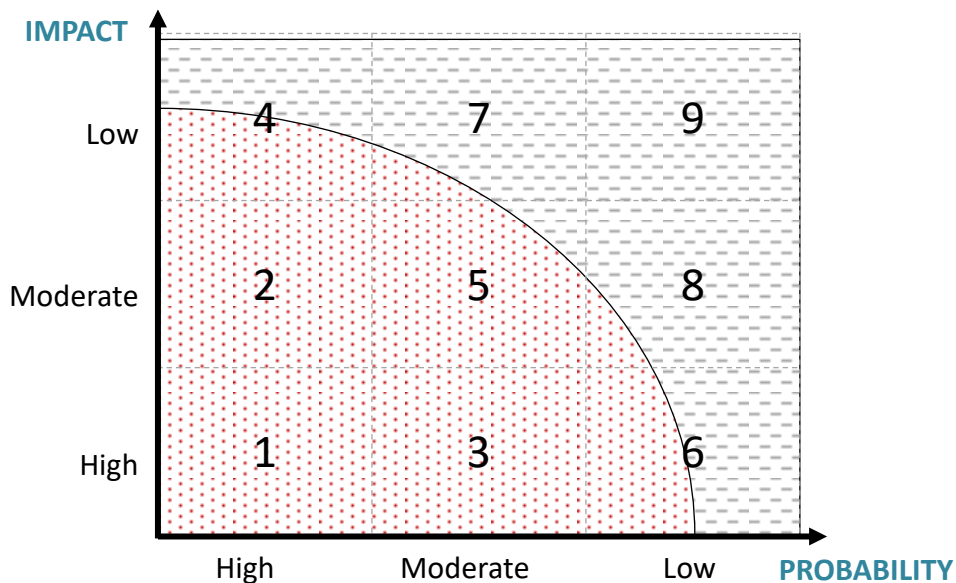
- Decrease risks early
- Decrease uncertainty early
- Deal with complexity early
- Tackle most difficult and risky parts first.



How ?
By Analysis, most risky elements first

Risk comes from not knowing what you're doing.

Warren Buffet



Risks are inherent to projects.

Major risks are often easily and quickly identified. This doesn't mean that all major risks can be identified immediately. A lot of smaller risks (smaller impact) may remain hidden for a while. Risks may surface at any time in the project.

It's a good habit to deal with major risks early in the project. Risks with minor impact can be dealt with later in the project.

The acceptance of risks depends of the company, of the stakeholders and of the product.

The severity of the impact, the probability and the cost to mitigate the risk are key factors in the evaluation. Other factors, like professionalism, responsibility, moral values and reputation, may also play a role.

Risk identification, which preferably has to happen as early as possible, require an understanding. This is provided by a decent upfront correct Analysis.

- **Tolerate** : Accept it & do nothing
- **Transfer** : Externalise it
- **Treating** :
 - Prepare for the risk
 - Take measure to reduce the consequences if the risk materialise
 - Reduce the likelihood of the risk to happen
- **Terminating** : Avoid / Eliminate the risk

30/04/2019





PROJECT



ONE OF THE KEY ACTIVITIES IN THE PROJECT


It increases the chance of future successes

How?

- Personal Improvement
- Development of company's capabilities
- Lessons have to be shared (learn from someone's else mistakes)
- Allows to avoid to make same mistakes in all the future projects (saves cost, resources, time and problems in the future)
- Allows to tackle more important challenges in the future

The project is the way companies use to develop and to transform themselves. If project and engineering skills lacks, the company can't adapt and evolve. If it can't adapt and evolve, it is doomed to die.

- What went wrong? What caused frustrations, dissatisfactions, ..? What could have been done better?
- Why did it go like that?
- How can it be improved? What had to be done? What was necessary? What would had been a better way?



INFORMATION SYSTEMS DEVELOPMENT FRAMEWORK

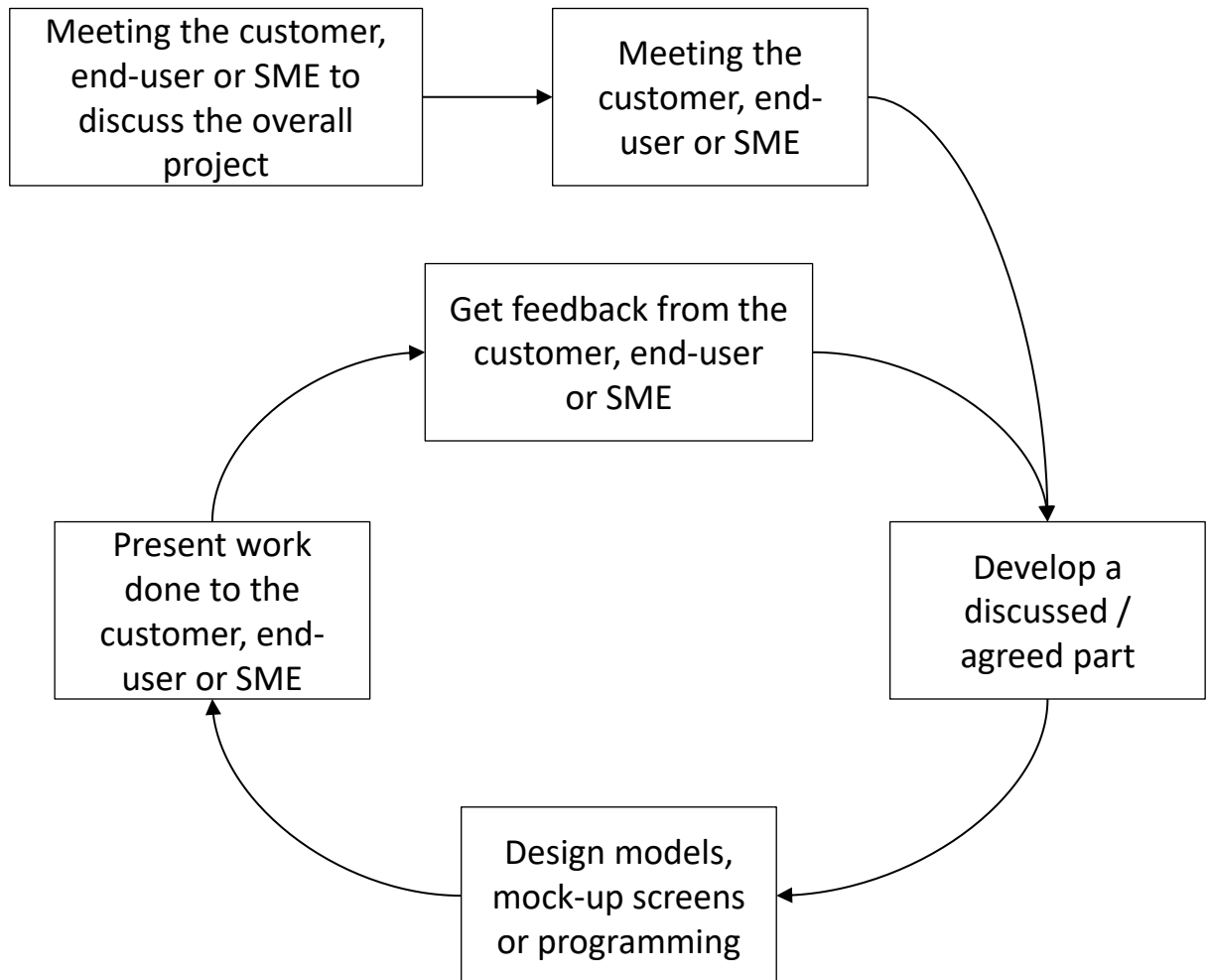
Approach, Framework, Methodology

30/10/2018

Philosophy	Set of values, ideas and principles
Framework	A basic global description of an organisation or structure. It may include patterns useable to conceive a global organisation for a specific initiative.
Approach	Vague description of a global process
Methodology	A precise description of a global process, consisting of activities and methods. It can be enriched, for example, with artefacts, principles, standards or rules and techniques.
Method	A particular process with a well-determined purpose to accomplish something specific and consisting of several ordered steps.

Goal

- To Facilitate the project, the work
- To help the team

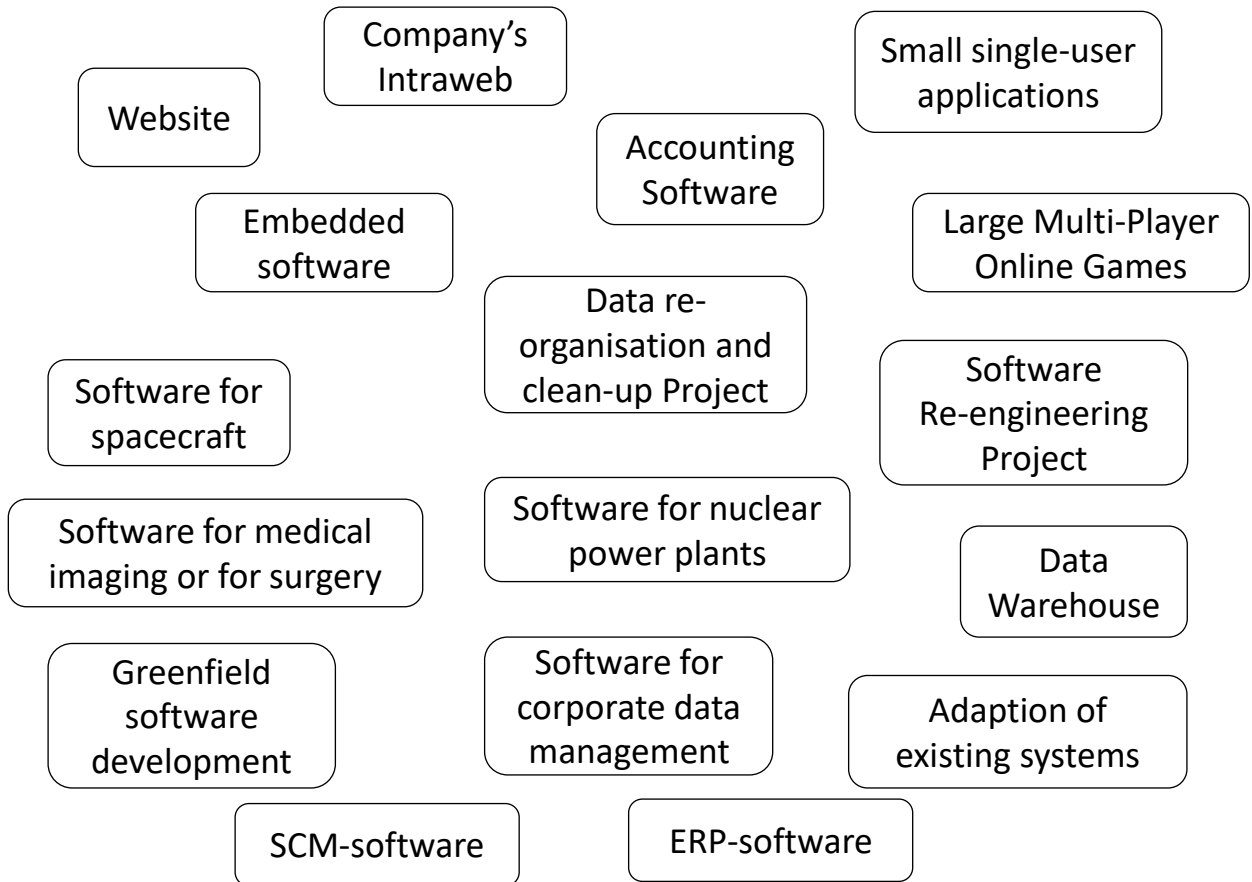


This works for small teams, simple problems, simple and local solutions.

For larger, tentacular, multi-disciplinary solutions crossing organisational borders which share components or resources and interacts with other systems a more holistic, supportive and organised approach is required.

Different Software Dev. Projects

30/10/2018



- Different philosophies, vision, values, priorities, constraints
- Different levels of integration
- Different spread of knowledge
- Different types of stakeholders
- Different ways of deployment, deployments constraints
- Different risks and risk acceptance (criticality, quality requirements)
- Minimal required software to be complete, functional and valuable

DIFFERENT PROJECTS → DIFFERENT METHODOLOGIES

Notes:

Some operators (users) can operate machine, but have no clue of the internals of the systems, thus of the required implemented logic.

A spacecraft can't be sent into space with only the most important features required for taking off installed. A spacecraft's software system must be complete before the spacecraft is launched.

In Software Engineering
doing things by the book
and
thinking dogmatically
is a recipe for FAILURE
regardless of the approach

Be PRACTICAL and ADAPT
so that your processes, methods
and tools fit your situation

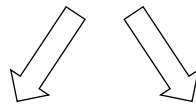
Software Development Initiatives

are

NOT initiatives based
on **STRICT FORMAL PROCESSES**

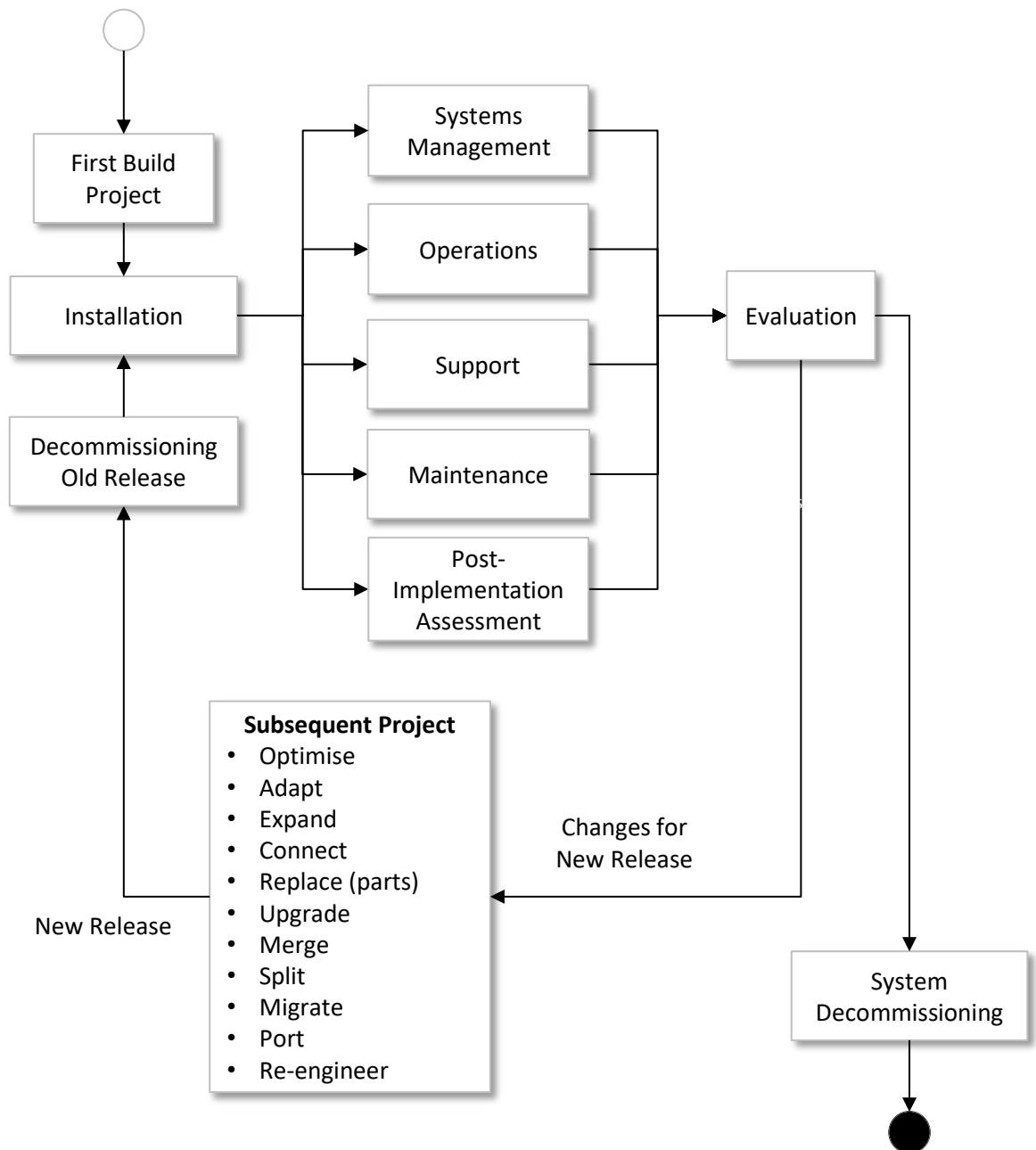
They are

**Semi-structured /
CASE-based Process
Initiatives**



Any Methodology or
Process defined upfront
will **have to be adapted**
during the execution

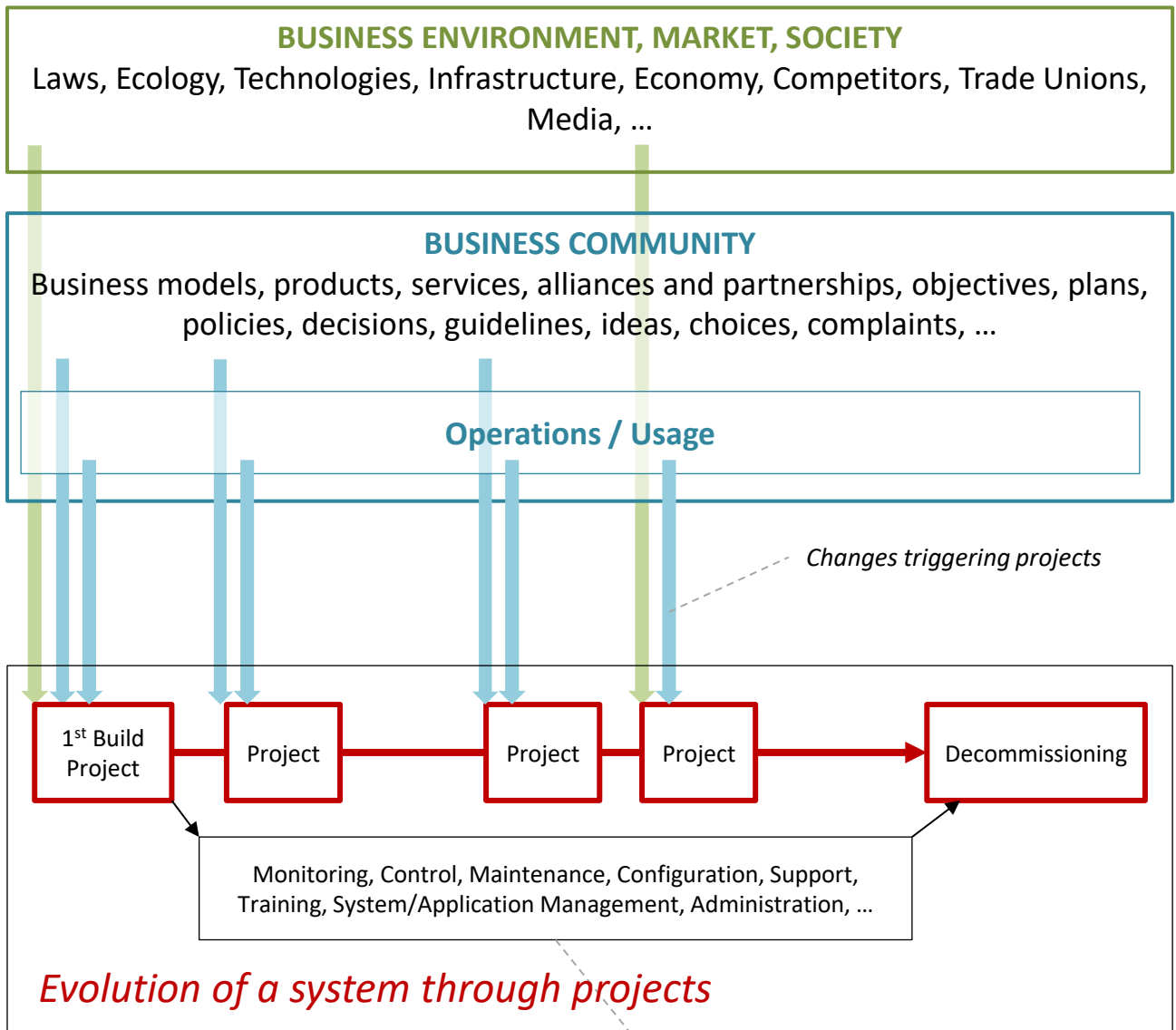
Mastery of the **Discipline**,
in-depth understanding
and common sense are
critical



Lifecycle \neq Development Cycle !!

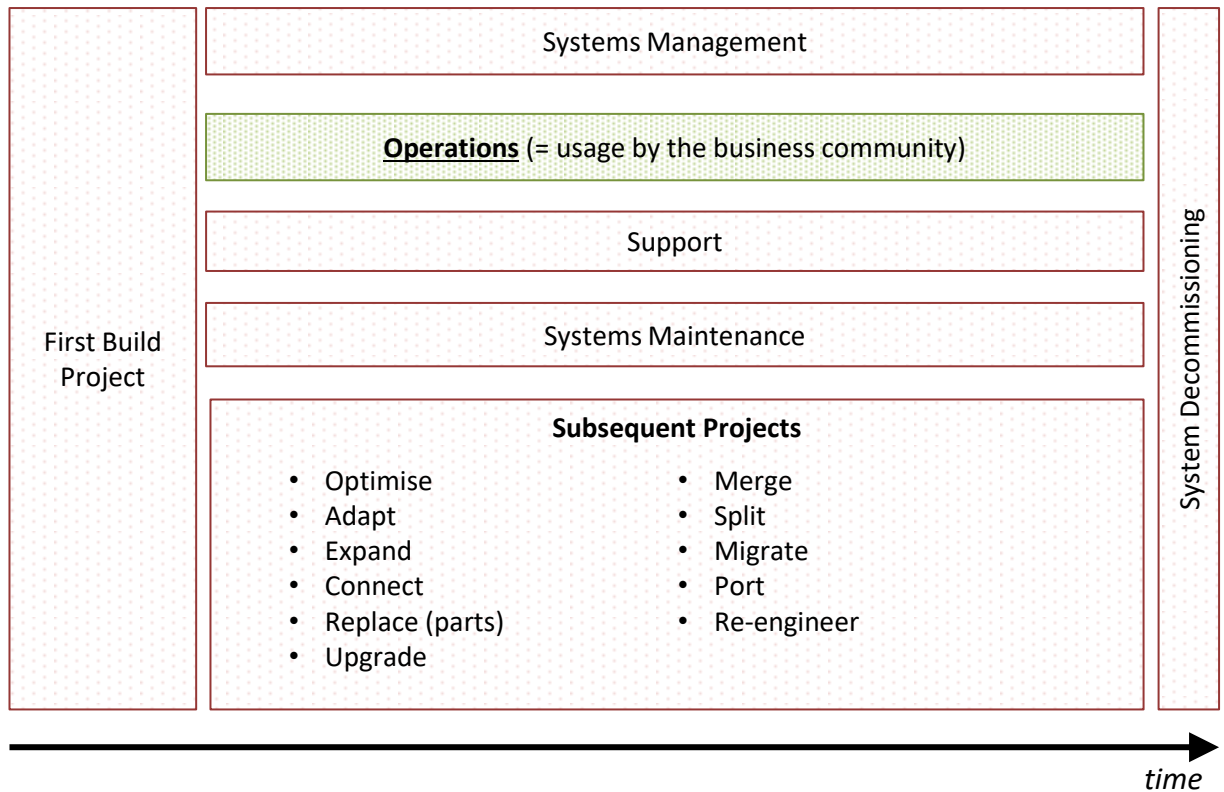
Software System's Life (2)


30/10/2018



Software System's Lifecycle (3)

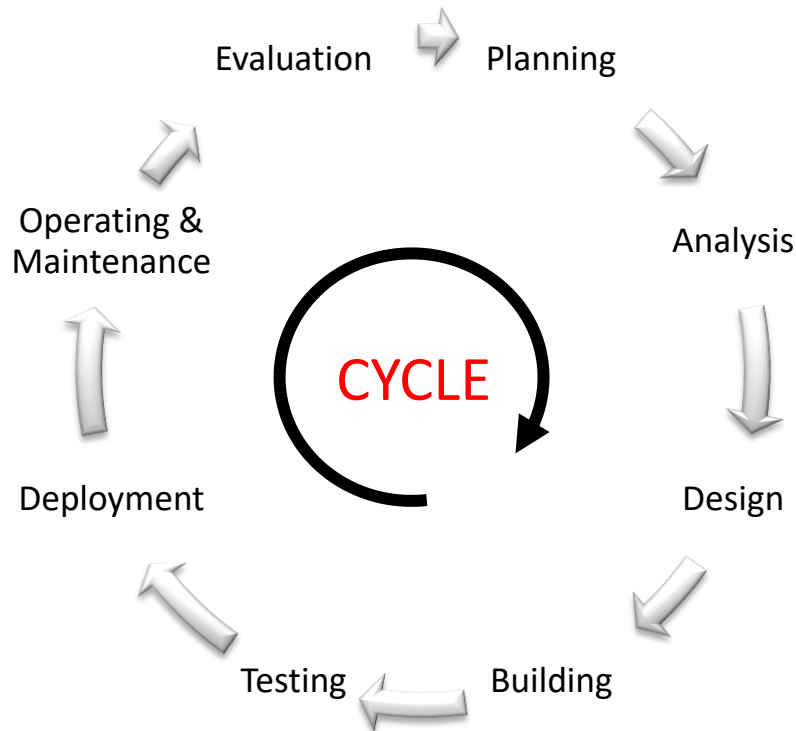
30/10/2018



 Responsibility of Business Community

 Responsibility of "IT"

System / Software Development Lifecycle



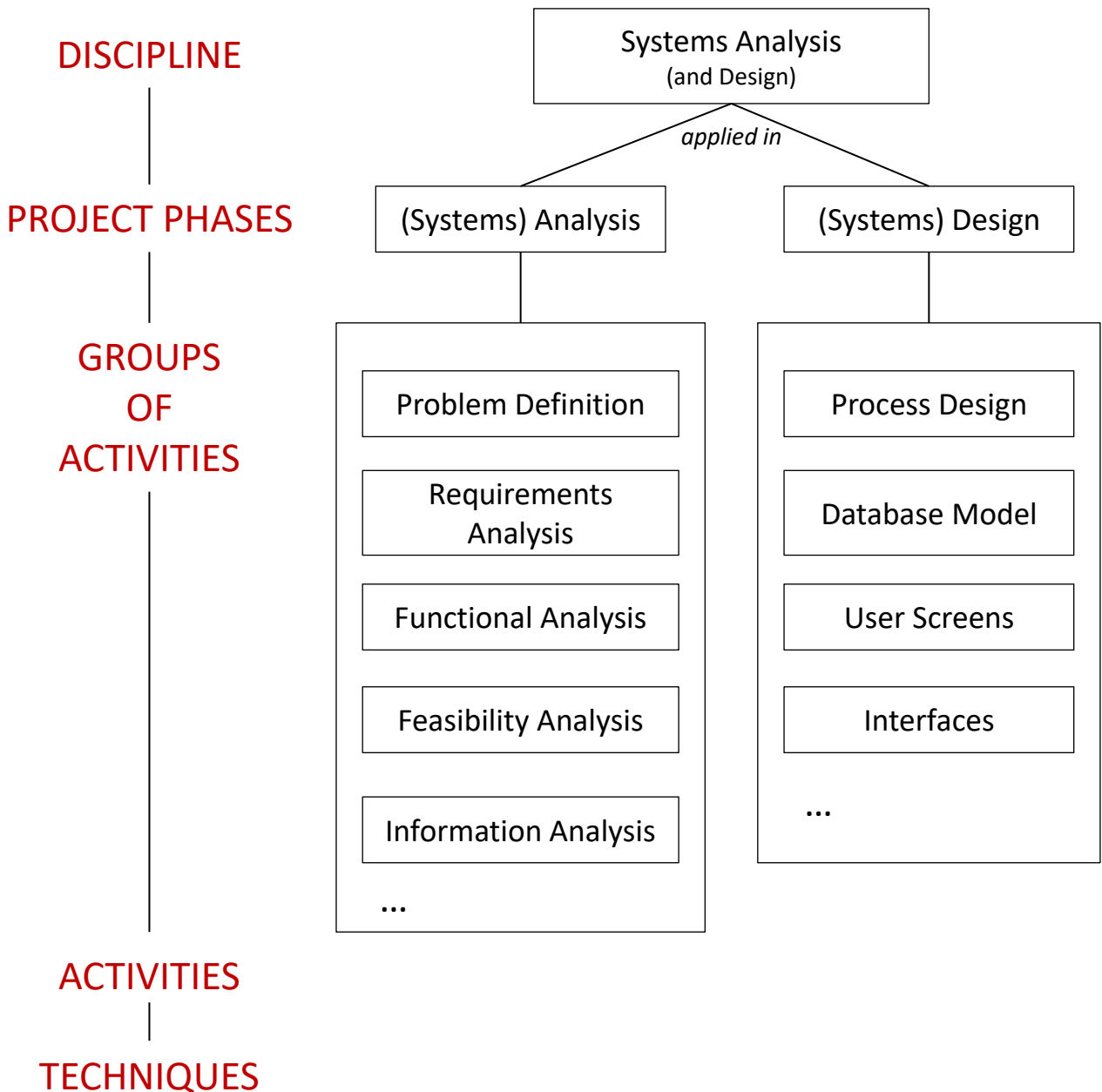
Note: Some SDLC have lesser phases or the names may differ. However, the concept, the idea, remains the same.

Development Cycle \neq Life Cycle !!

“Requirements” or “Requirements Analysis” are techniques, not phases !!

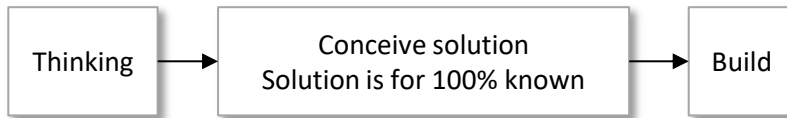
Discipline → Activities

30/10/2018

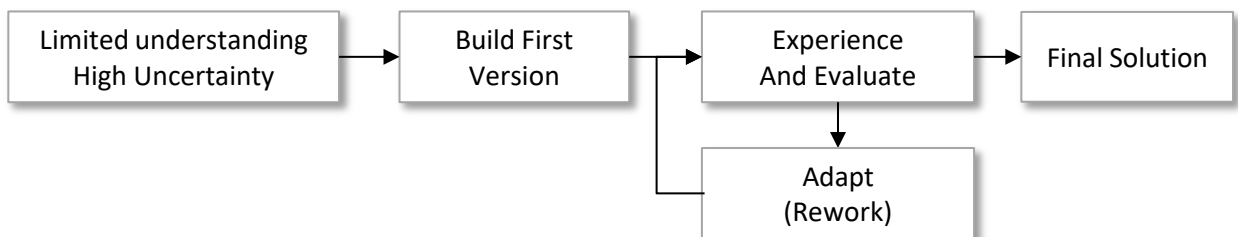
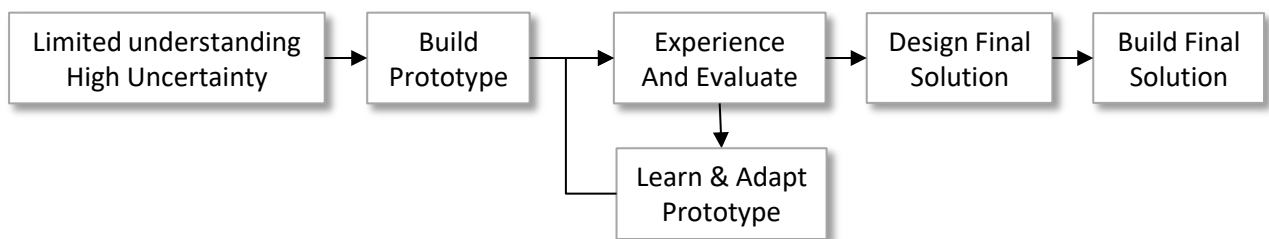
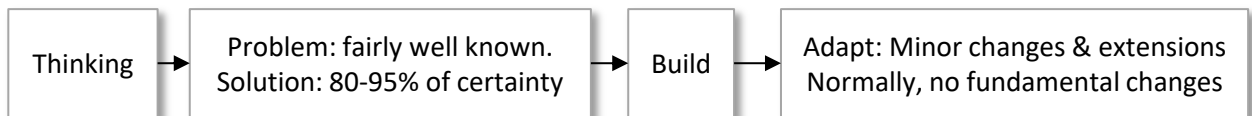


Notes:

- Systems Analysis is exercised throughout the whole project. However, it is essentially exercised during the Analysis and Design phases (whatever their names). The “Groups of activities” represent a part of the “Analysis” phase. They can be defined as a phase on their own, but the practitioner must make sure not to forget critical activities in the process.
- “Requirements” is rather a technique requiring several activities like (requirements elicitation, req. analysis, req. management, req. prioritisation, req. validation, ...)



➔ *Ideal case. Not possible in most cases.*



➔ *Slow, Costly, Time and budget depending on # cycles and amount of rework, late final delivery, ...*

METHODOLOGY is a global methodical (structured) process, which is commonly decomposed in phases, sub-processes, gates, activities and milestones, and which can propose or be enriched with values, principles, rules, roles, skills, products, methods, techniques, standards, artefacts, tools, technologies, ...

It's a an organised and ordered collection of elements **proposed** to projects (not just one) and which **can be useful** to them.

Advantages

1. Facilitates transfer of Project Experience
 - The methodology accumulates this experience, insight, ...
2. Facilitates the set-up and planning of projects
 - The project can start with a 'template' instead of from a blank slate. Not everything need to be redefined from scratch. It's "just a matter of" adapting the standard methodology to the specific case of the project and its needs.
 - It supports the knowledge and experience (lesser chance to forget taking something into account or to include something.)

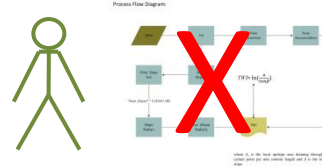
As a way to introduce some standardisation

2. Facilitates communication
 - Usage of common terms across projects
 - Common concepts facilitates reporting to management
3. Facilitates collaboration
 - Easier to step into a project.
4. Facilitates the management of programs and project portfolio
 - Obligatory steps and artefacts common to all company's projects can be defined.

Risks

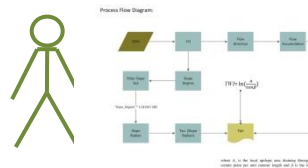
1. Overly trusting and relying on the methodology.
2. Assumption that a methodology replaces understanding of Software Development Processes and projects.
3. Misuses and Abuses of the methodology. Confusion between a methodology and a procedure.

No Framework, Methodology, Process, Procedure, Guideline, Method is available



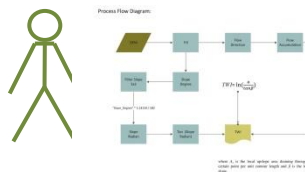
- Free to act
- Having to think and to take decisions
- In same situation, people will act in different ways.

A Framework, Methodology, Process, Procedure, Guideline, Method is available



- Individuals may, have to, or assume they have to follow it faithfully
- Assumption:
 - That the process description is perfect
 - That every case can be resolved by following it
- More superficial understanding, lesser thinking, lesser taking responsibility
→ Effects on result and on risks?

A Framework, Methodology, Process, Procedure, Guideline, Method is available

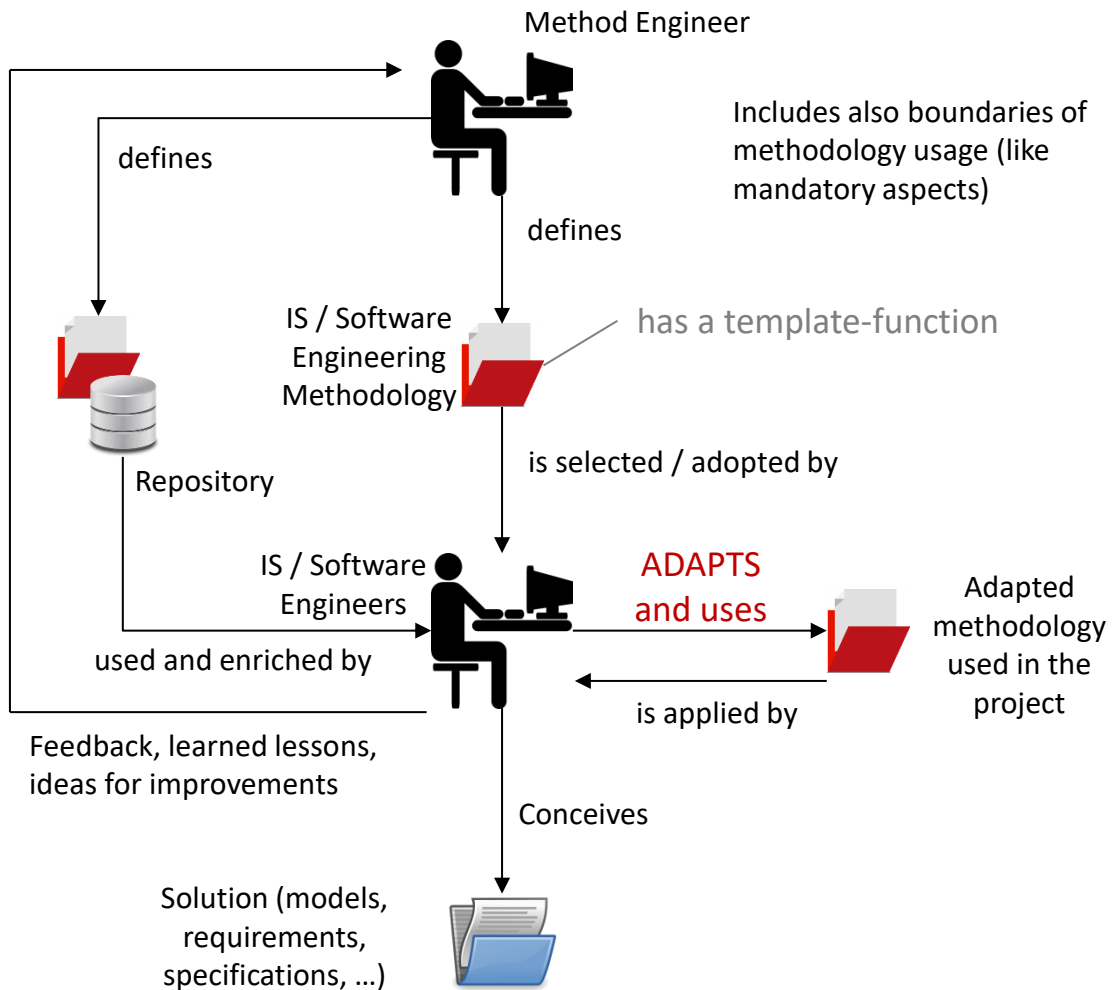


- Used as input for taking own decisions of the right course of actions
- Used to support the thinking. If the step makes sense, then it is performed. If the order of steps make sense, then it is respected. If the artefact is useful, then it is produced. Else we adapt, reorder, ..
- Require true expertise. Responsibility is being taken.

METHODOLOGY =
A GENERAL PROCESS
supporting specific cases

PROJECT = SPECIFIC CASE

- A methodology ≠ a procedure, a recipe.
 - A methodology does not prescribe. It is not a prison. It is not a master that dictates.
 - A methodology helps by describing and suggesting. It shouldn't and can't be followed blindly. Never do something "because it is prescribed in the methodology".
- A methodology must be adapted to suit the specific situation of the project
 - Is an activity, model, document or other product needed ?
 - Adding or removing products – selection of methods
 - Adding, reordering, removing activities
- Richer methodologies are usually better / offer more possibilities
 - The practitioners have more suggestions, more choice
 - It's easier to ignore (leave out) an unnecessary activity, artefact or technique than to search, create and add one.
- A company can decide to have some parts of the methodology to be mandatory to every or to certain classes of projects (milestones, artefacts, KPI's, ...).
- A methodology is based on knowledge. It contains (some of the) knowledge the practitioner should possess. However, it doesn't replace this knowledge !! Without knowledge, deeper insight, critical thinking and sound judgement ... follow the methodology is a recipe for disaster.
- No methodology, approach or philosophy can cope with the consequences of lack of insight, a restricting vision or a rigid mind-set. The vision must be clear and large. Mind must be open and flexible.
- The application of a methodology, on its own, is NEVER a guarantee for success. Let alone a strict and faithfully application.
- A methodology is never responsible for a project failure. People who chose it and executed it are !!



Method Engineers and the IS/Software Engineers (including all types of Analysts and Architects), require a profound understanding of projects and their products in general. The IS/Software Engineers need also to have a good understanding of the particular situation of the initiative and its context.

The problem with framework, methodologies, processes and other standards is not that they are predefined.

The following **FALSE assumptions** are made too easily:

- These standards have **to be respected**.
- The experts have thought about the framework, methodology, etc, so the **practitioner don't have to think about it anymore**.
- They have the role of an **authority** that dictates.
- They **tell** the practitioners, the experts, **what to do** or how to do their job.
- They **have to be applied as defined**.
- They **oblige and limit** practitioners.

Rigid
Mind-set

They are used as a **replacement**

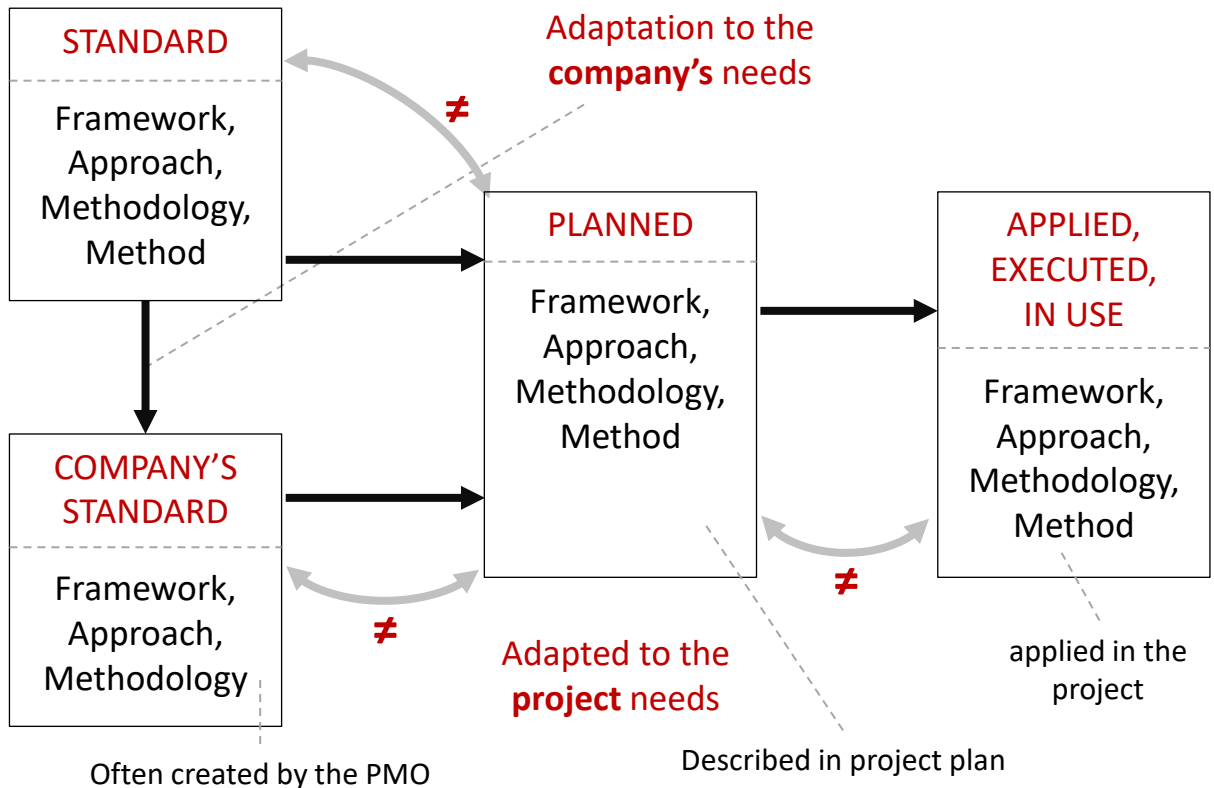
- to **insight in the profession** (mastery)
- and to **autonomous thinking**.

while they aren't and can't be.

Actually, they are only a basic template, a toolbox, with which the practitioner can start to work with. They only suggest steps, activities, methods, etc. They support practitioners in their thinking and decision making about the approach.

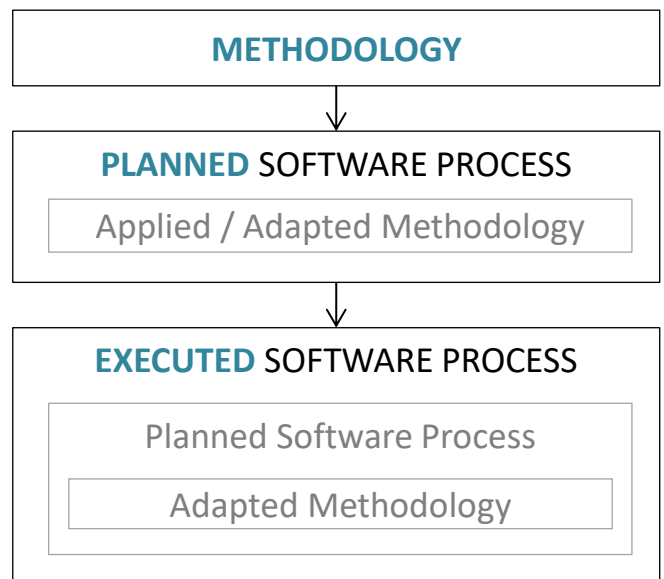
Note:

Some organisations may indicate some parts of methodologies, of frameworks, etc . as mandatory.

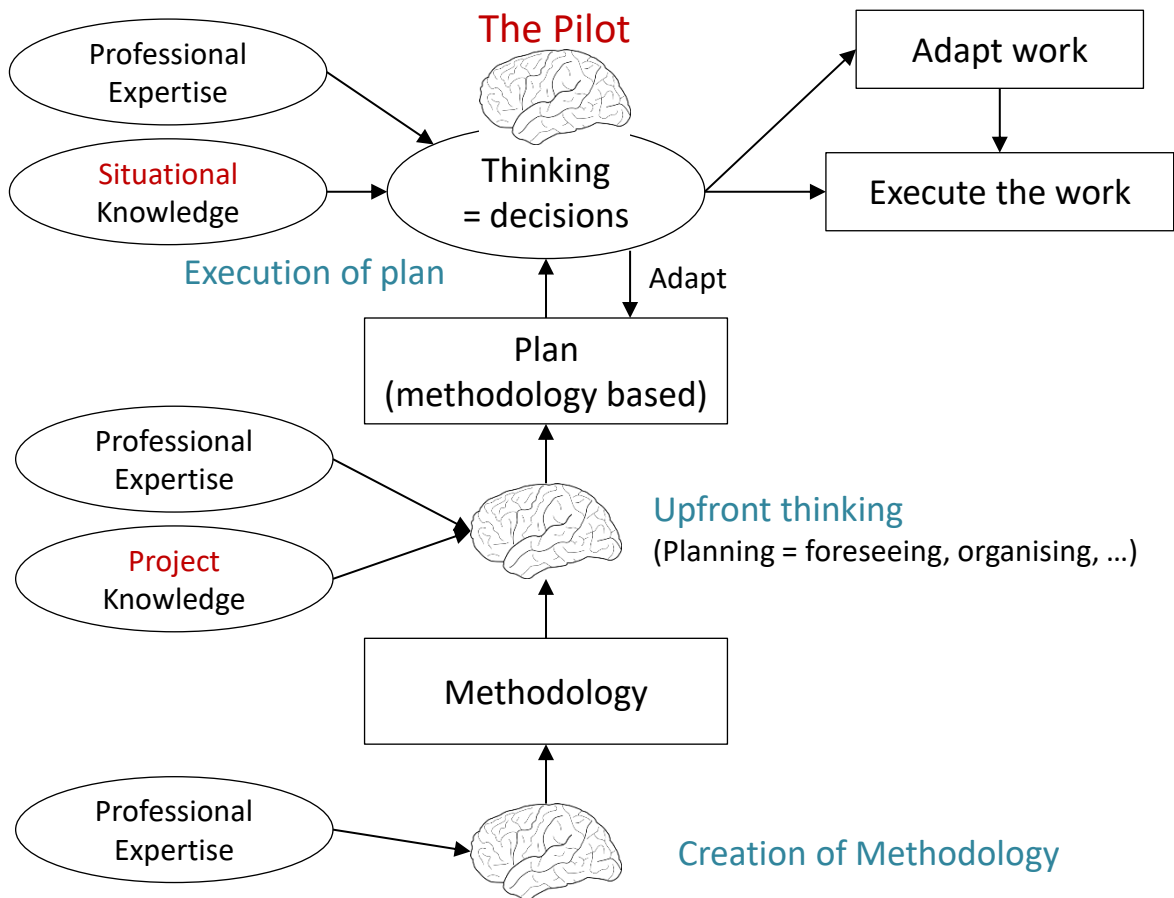


The execution of a plan and the plan are different things. And both, from a process perspective, always differ to some degree.

The necessity during the project execution determines how a methodology is executed. Practitioners have to understand these necessities and adapt the application of the methodology to it.



Despite the divergence between these 3 concepts, a great similarity should be distinguishable. A lot of elements and patterns should still be present and recognisable.

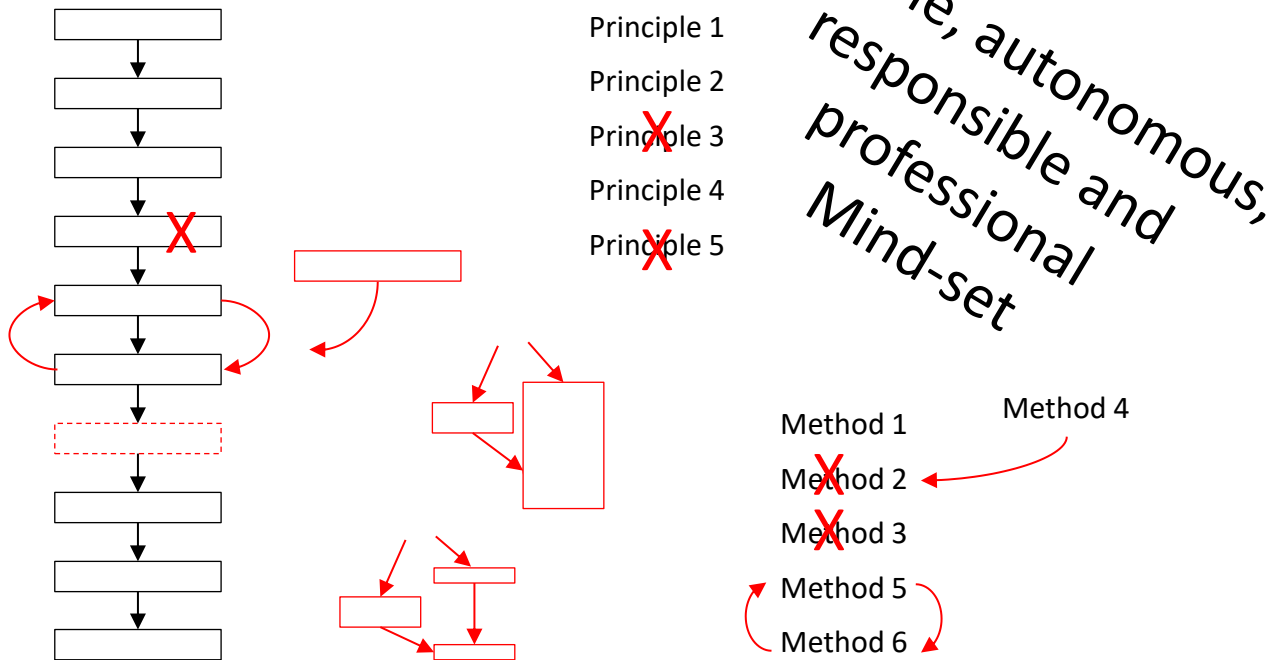


It is ALWAYS the responsibility of the project manager and the project team to decide how the project is executed and what is needed, what is necessary to succeed with the project.

A methodology is a basic template suggesting activities, artefacts, techniques, an order of activities, roles, ...

The main role of a methodology is to support the thinking of the project manager and the team, it does not replace this thinking.

Adapting a Methodology



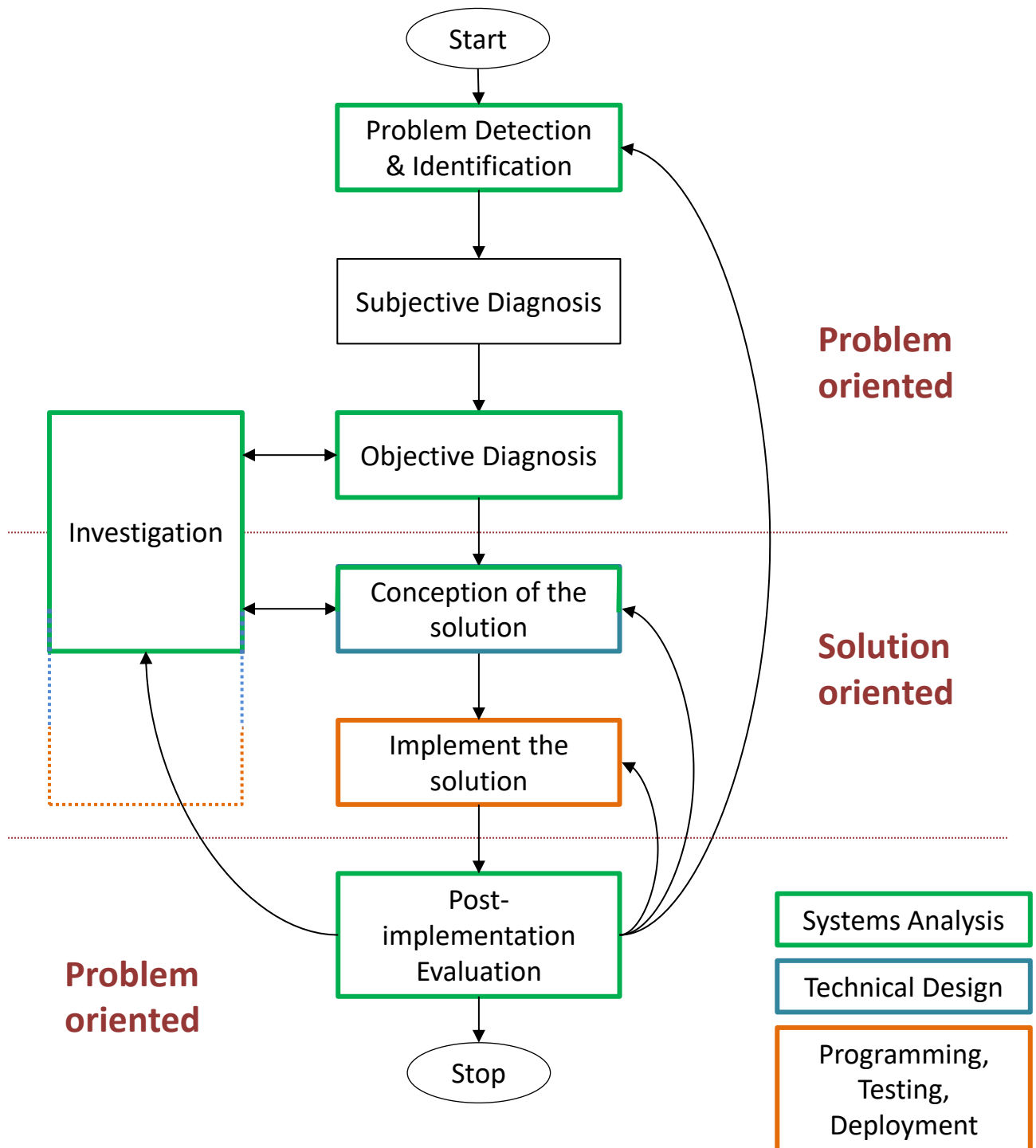
A predefined methodology is a template that has to be manipulated to suit the project.

Possible of Adaptations

- **Phases** can be dropped, expanded, reordered, split up, added, merged, ...
- **Process steps** can be dropped, expanded, reordered or added. Choice about timing, trigger, sequential, overlap, ... are other aspects to decide.
- **Artefacts** can be ignored. Or they can be enriched or adapted in any other way.
- **Methods** can be applied or not, adapted or other methods can be used instead or added.
- **Principles** can be applied or not, or applied in an adapted manner.
- **Tools, ...** chose what suits.

Mapping Methodological SD Approach on Problem Solving Approach

30/10/2018



Fundamental Steps in Software Development Approaches

Evolution of the Software Development Process:

*Small
and
Simple*

PROGRAMMING

Most fundamental
process for simple
software application

DESIGN

PROGRAMMING

Thinking about how to solve a need/problem and how to organise UI, database and the source code.

Reason: Avoiding spending time into re-organisation and rewriting of source code

DESIGN

PROGRAMMING

TESTING

Reason: Avoiding to deliver a bugged application to the end-users. Limiting the amount of bugs reaching the production environment.

ANALYSIS

DESIGN

PROGRAMMING

TESTING

Reason: Avoiding to solve the wrong problem or to design solutions without understanding the problem and its context.

*Larger and
complex systems*

Challenges as Initiative Scales Up

18/12/2018

- 1) Issue: Development of a very simple and small software application

Programming

----- Can be started right away



- 2) Issue: Development of larger software application

Software Design

----- Design and organisation of the software elements and components.

Assumption: Solution is known



- 3) Issue: What if the solution is unknown

"Solution Engineering"

----- Solution-oriented analysis: SA&SD analysis not only the system, but also the system's environment.

Assumption: Existing systems and environment are simple



- 4) Issue: What if systems and environments are complex

Systems Analysis and Design

----- Problem-oriented analysis: Study of existing systems and their environments.

Assumption: Problem is known



- 5) Issue: What if the problem is unknown

Complaints and symptoms surface. The problems are not understood or not even correctly identified (root cause). The existing systems and environment is insufficiently unknown.

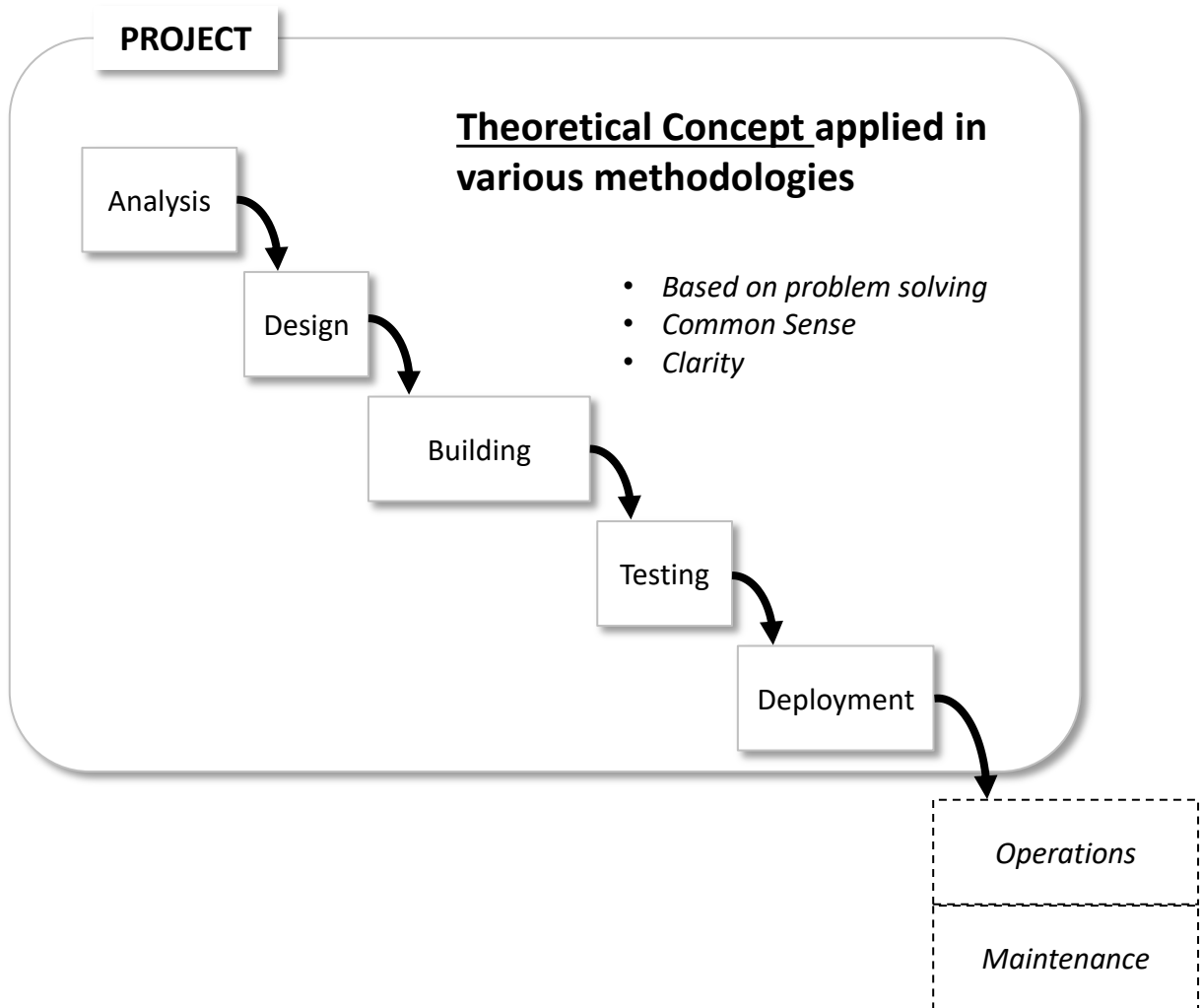
Systems Analysis and Design

----- DIAGNOSIS !!!

Continuing on next page

6) New issues emerge ...

- Sharing information
- Reuse of information, allowing multiple usage of information
- Coherence in the business logic across the company
- Ensuring real information problems (root causes) are solved
- Dealing with multiple users groups, different business domains and different objectives (sometimes conflicting)
- Conceive solutions that maximally exploit the possibilities of software and computers (no simple-minded solutions or impossible solutions)
- Integration in the environment
- Keep the system of systems coherent and clean
- Information security
- Ensuring evolvability of software systems (easy evolvability)
- Reuse software components, component based systems
- Ensuring the manageability of the entire collection of stored information
- Preserving the value of the information
- Information entropy
- Possibility to maximally exploit information
- ...

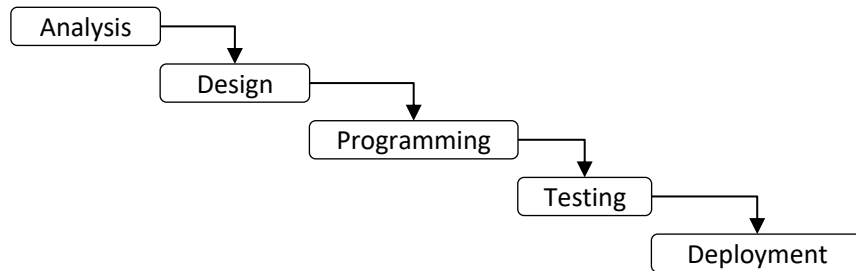


The concept of this simple SDLC is a pretty simple and straightforward process. It represents a logical approach that puts understanding the problem and the solution requirements before starting to build it.

Thanks to its modularity, it allows to deliver intermediate solutions, increments, systems and sub-systems, releases, ...

This flexibility makes this SDLC also very scalable.

What is the “Waterfall” ?



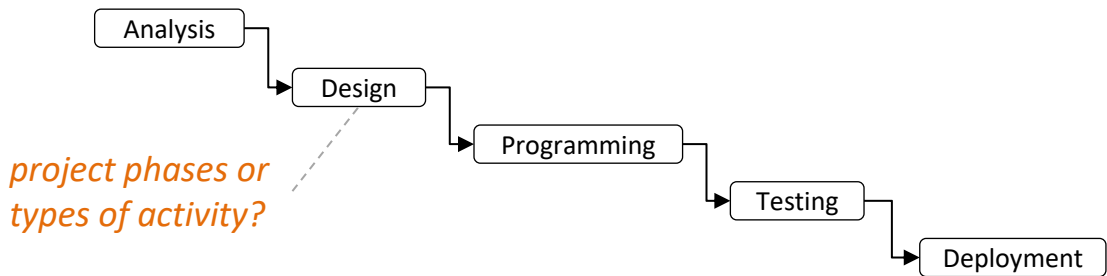
- A ~~methodology~~
- A methodological idea, concept, principle, pattern
 - based on earlier problems and general problem solving
- An SDLC

Remarks:

- A drawing, a model, that may slightly vary
- No established number of phases or defined names
- No official set of practices
- No official established rules and principles
- No official standard of the waterfall exists
- No official guideline on how to apply it
- But a lot of interpretations and assumptions
- A lot of principles and bad practices added to it → CONFUSION

How do we know how to apply the waterfall?

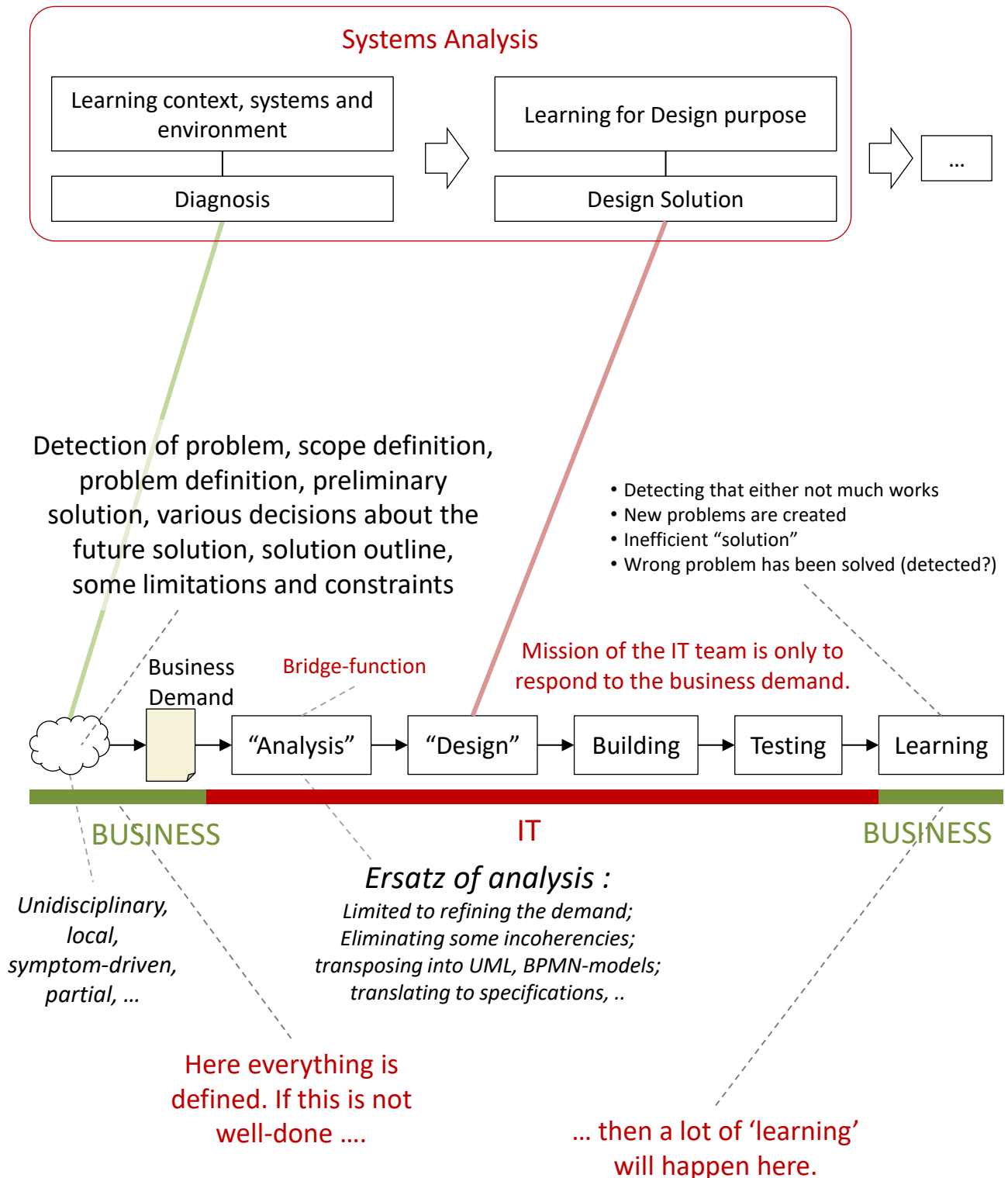
How do we know how to apply it correctly?



Practices Assigned to the Waterfall (to be abandoned):

1. The sequence of phases is strict and unidirectional
2. Complete a work for 100% before moving to a next phase
 - *100%? Is this possible?*
3. Phases may not overlap
 - *What with projects with sub-projects running in parallel?*
4. Only activities of one certain type may be executed during a phase
5. Follow methodologies and methods by the book
6. Big Up-Front Requirements
7. Big Up-Front Design
8. Plan in advance and in great details and don't do course corrections
9. Freeze work done earlier before moving to a next phase
 - *verifying and validating ≠ freezing*
10. Perform work imposed by the methodology (even if it is unnecessary)
11. Perform unnecessary work
12. Perform a lot of bureaucratic work
13. Write extensive documentation
14. Death march. Do not correct a situation by performing activities of a type performed earlier or don't jump back to an earlier phase. Continue until delivery.
15. If a type of activities that have been performed earlier have to be executed, you have to move the whole project to the corresponding earlier phase.
16. Testing happens only at the end of the project during the Test phase.
17. "Big bang" deployment
- ...

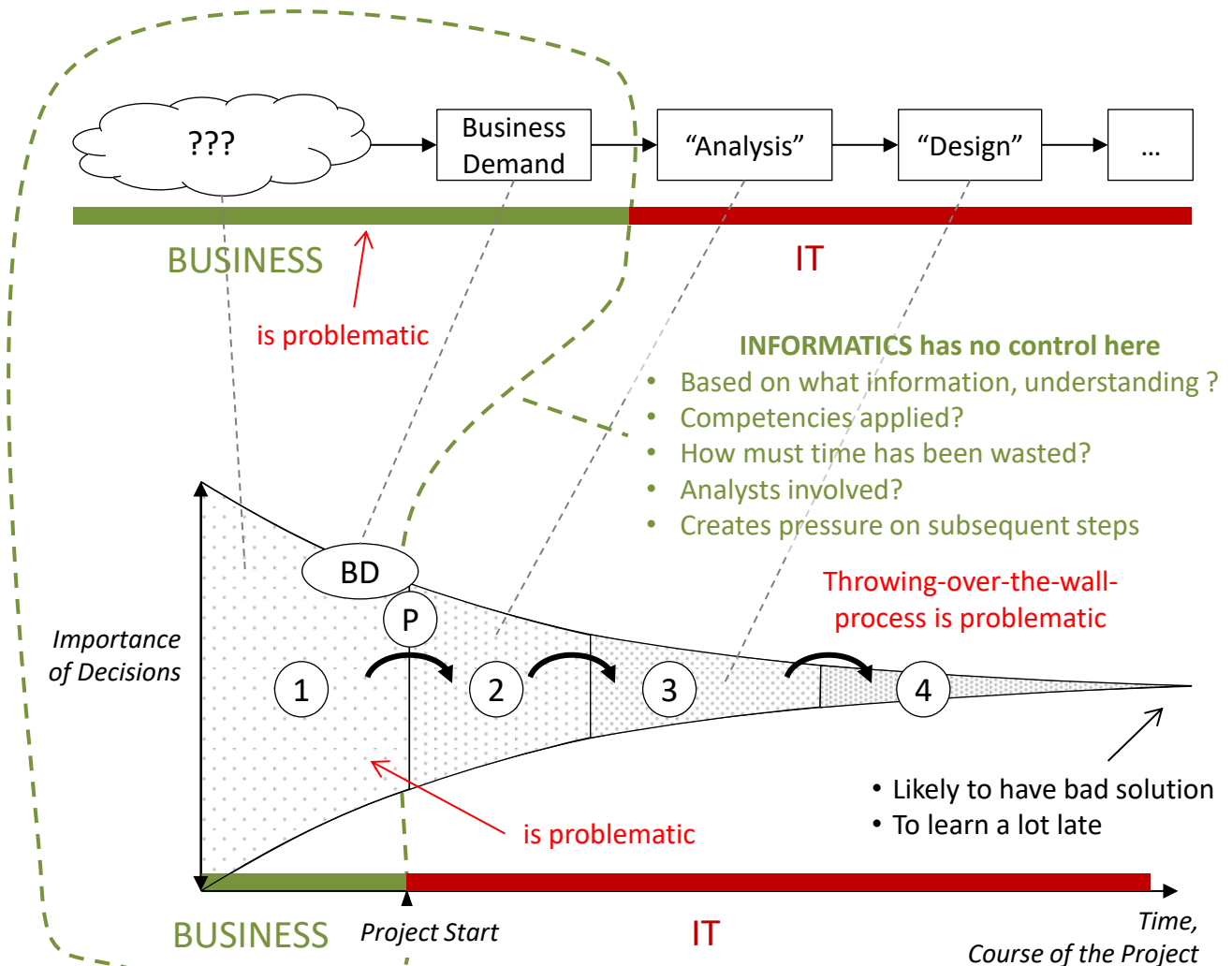
Groundless interpretations of the Waterfall-diagram
lead to nonsensical practices of the Waterfall SDLC



How can meaningful innovation happen in this context?

Common Bad Practices

10/01/2019



(1) The business stakeholders, SME's, end-users determine the problem, what they want, the scope and the outline of the desired solution.

(2) The analyst receives (gathers) requirements, refines, checks, translate, models, turn into specifications

(4) The developers build the requested software system

(3) The IT analyst, designer, SE or developer(s) elaborate the technical solution.

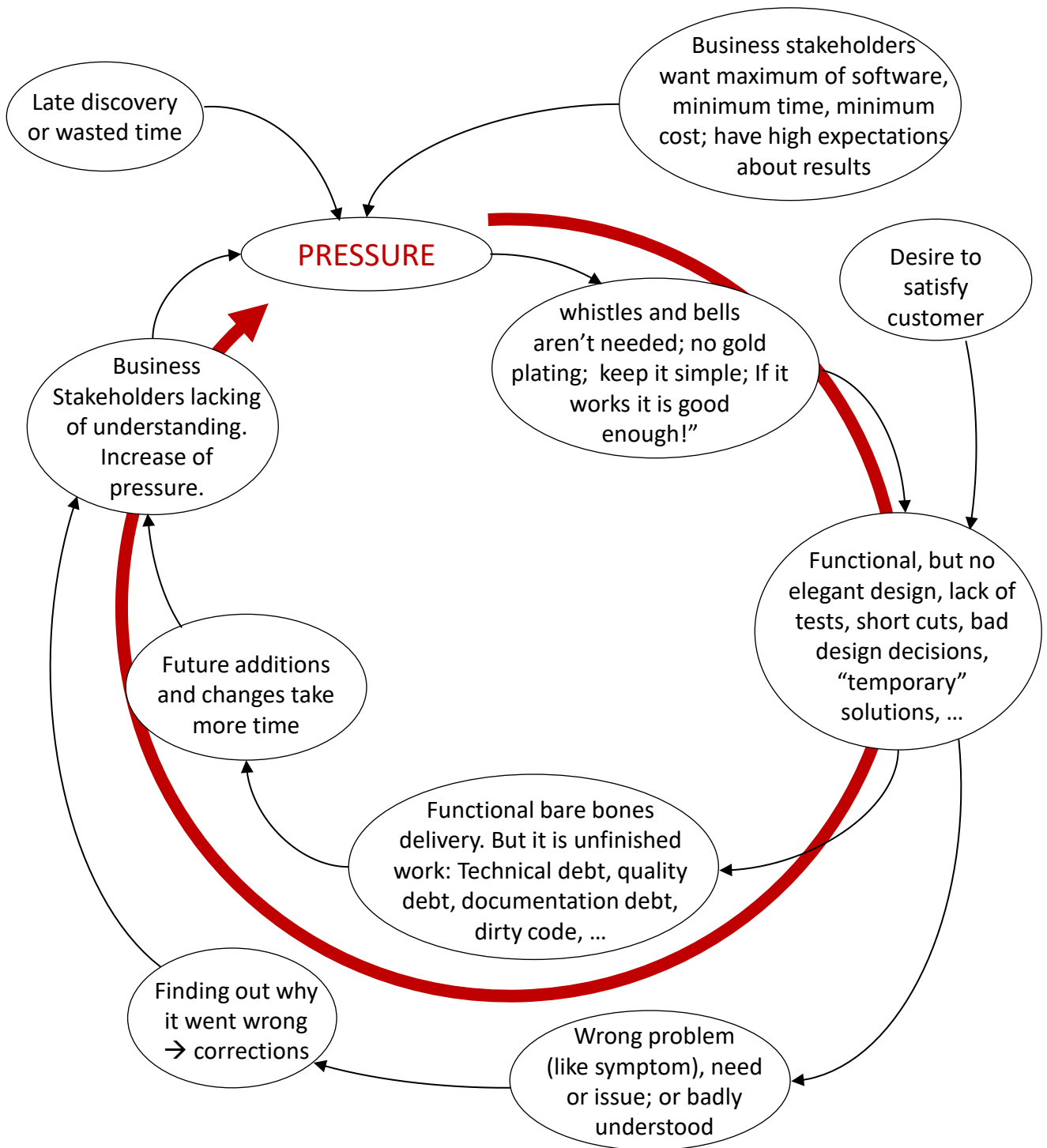
P

Plan

- Scope
- Delivery Date
- Budget
- Solution Outline

Failure Cycle

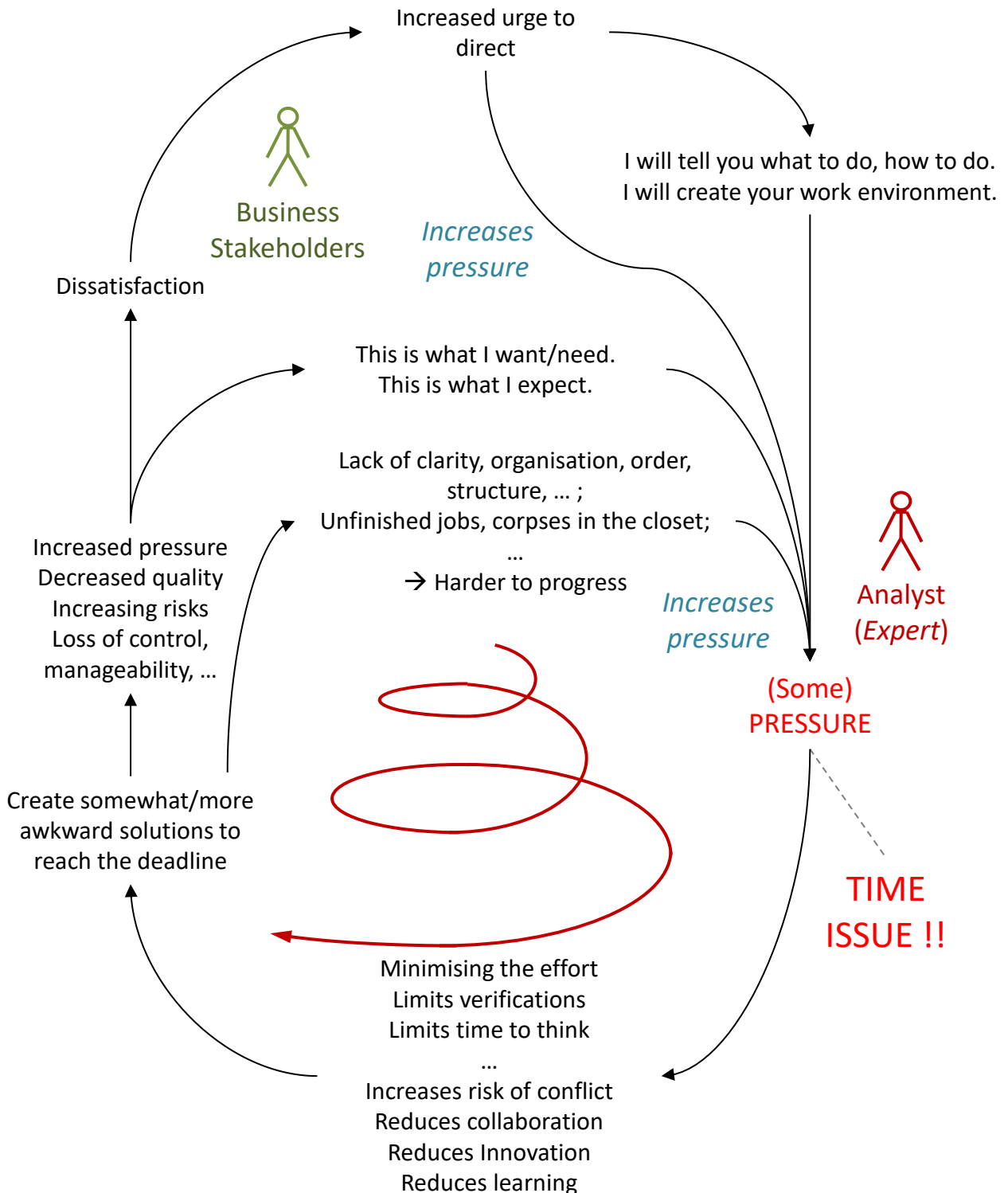
15/03/2019



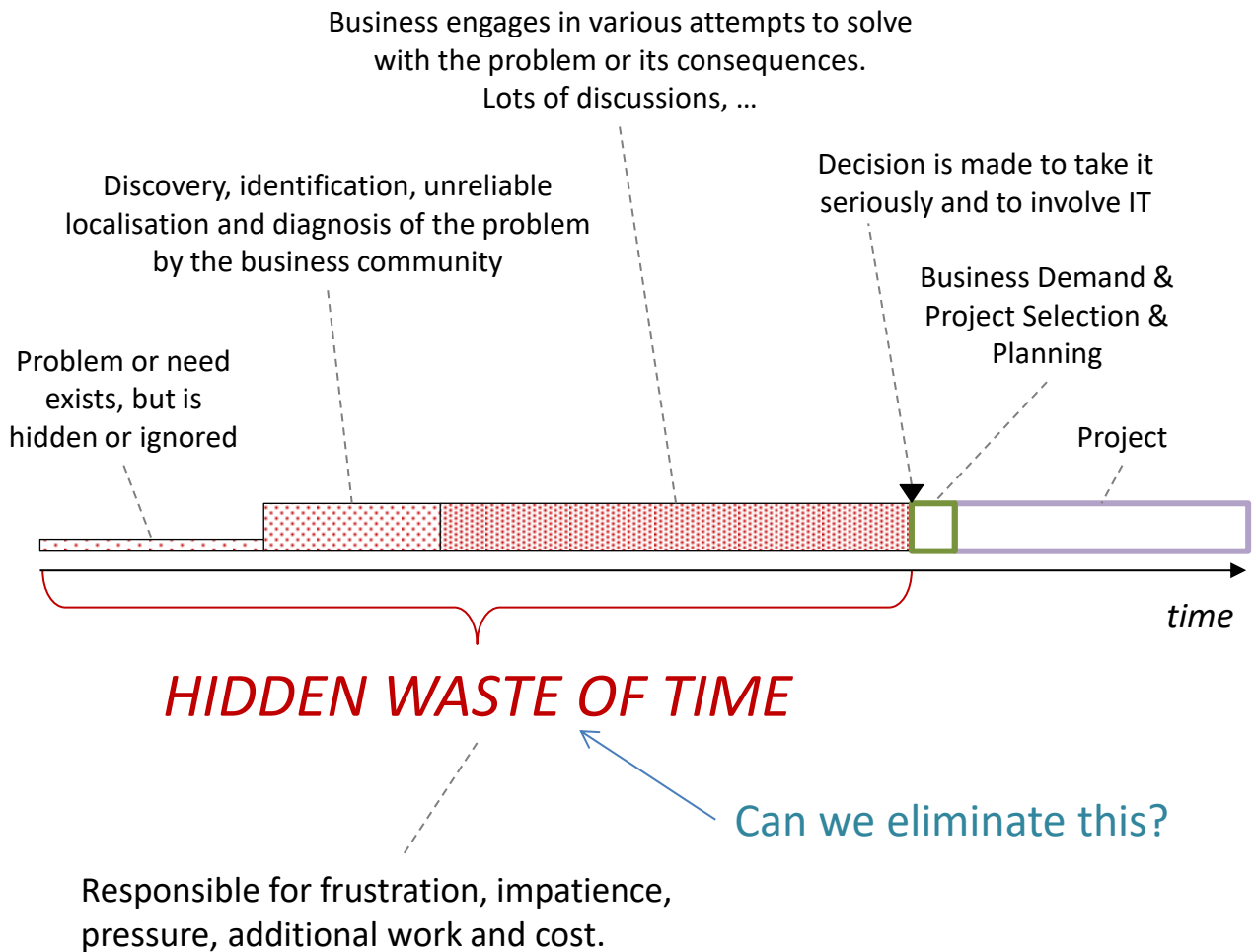
Causes: Part of the Analysis is performed by business people, by an inexperienced analyst or by developers; no verification; no alignment of expectations with delivered results;

The effect of the "poor solution" shouldn't be ignored either.

Deadly Spiral of Debt, Pressure and Lack of Quality



How to Speed Up Software Development



Since a lot of time has been wasted:

- Needs and problems had the time to exist, to evolve and to cost.
- Pressure is created on the project. Once the decisions made, once the demand is transferred then, suddenly, everything must go fast.
- While no guarantee that the demanded solution will solve the problem or even that the tackled problem is the right problem.

How to Speed Up Software Development (Answer)

1) CONTINUOUS ANALYSIS

to hunt for problems and improvement opportunities

- by analysing the implemented systems;
- by analysing goals, policies and plans
- and by continuously gathering ideas and complaints

2) Don't design flawed solutions

The solutions designed today cause the problems of tomorrow. Information solutions must be designed by top professional analysts, architects and designers (particularly if system or solution is critical).

Try to do the right things right from the first time.

3) Have models, rules and principles matching the implemented systems

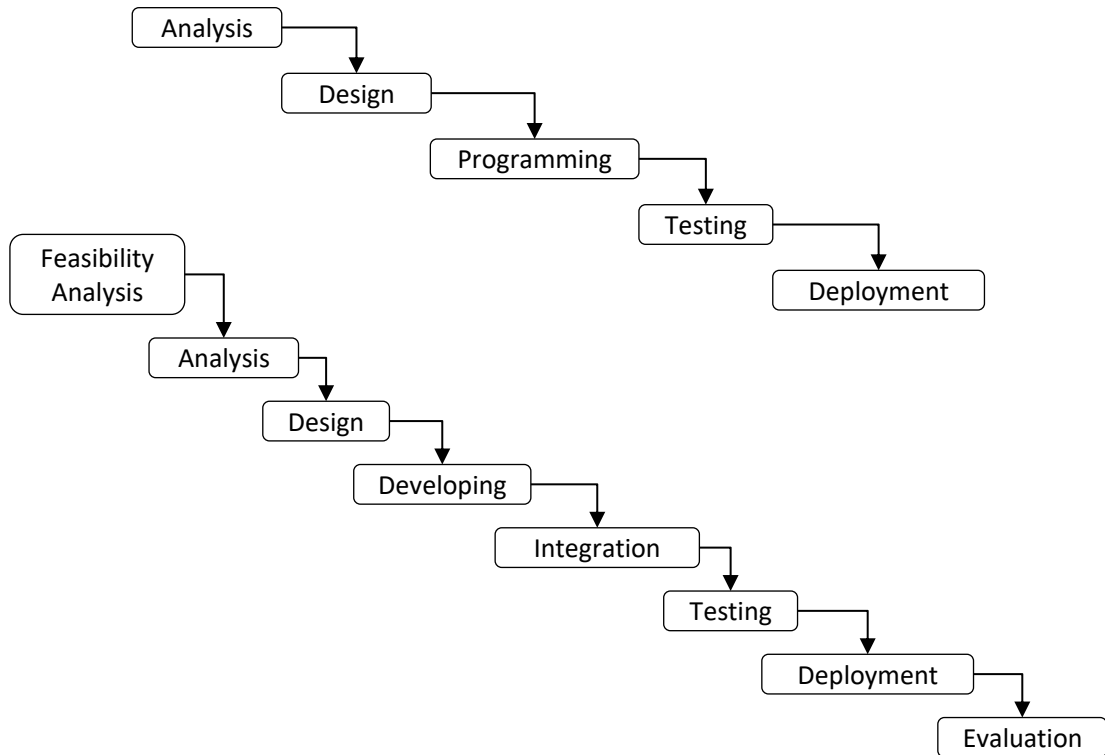
Never document the “As-Is”. It is the former “to-Be”

4) Use an architecture, components, services

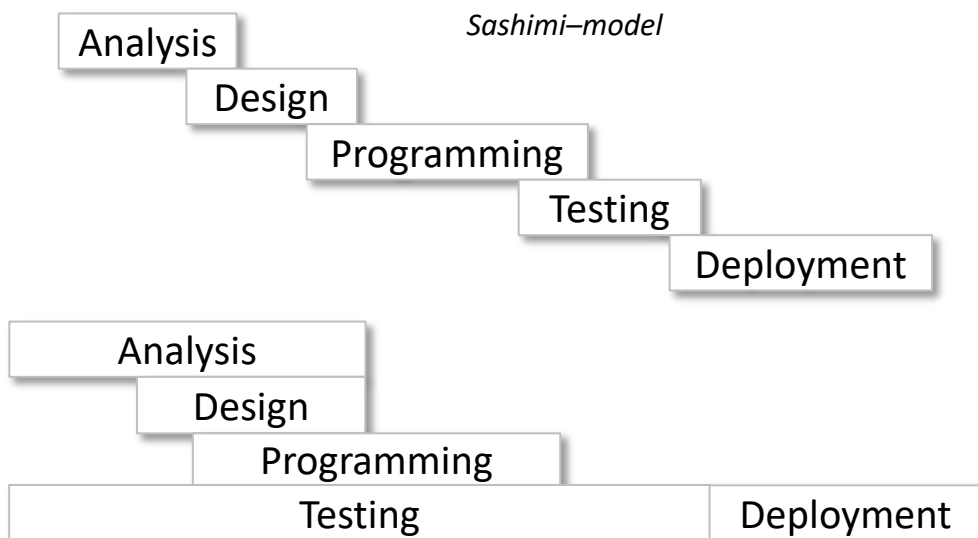
5) Design for Re-use

- Preferably linked to goal, outcome rather than to a defined solution
- Focuses on and maximisation of value creation
- Innovation
- Doing more the right and necessary thing and less responding to demands, wants, preferences
- Servant Leadership
- Structure and organisation
- Scalable
- Multi-disciplinary
- Collaborative
- Freedom not to follow rules, principles, structure, ...
- Iterative
- Incremental
- Modularity in waterfall - phases or stages, releases, sub-projects, sub-tracks, ...)
- Continuous investigation-driven over demand-driven
- Preference for proactivity over reactivity
- Righteousness, quality and steadiness over Speed and Deadlines
- Sustainable
- Take decisions as early as possible, but not before understanding
- Put risky, structuring and critical elements first, then most valuable, and then other aspects.
- Continuous Learning & Root Cause Analysis
- Planning and Analysis over testing and rework
- Predictive and Adaptive
- Holistic - including in time
- Top-down, bottom-up and bottom-top-down
- Understanding precedes Action - Know what you do before you do it
- Analysis-driven
- Model-based
- Integrated
- Feedback
- Flexible product over flexible process
- Component based and solution scalability when possible and useful
- Build stable systems – foundation, layers, agents, ...
- Reuse
- Avoid solving symptomatic solutions
- Conscious execution - Know why you are doing something, why not, why this way and not the other way
- Situational Process Decisions - Adapt the process
- Testing during the whole project
- Remove waste - Don't do unnecessary work.

Adding Phases and Activities

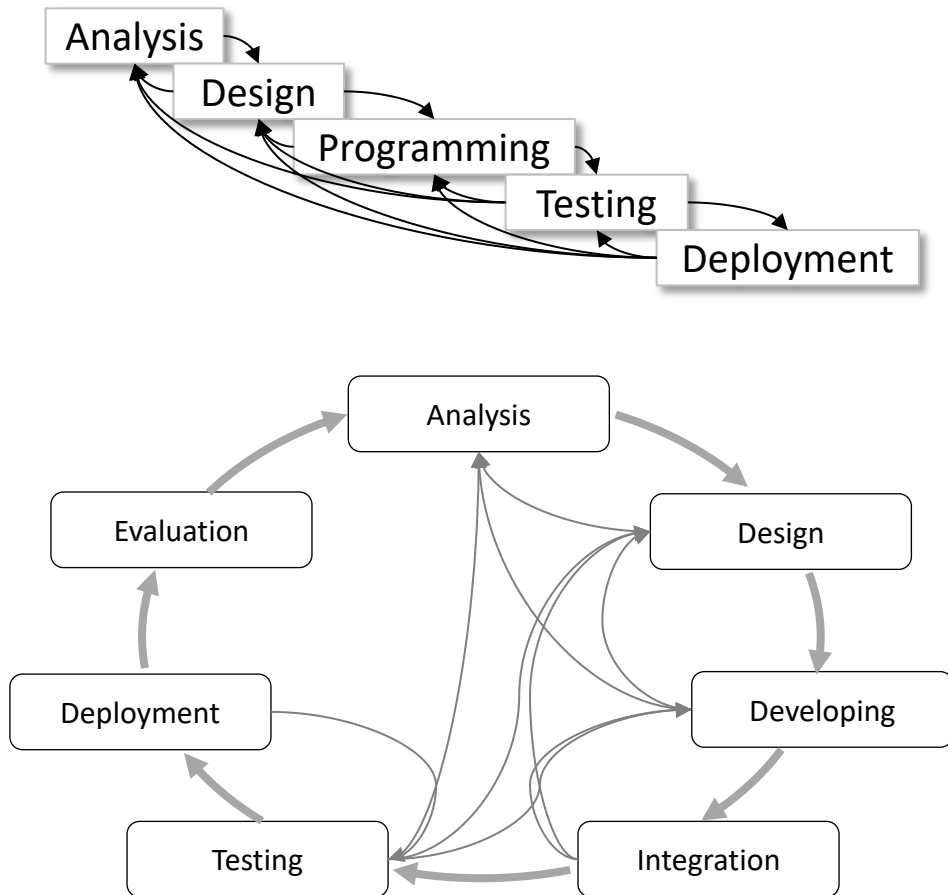


Overlapping

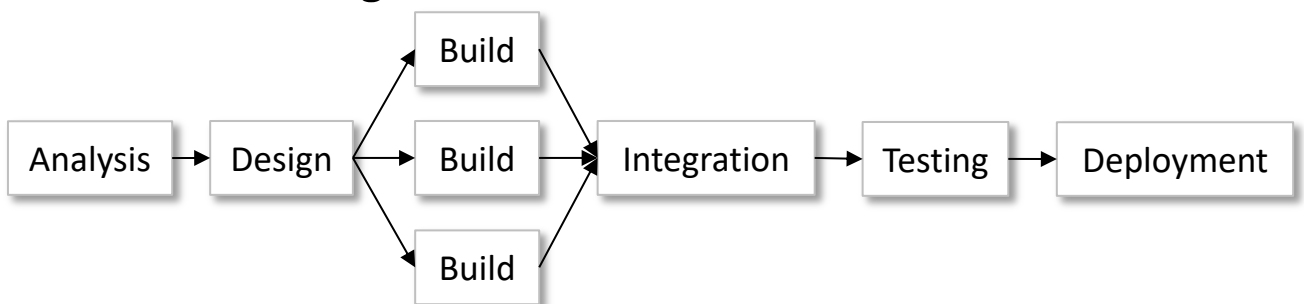


Elaborating the Basic SDLC

Iterations

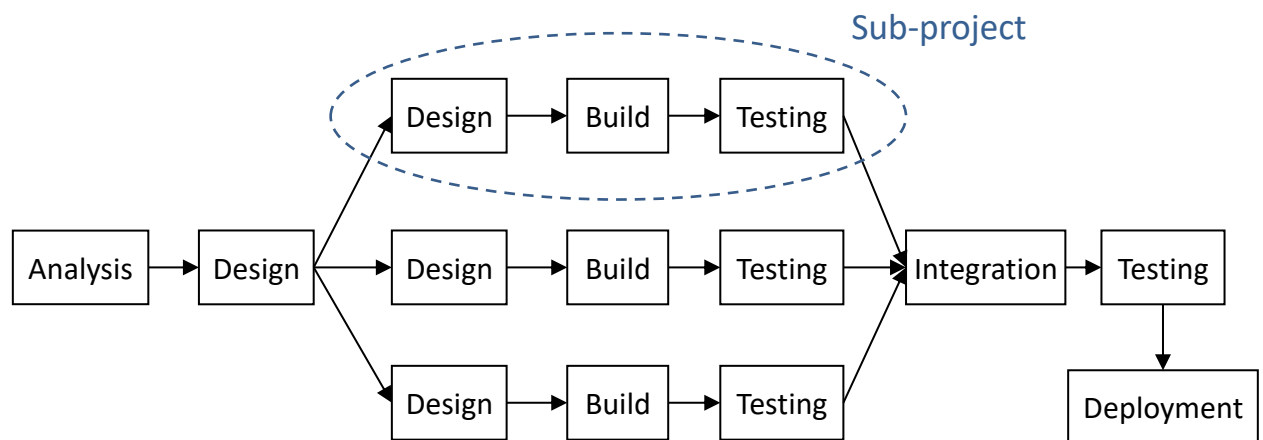
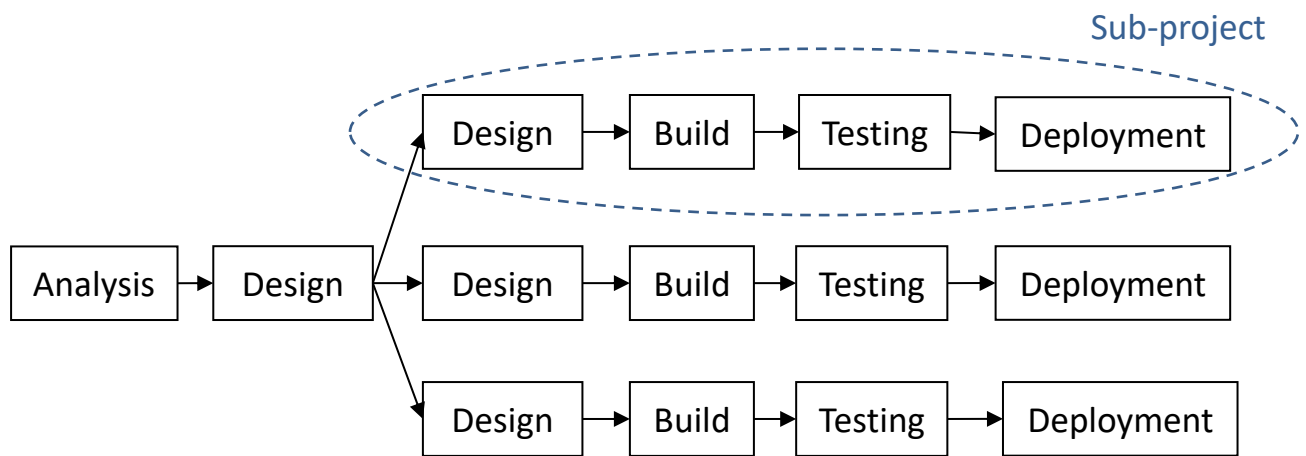


Parallel Building



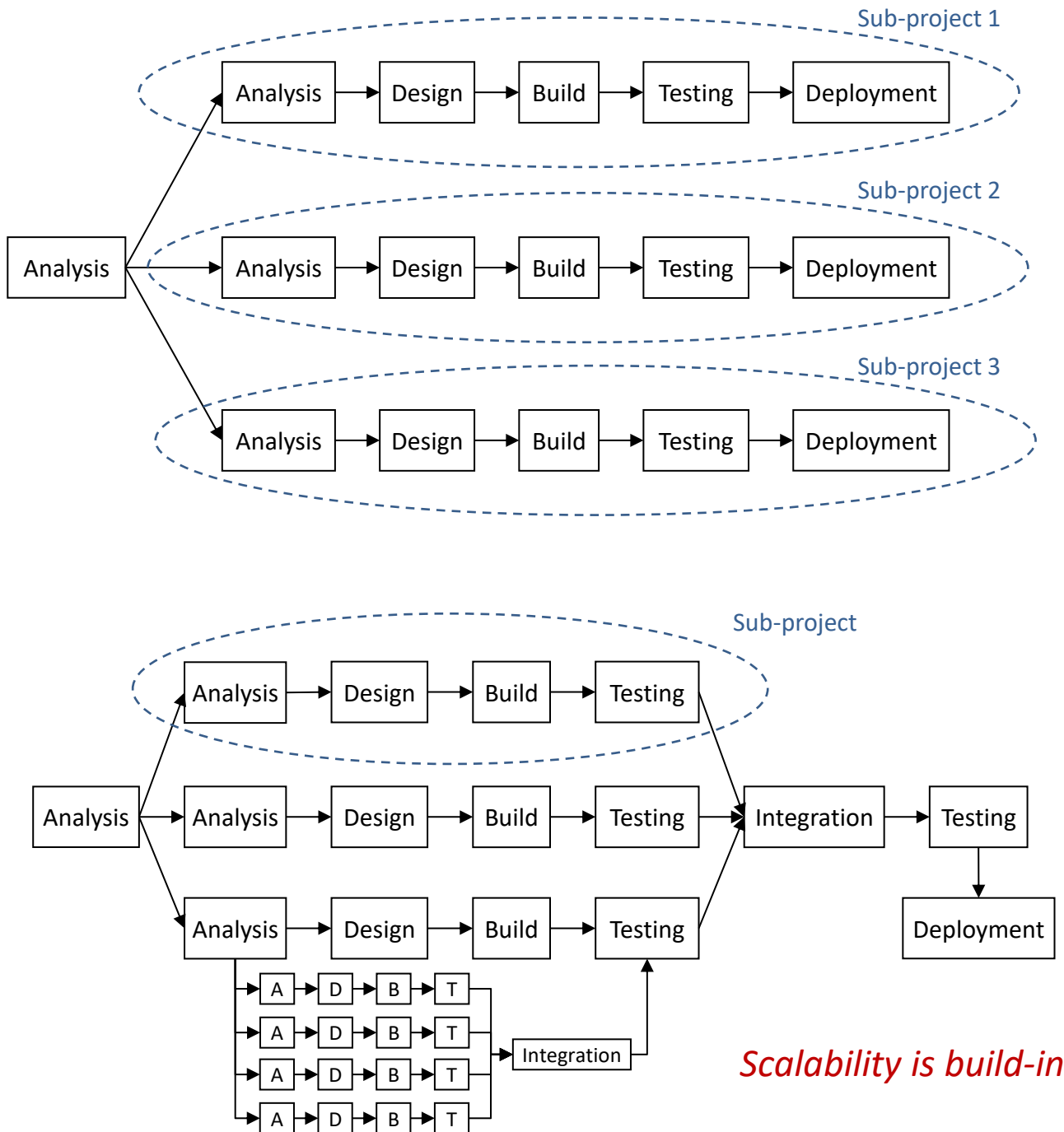
Phases used as Building Blocks

Parallel Development



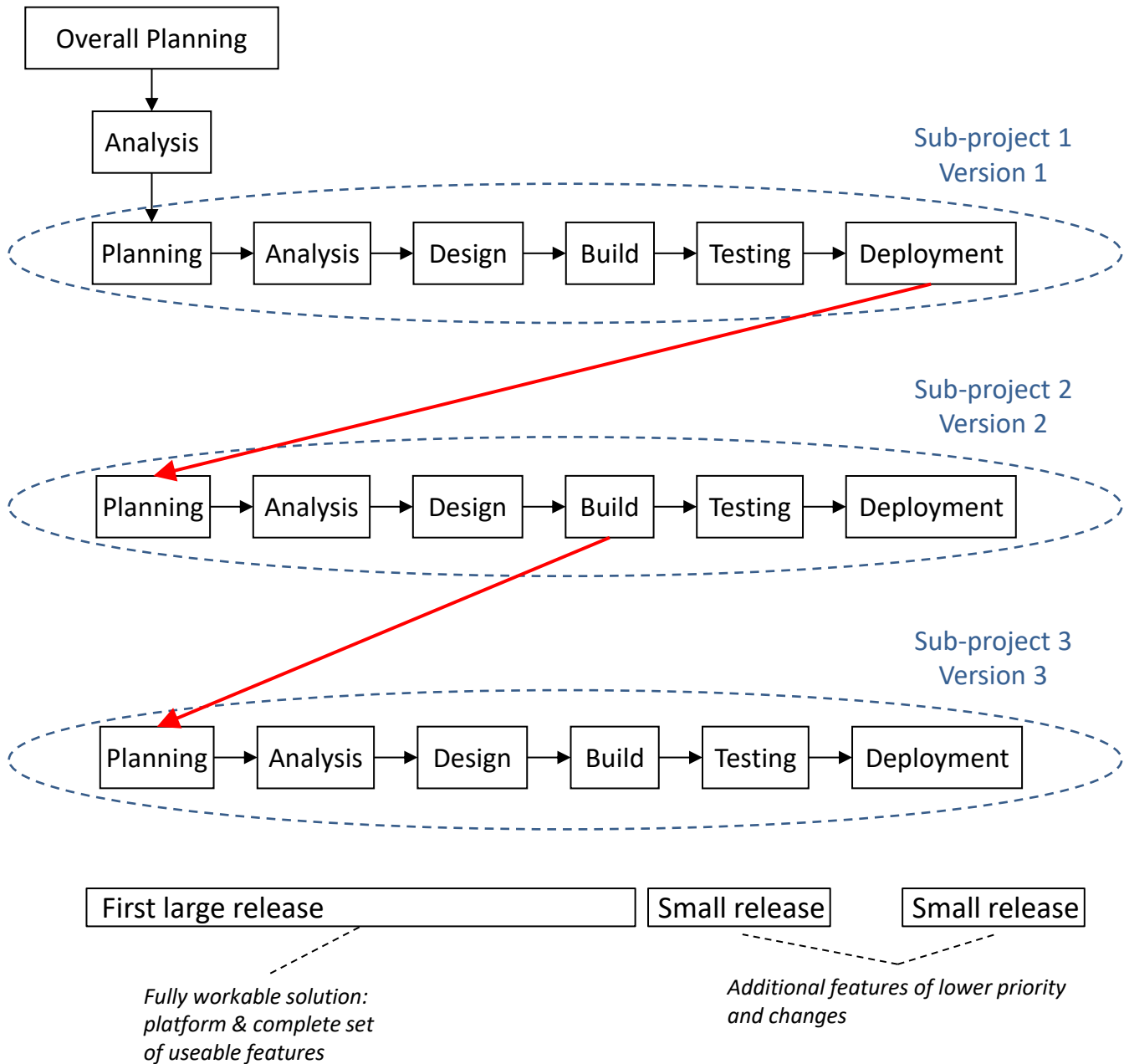
Elaborating the Basic SDLC

30/10/2018

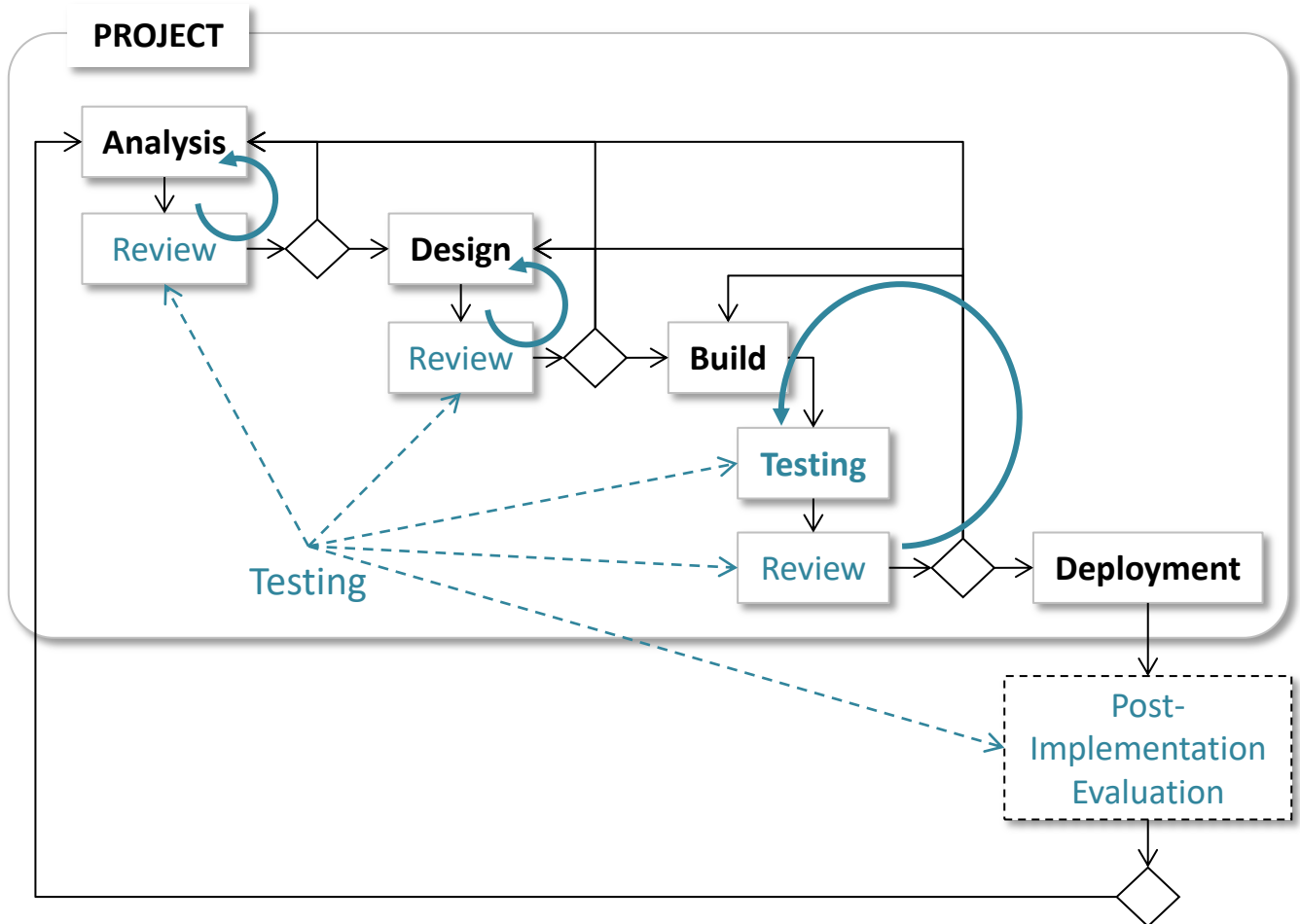


Phases are like building blocks. They help to organise and manage projects, and to focus the attention without limiting the activities.

Phased Development



Feedback and Testing during the whole process



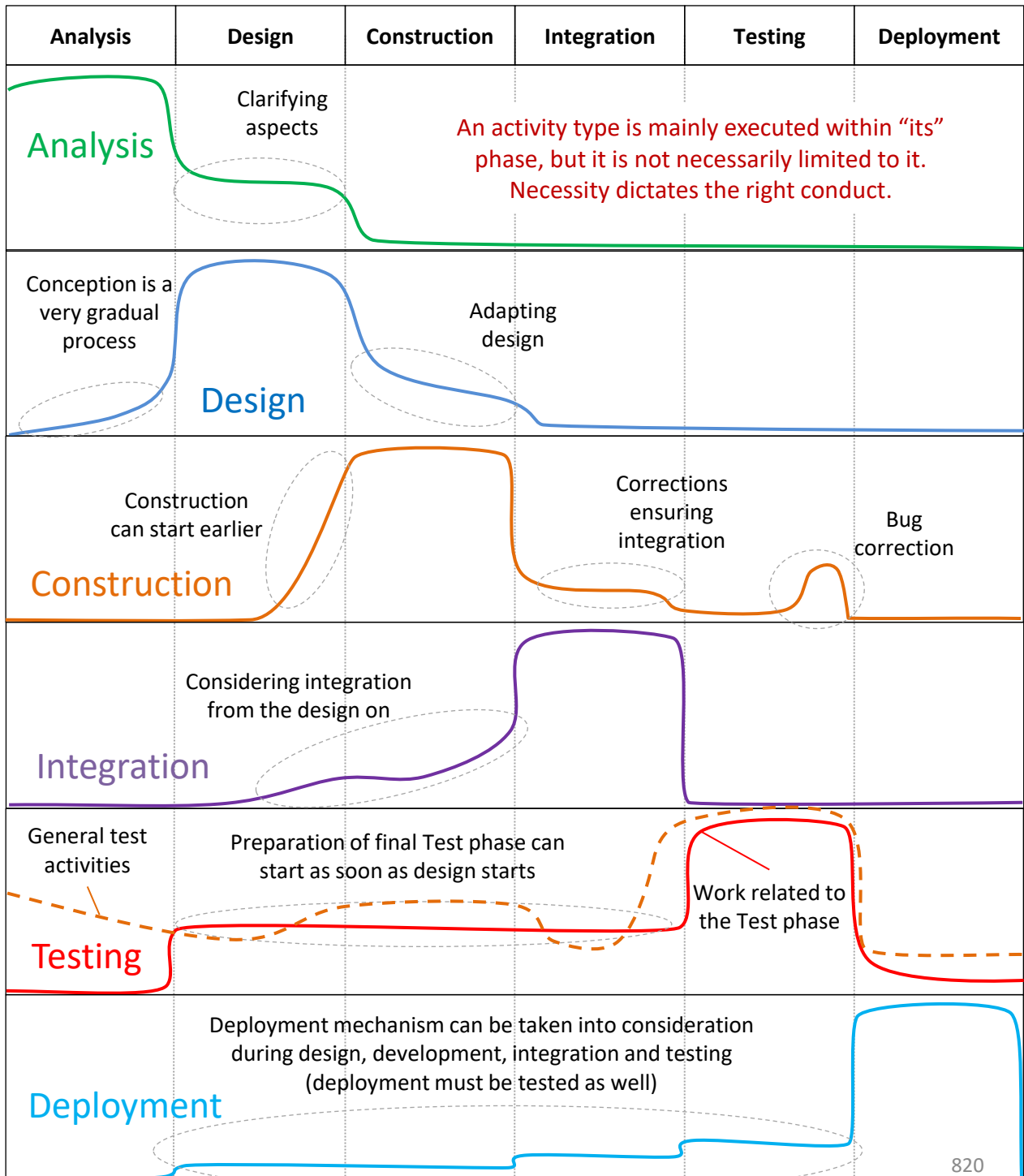
Flexibility in Basic SDLC 'Execution'

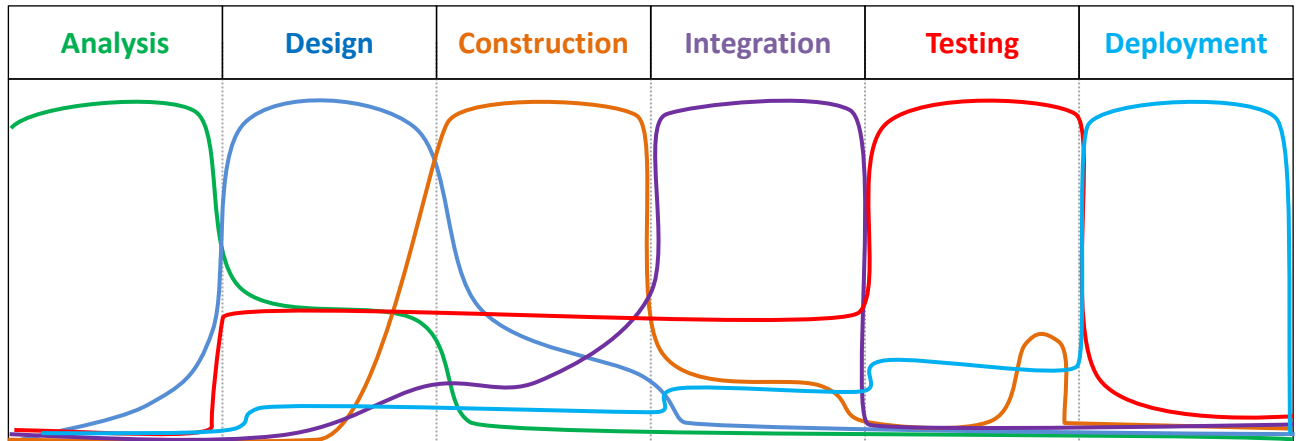
30/10/2018

A possible execution of a the SDLC

PROJECT PHASES

ACTIVITY TYPES



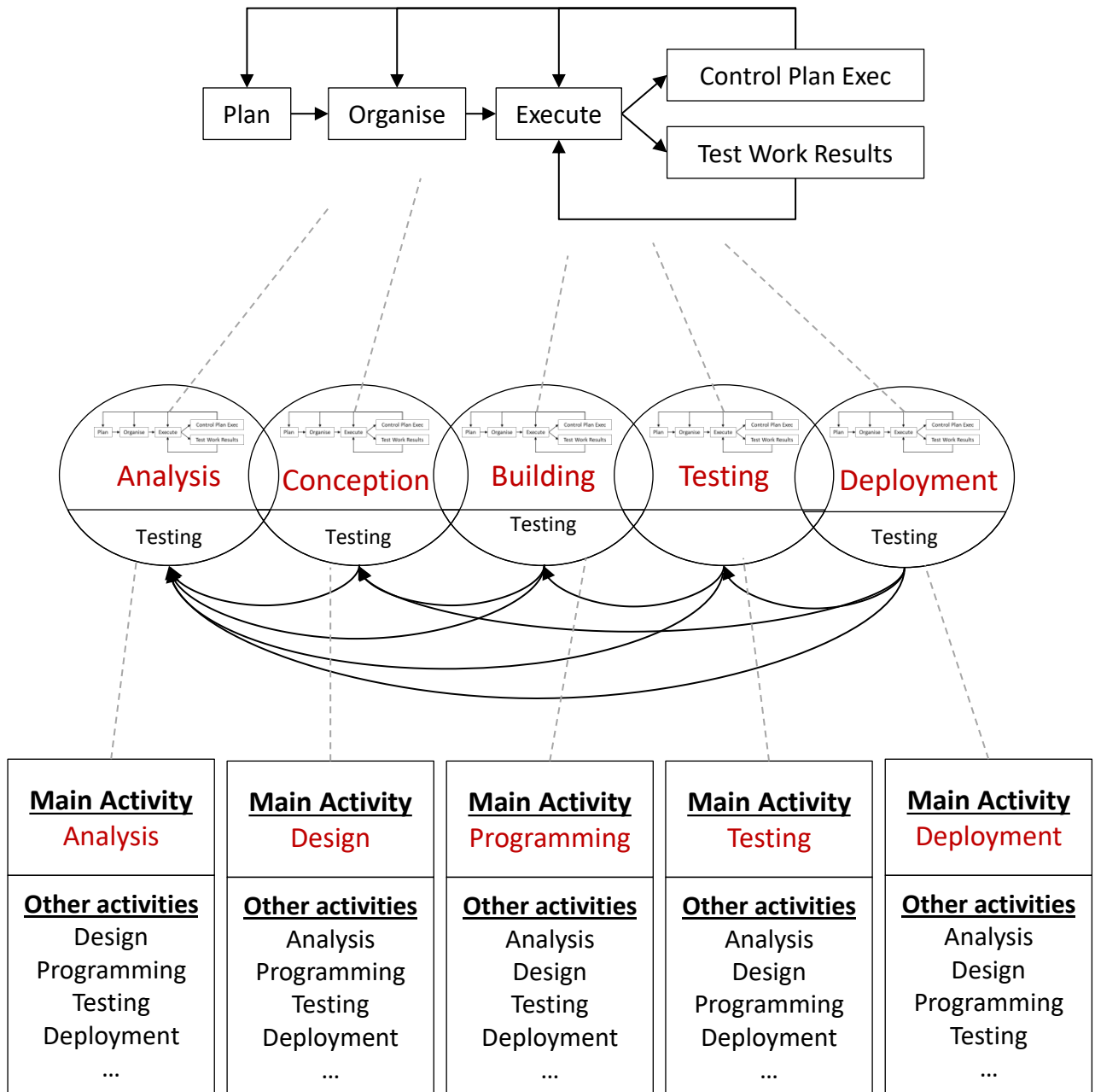


Why tasks are normally executed with overlaps?

- The **mind thinks often ahead** on certain aspects or parts of the future product and on subsequent work. For example:
 - During the Analysis phase, the Analyst may already think on possible solutions (Design). (S)He can already prepare some sketches and models of the future solution.
 - Integration and testing have to be taken into account well before the actual project phase.
- Despite the effort, it happens that a task has **not be completed for 100%** or that the result is **not for 100% correct**.
- **Some parts may already be settled and ready for future work** to be started, while other parts still need additional work.
 - For example: Some parts of the design may be ready for construction while other parts still have to be further defined.
- There is **not always a clear border between two types of different activities**.
 - For example, integration and testing are also a matter of design.
- An **activity requires often specific activities like planning and preparation**. These activities may happen before the actual phase. Example: Writing test scenarios for the test phase can start once the Analysis phase is finished.

Elaborating the Basic SDLC

30/10/2018



Types :

- **Forms:**

Proof of Concept, mock-up screens, simulation, paper prototypes, computer animations, video, scripting language, ...

- **Categories:**

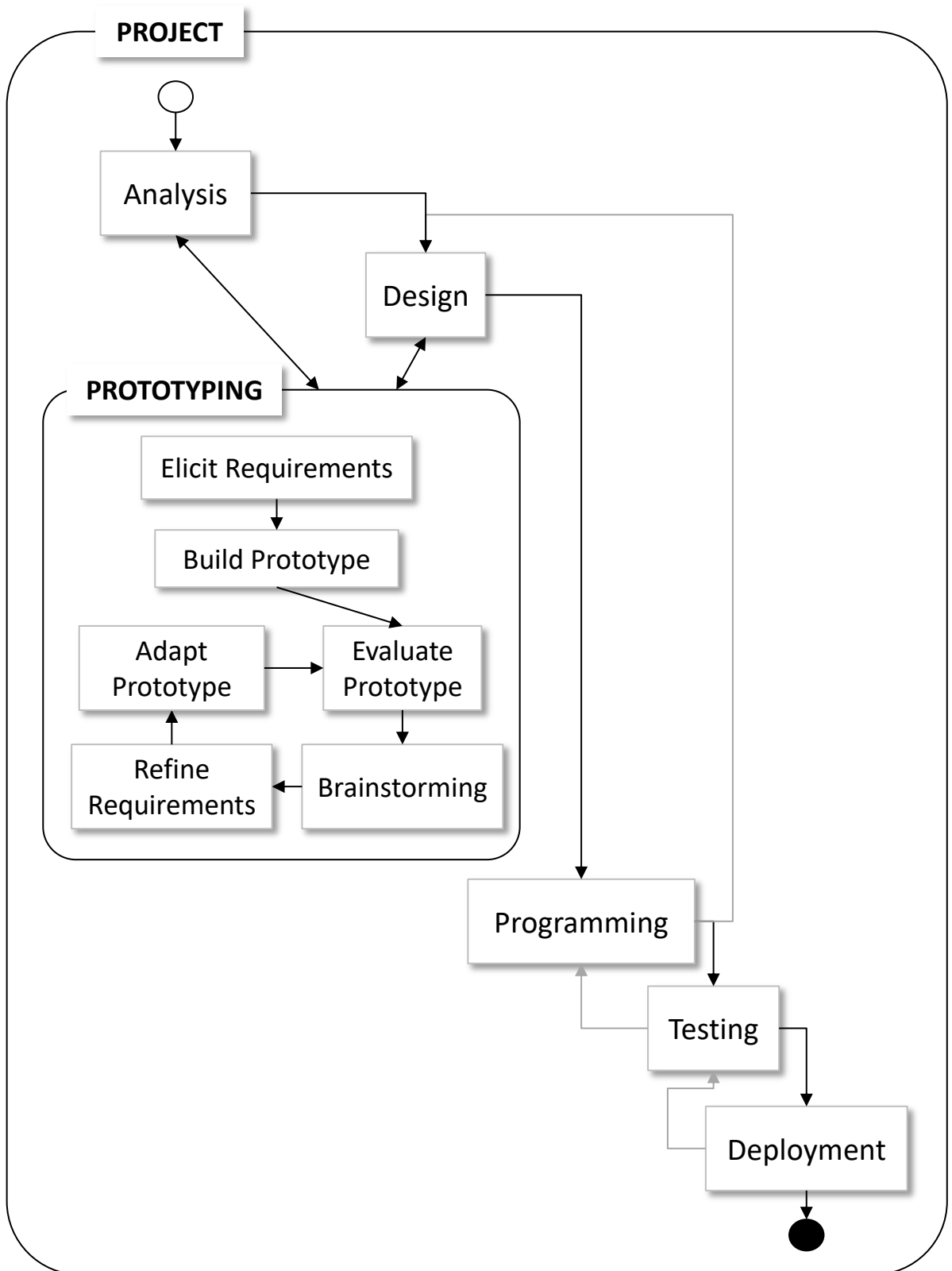
discovery prototype, rapid prototypes, iterative prototype, developmental/evolutionary prototype (to be part of the final system), ...

- **Strategies:**

horizontal prototyping, vertical, task-oriented, scenario-based

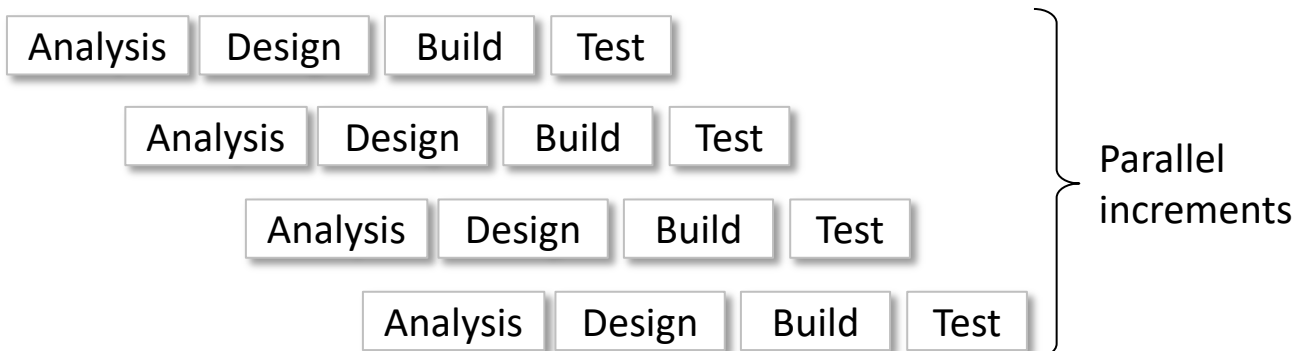
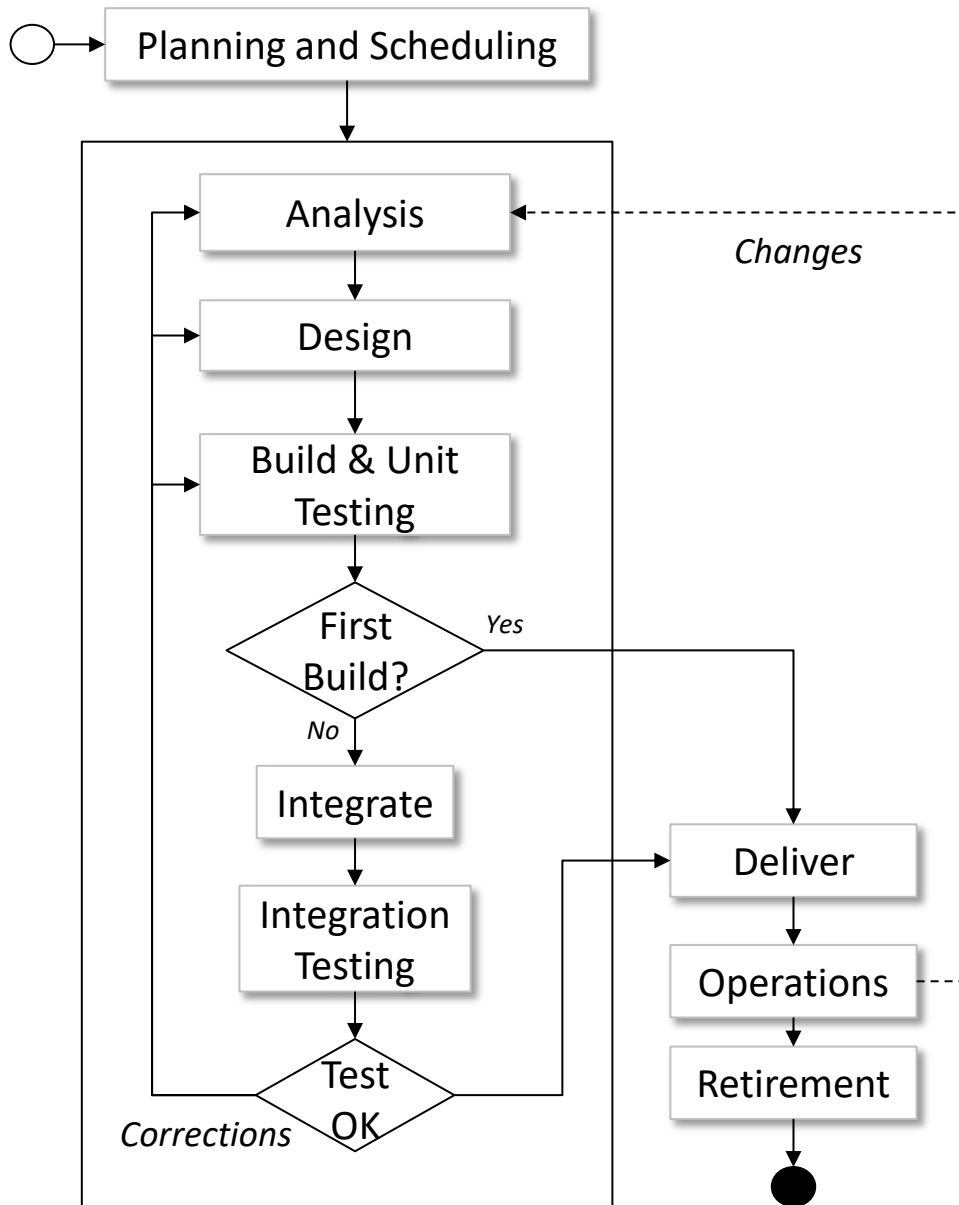
Prototyping in the Basic SDLC

30/10/2018



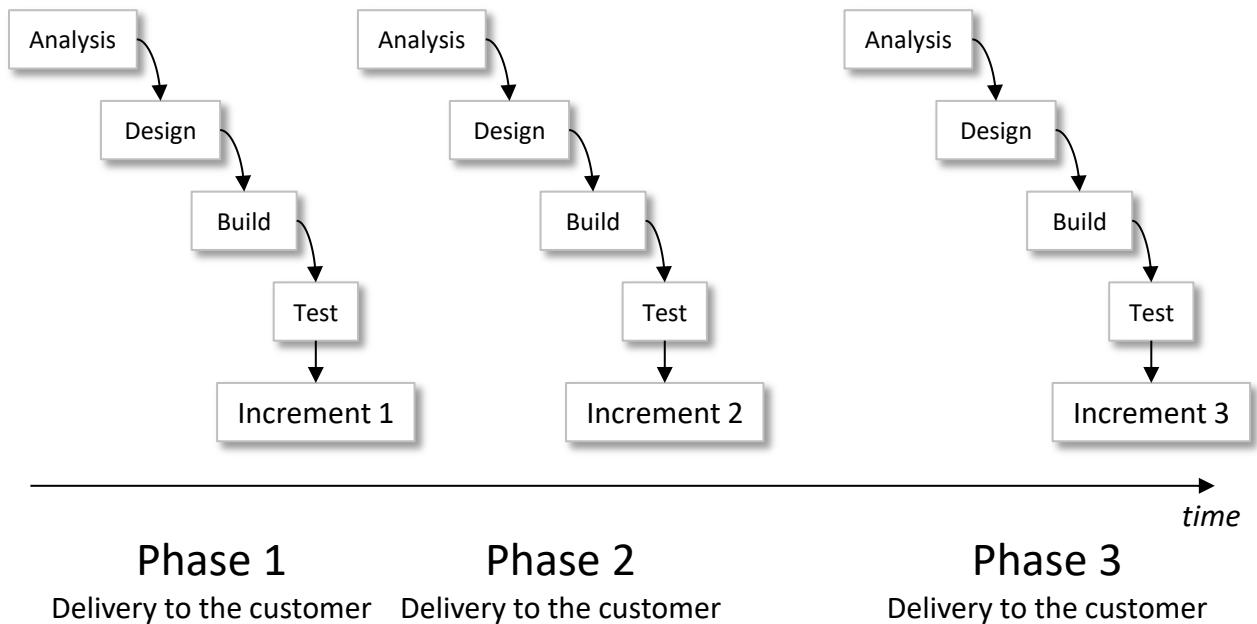
Incremental Basic SDLC Model

30/10/2018



Incremental Basic SDLC Model

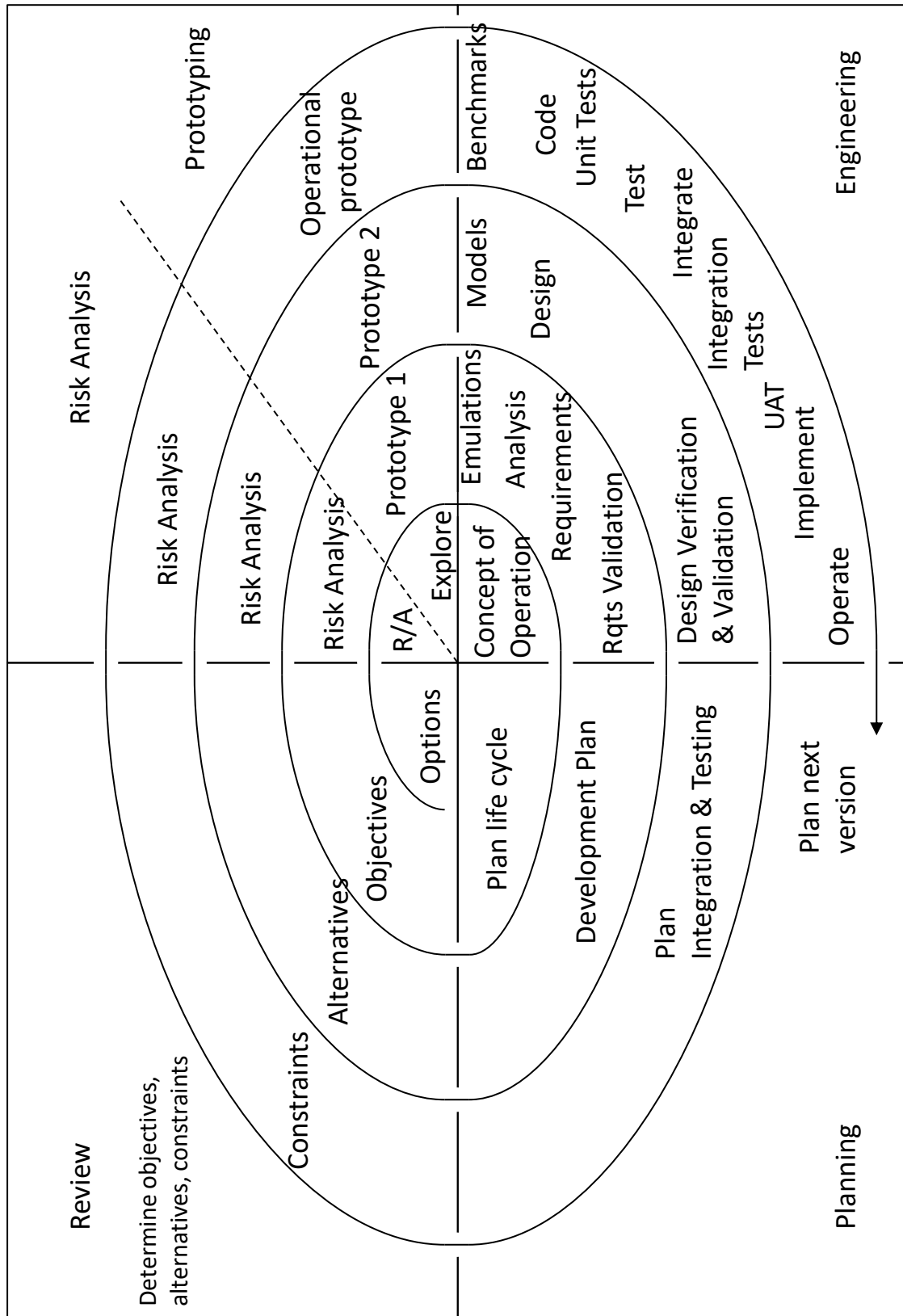
10/01/2020



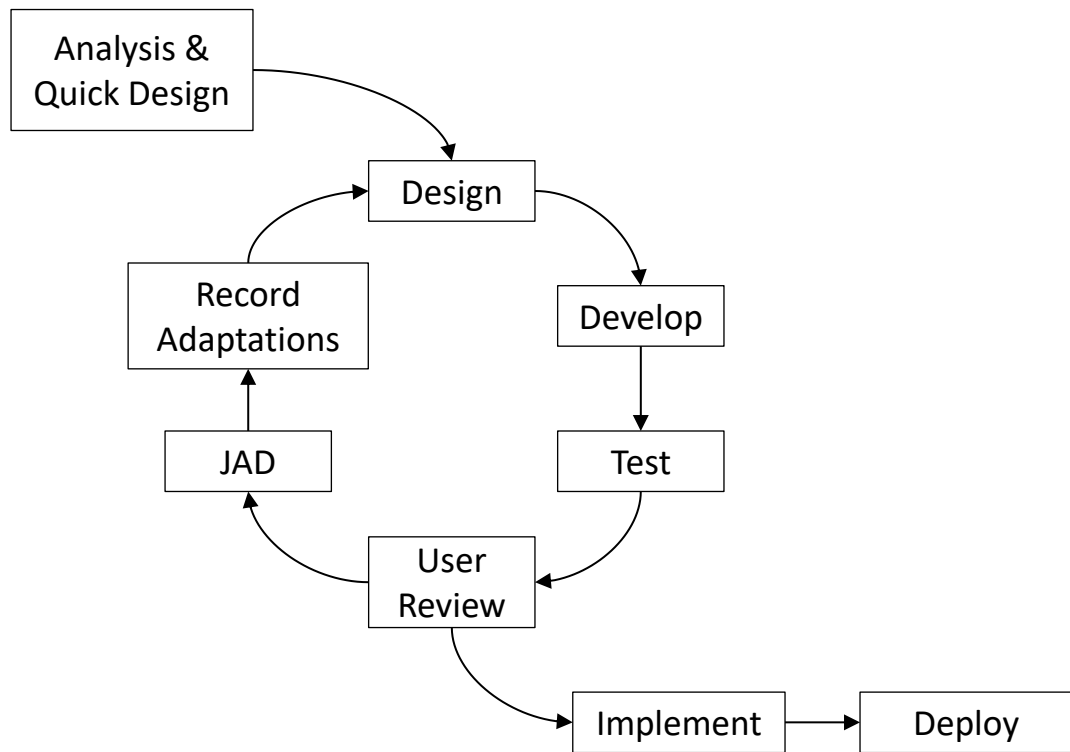
After each phase the customer has a workable system which is adapted and/or expanded in the future phases.

Spiral Model (Barry Boehm)

30/10/2018



Rapid Application Development (RAD)



Other Types of Approaches

30/10/2018

- Joint Application Design (JAD)
 - Exploratory Programming
 - Lean
-
- Developed in-house
 - Turnkey
 - Off-the shelf
 - Contracted out - Offshore

- Project Management Methodology
- Software Development Methodology
 - Analysis and Design Methodology
 - Software Integration Methodology
 - Software Testing Methodology
 - Data Migration Methodology
 - Software Deployment Methodology
 - ...

METHODOLOGY?

- Based on **principles**
- Based on a **set of** common/best(?)-**practices, methods, ...**
- **Adaptable** process
 - One-size-fits-all methodologies don't exist
 - Ready-to-use methodologies don't exist !!
 - A Methodology \neq Procedure !!

(can't be executed like a procedure)

Traditional Main Processes

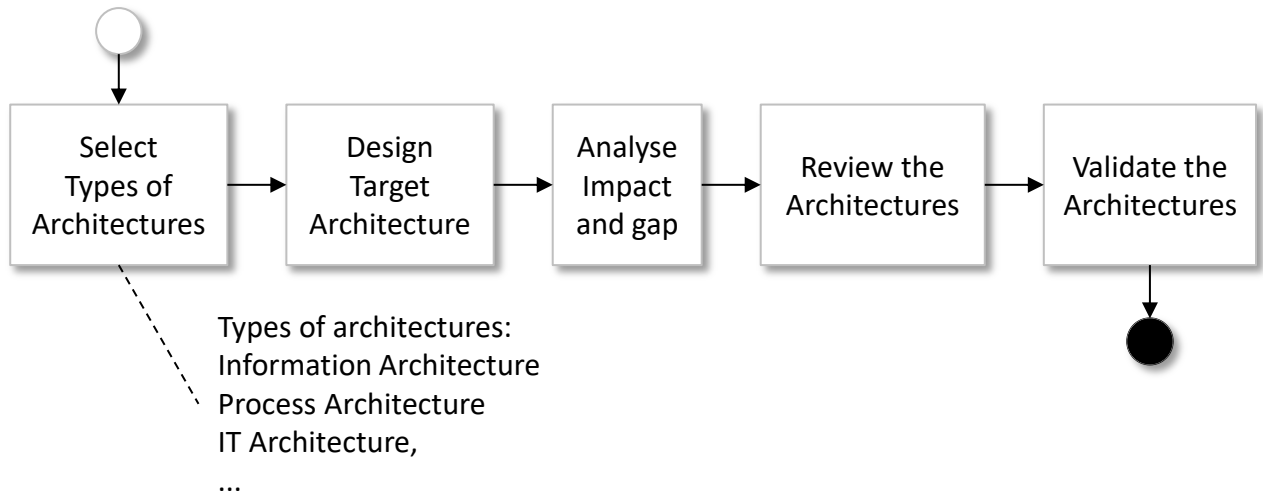
30/10/2018

Program / Project Portfolio Mngmt	Project Proposal	Project Evaluation	Project Selection	Project Prioritisation	Project Planning	Project Authorisation	Post-implementation Product Evaluation
Project Mngmt	Project Launch	Initial Planning	Project Set Up	Project Planning	Guiding and Managing the Project Execution	Project Evaluation	Project Closure
Business Analysis	Enterprise Analysis	Business Analysis Planning and Monitoring	Reqs. Elicitation	Reqs. Analysis	Requirements Management and Communication	Solution Assessment and Validation	
Architecture	Select Architecture / Architecture Types	Design Target Architectures	Analyse impact Architectures	Perform Gap Analysis	Review and Validate Architecture		
Software Development	Functional Analysis	Design	Programming & Unit Testing				
Environment Management	Environment Planning	Environment Installation	Environment Testing				
Testing	Test Preparation	Test Planning	Test Set Up	Test Execution	Test Evaluation	Classification and Prioritisation of Test Results	
Data Migration	Plan Data Migration	Design Migration Process	Develop Data Quality Control & Migration programs	Test Programs	Migrate Data	Test Data	
Deployment	Planning Implementation	Implementation	Configuration	Data Migration	Planning Implementation	Deployment	
Training	Assess Training Needs	Plan Training	Prepare Training	Provide Training	Evaluate Training Results		
Documentation	Plan Documentation	Structure Documentation	Write Documentation	Test /Review Documentation	Release Documentation	Assess Documentation	

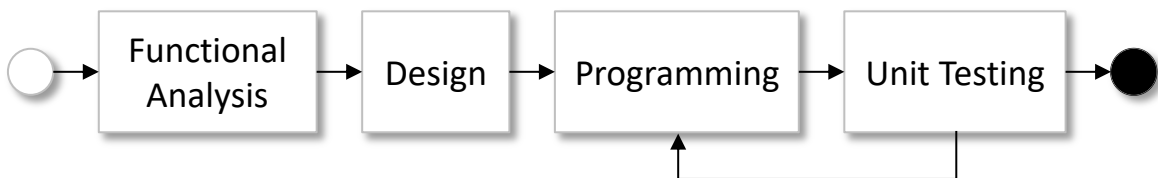
83

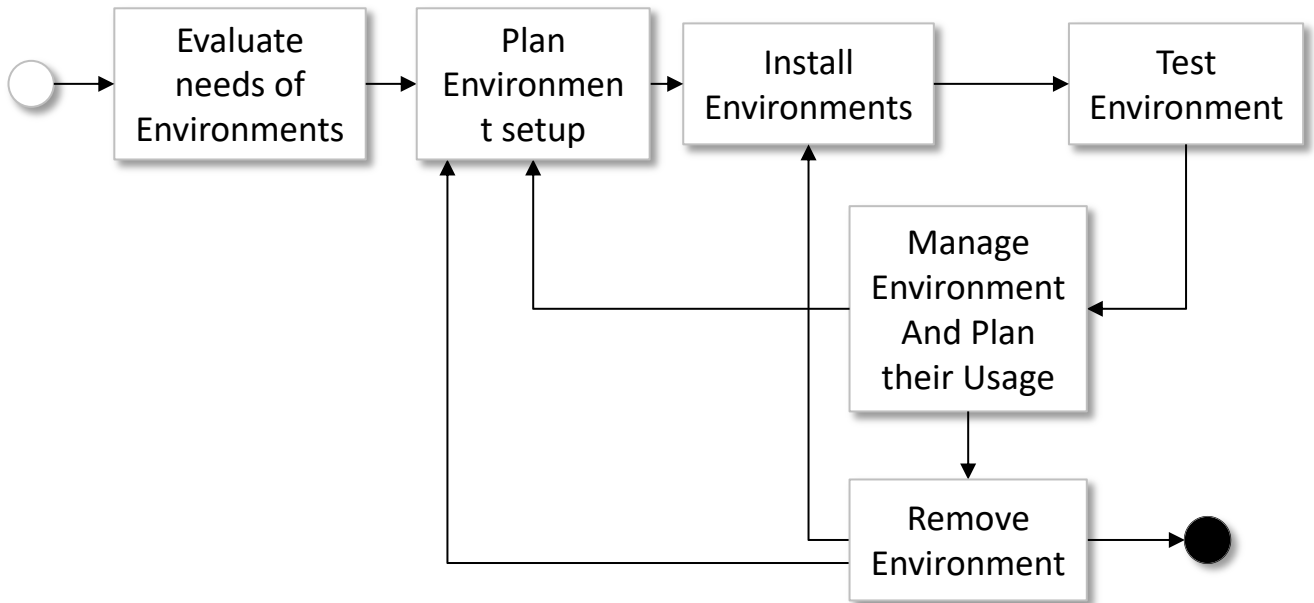
Process: Architecture / Software Development

ARCHITECTURE



SOFTWARE DEVELOPMENT





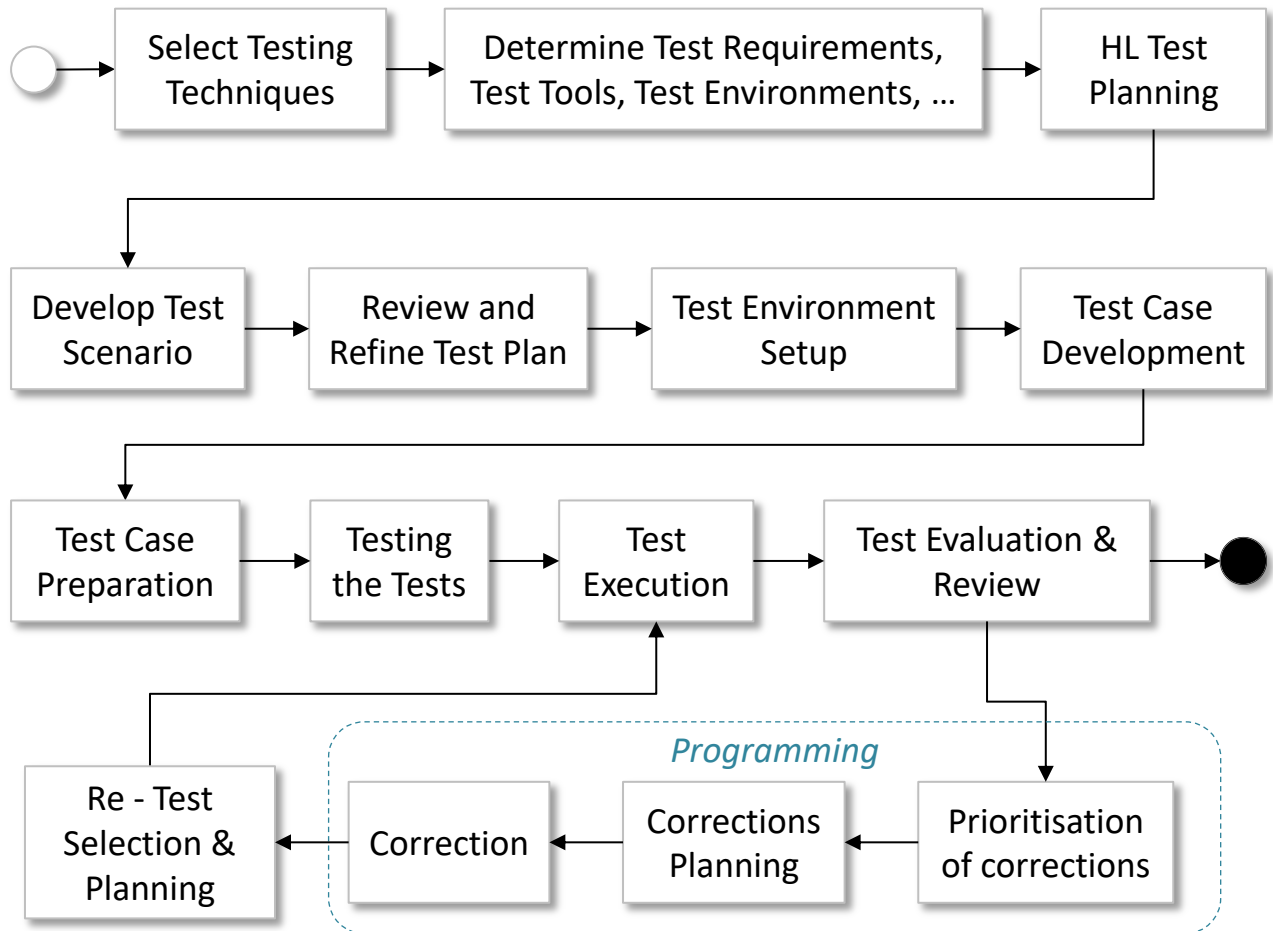
Different Types of Environments

- Development Environment
- Test Environment (examples: load tests, stress tests, .. Each requiring a specific environment)
- Training Environment
- Sandbox Environment
- UAT Environment
- Stage Environment
- Production Environment
- Disaster Recovery Platform

Some environments may be duplicated when dealing with different releases at a time.

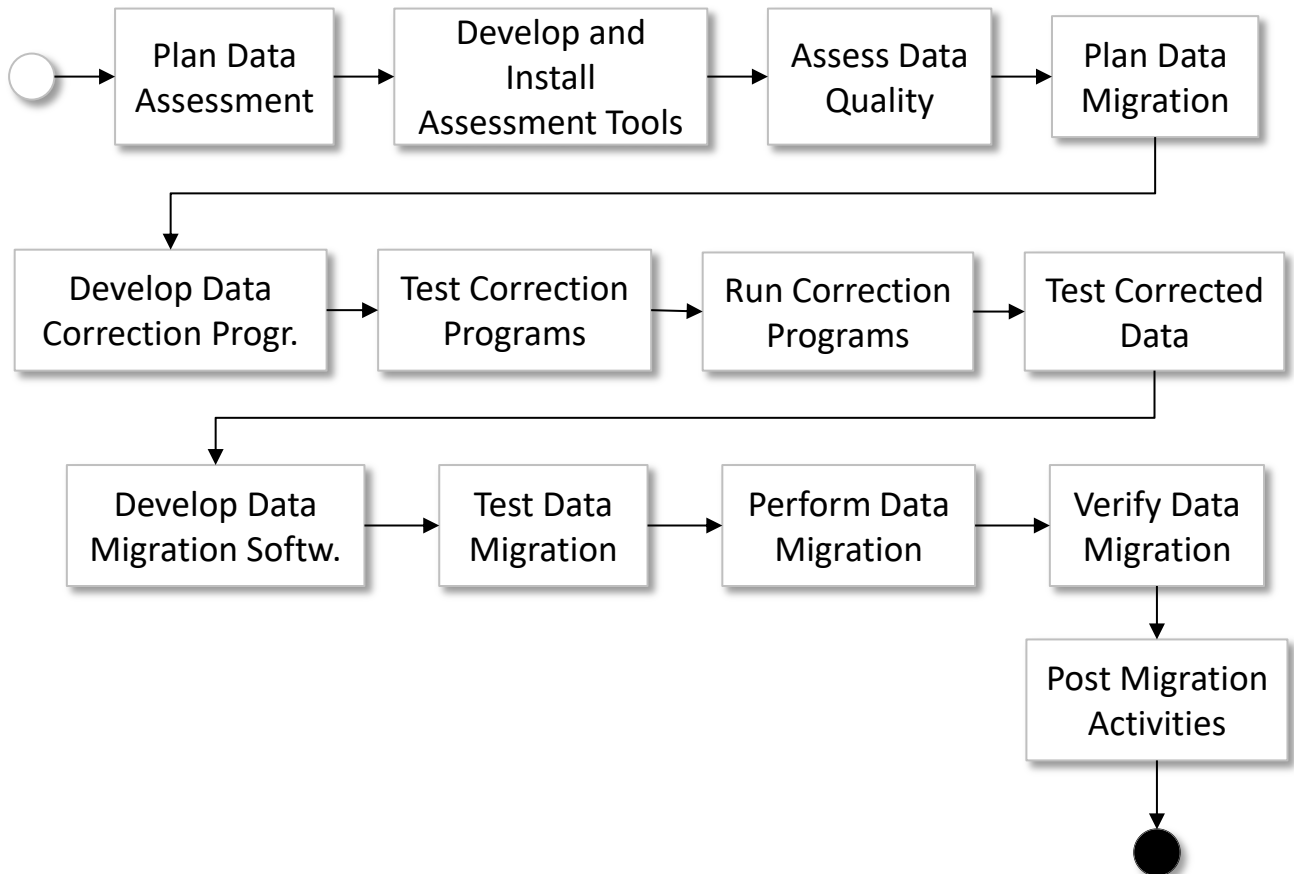
Process: Testing

Software Testing LifeCycle(STLC)



Process: Data Migration

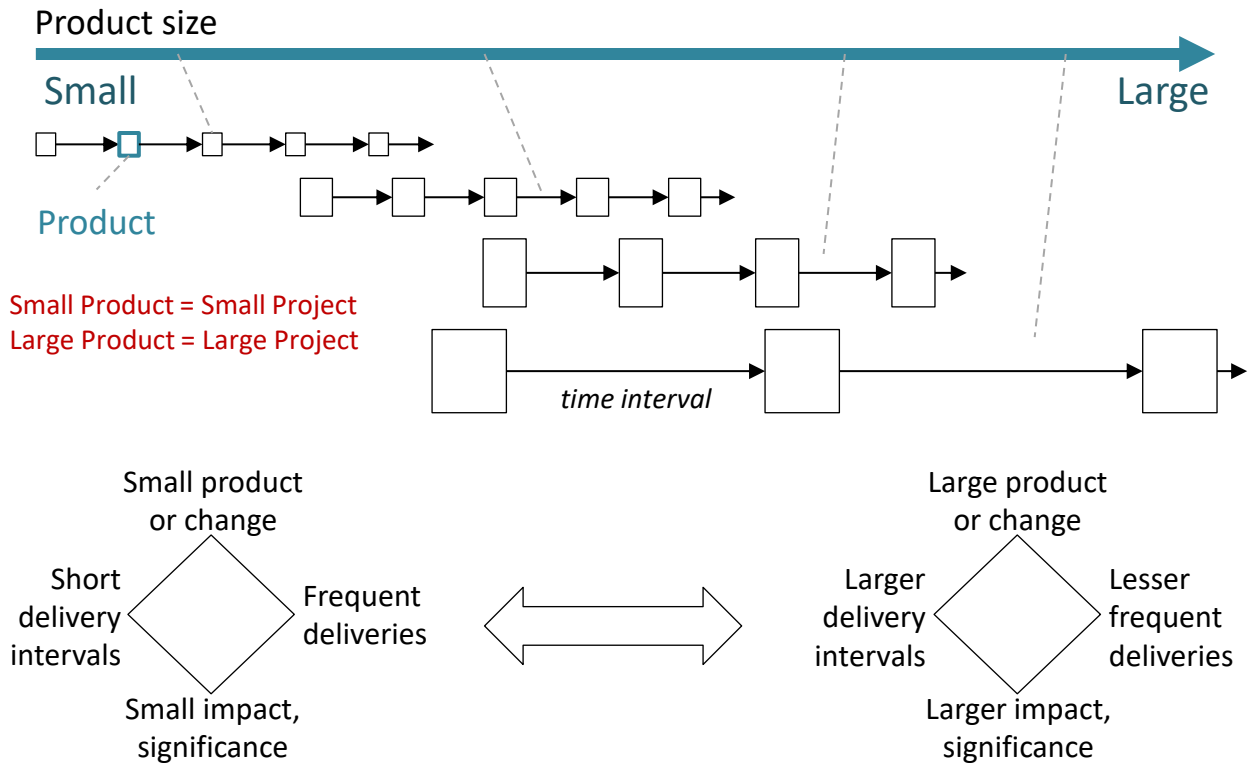
30/10/2018



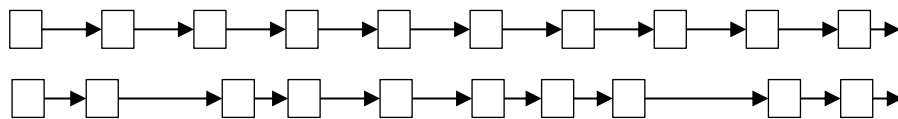
Product & Project Size

10/01/2020

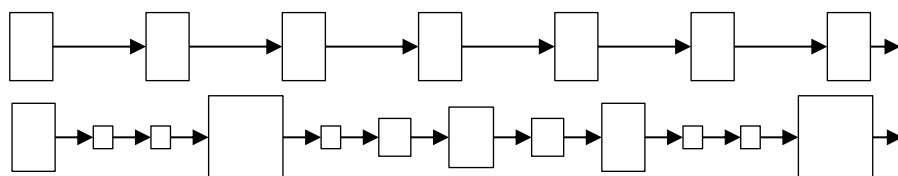
Size – Time – Deliveries - Significance



Regular / Irregular Intervals

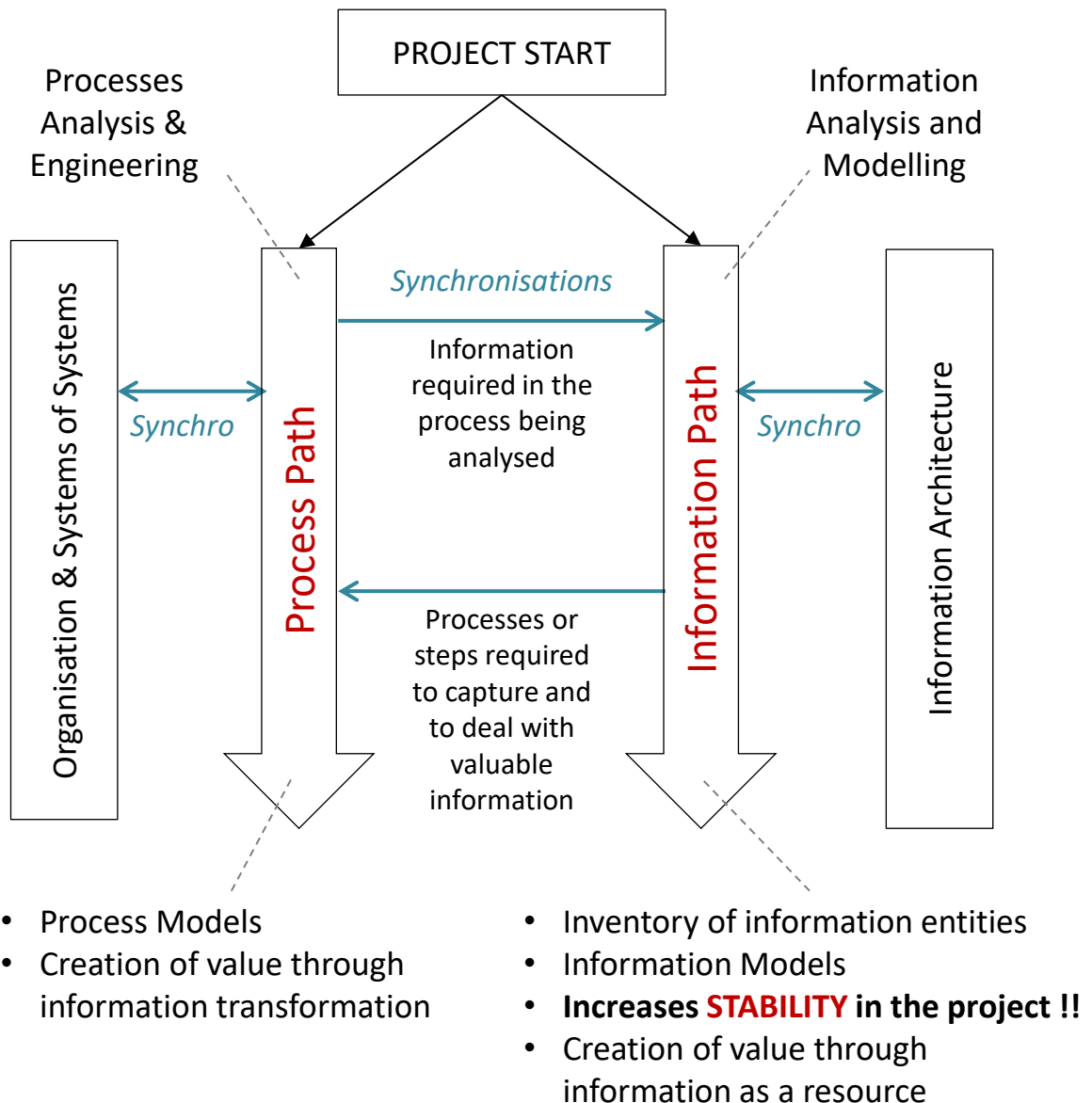


Similar Size – Variable Size



Typically this depends of

- Environment
- Product, solution, change
- Urgency of the product



Notes:

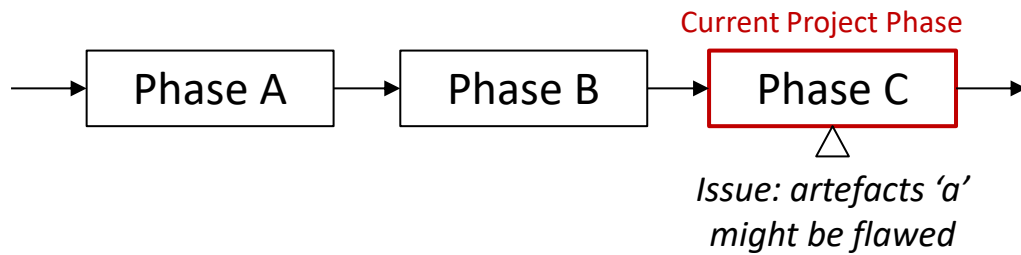
Information Path: Defining what information is required and stored. Modelling this information.

Process Path: Identifying the required processes and engineering them.

Synchronisation: Processes need information. The synchronisation makes sure this information is foreseen in the information models, databases and other information artefacts.

Information has value. Interesting information has to be captured and processed. Each identified information entity will thus have to be captured and transformed by processes.

Flawed Work in Previous Phase



Phase A: Mainly activities of type 'a', producing artefacts 'a'

Phase B: Mainly activities of type 'b' producing artefacts 'b'

Problem: Artefacts 'a' resulting from phase A are unclear. They don't take some aspects into account. There might be something wrong or incomplete.

- 1) Keep the project in phase C
- 2) Recheck the concerned problem area, rethink it
- 3) Adapt the artefacts 'a'
- 4) Check the broader impact of the adaptation
- 5) Validate artefacts 'a', with focus on changed and impacted area. Rest of the artefacts were unchanged. Review and validation should go swiftly.
- 6) Repeat for artefacts 'b'

Difficult? Strange? Developers do this all the time in waterfall-approaches

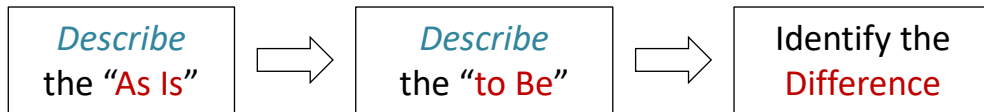
Bugs found during the test phase are corrected by programming activities without having to move the whole project back to "programming phase". Only the corrected software features are retested before the software code gets clearance to be released in production. Sometimes regression tests are required.

This same process can be applied to analysis, design, architecture, ...

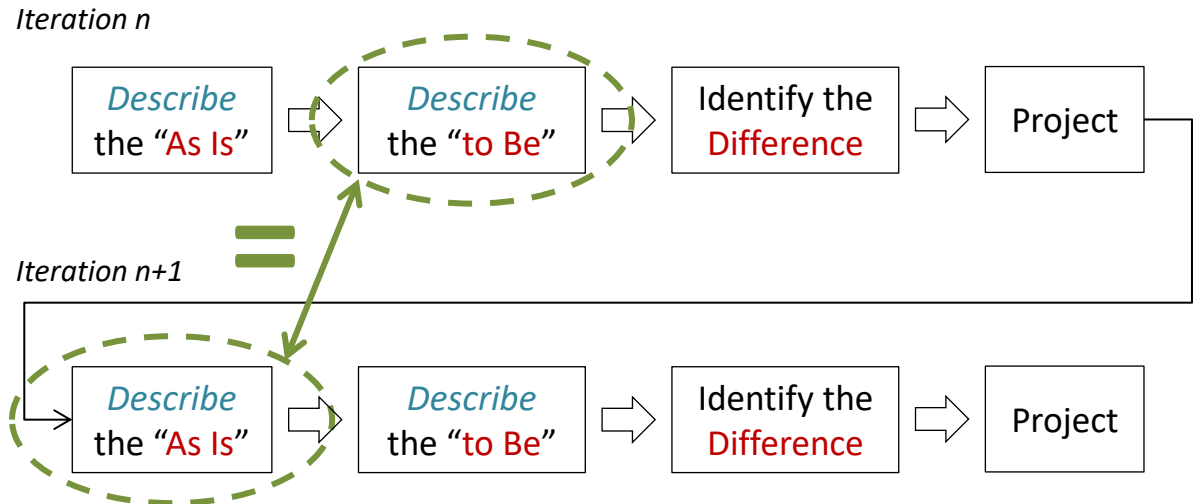
Sufficient verification and testing has to be implemented during the whole course of the project. Often, a flaw may already be detected in the next phase.

If a previous phase has been very badly executed, it might be necessary to redo it. This is a drastic decision.

Traditional Approach

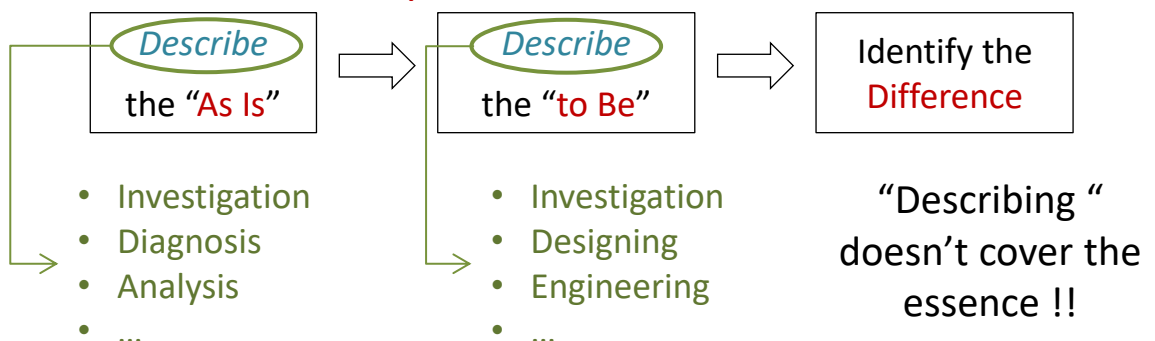


Issue 1 : To Be = As Is



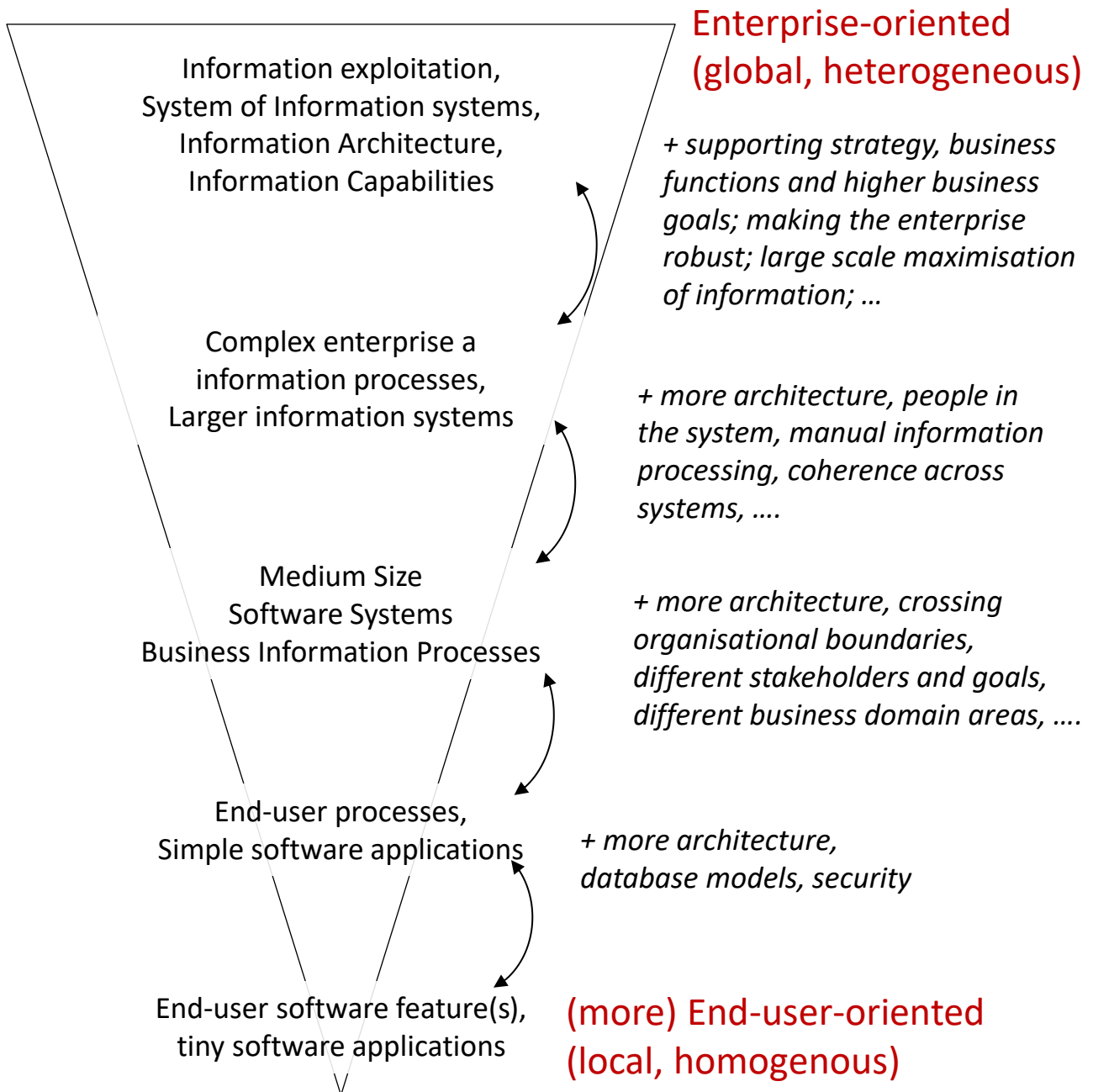
- Doing 2x the same job is a waste of time and resources.
- If former "to Be" is different from present "As Is", then there is a problem. What happened ?
- Use former Analysis and Design artefacts to think and conceive.

Issue 2 : Common Assumption: "Describe"



Limitations in Scalability of Methodologies

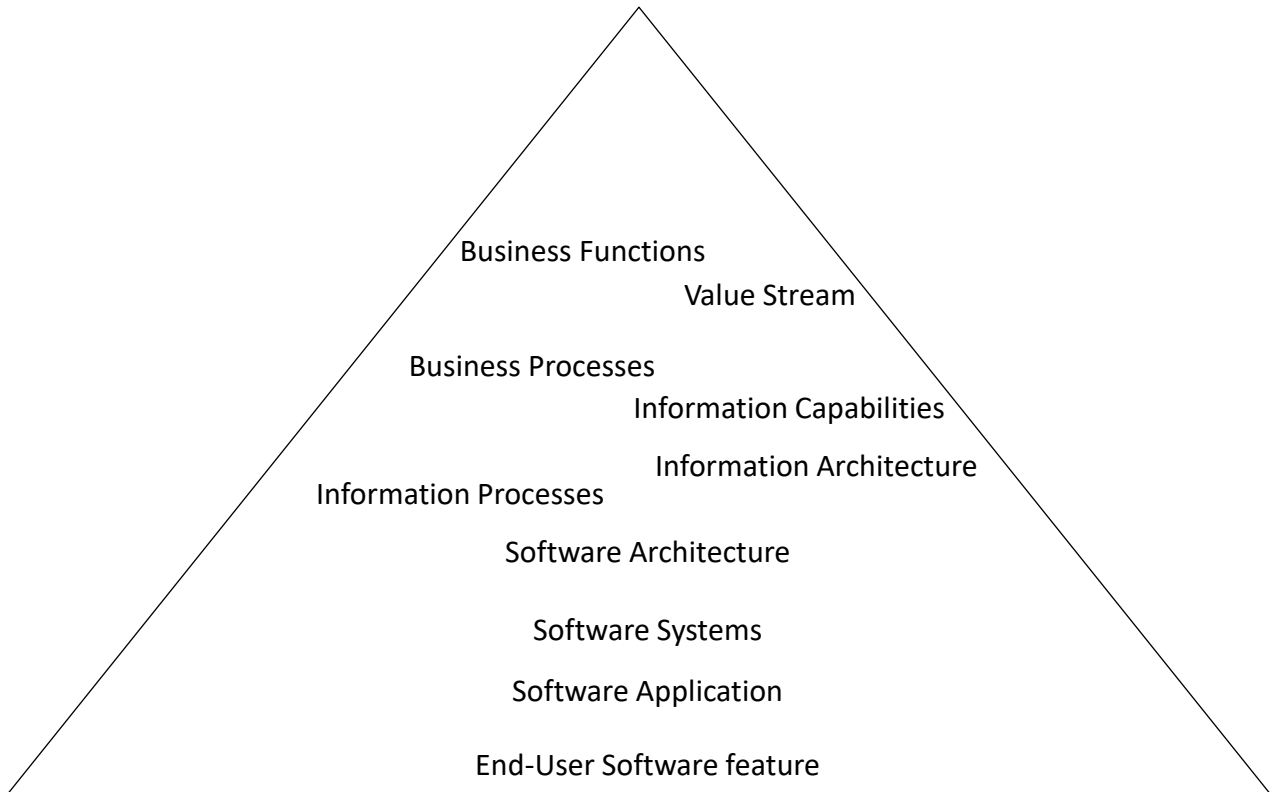
14/02/2020



The nature of the problems to be solved at each level is different. Hence, it requires a different approach, different methods and knowledge and even a different way of thinking. **!**

Limitations in Scalability of Methodologies

14/02/2020



Each level has its own level of thinking, own laws and principles, own knowledge area, own purpose, own goals.

Issues on a higher level can not be solved with the thinking of a lower level.

Notes:

This is similar to previous page. A different representation for clarification.

A few strategies and tactics

- Avoid first creating more chaos
- Avoid first to continue to allow bad input
- Identify the garbage and remove it
- Map what is known and gain understanding by further analysis and mapping
- Work follow-the-flow, top down or from fundament to what's built on top (bottom-up)
- Separate what's good and in order from what's bad. Protect what's good.
- Identify areas, clean up area by area (or system by system, process, by process, DB by DB, ...)

Avoiding creating chaos

- Put in controls
- Build upon truth and understanding of reality
- Think of consequences, impacts, ..
- Is the result an increase in order or complexity (hiding complexity also increases complexity) or increase of order and clarity.
- Consider a decision, an action, a behaviour, an exception, an issue, an event, a project, anything and imagine it to happen 1000x, or more. If order is maintained, then it is fine. If the expected result is a dysfunctioning, imbalance, waste or chaos, then it is a bad practice that shouldn't happen or happen with prudence and sparsely.

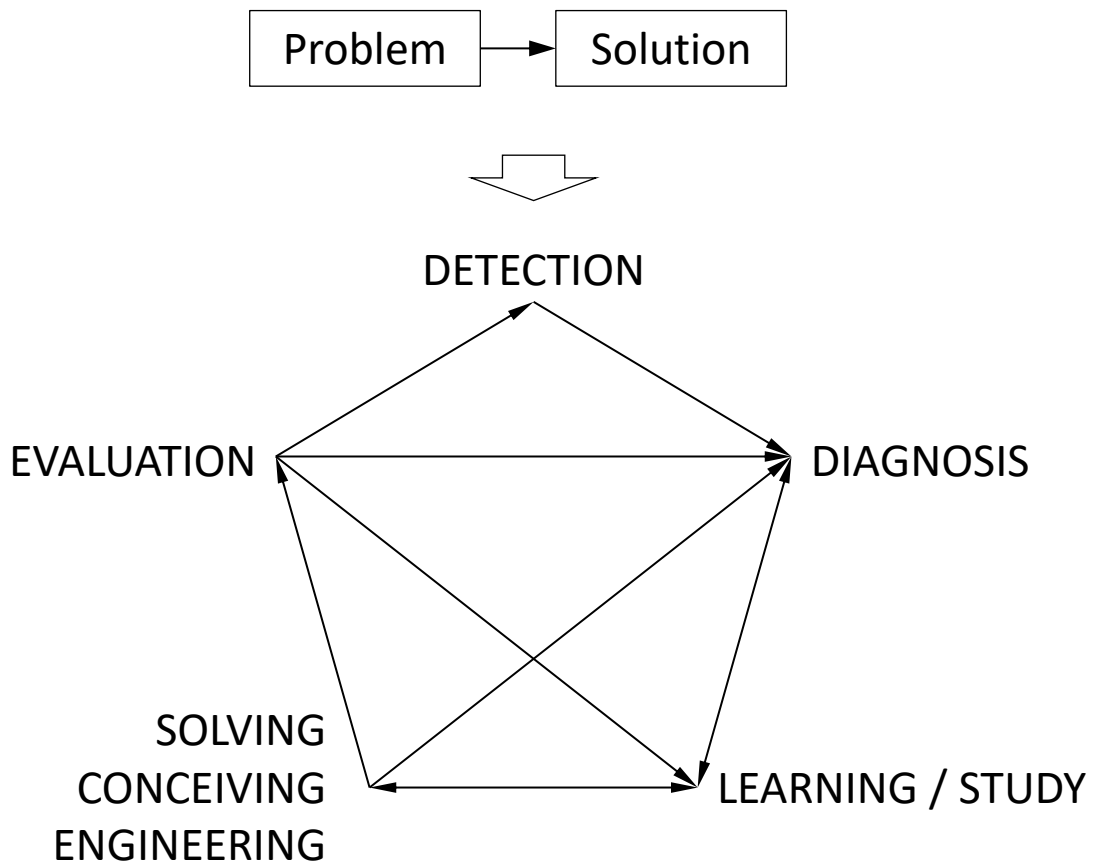


- ANALYSIS -



Key Elements of Analysis

30/10/2018



All five

- are **equally important**
- **must be performed by a professional Analyst**

A demand, requirements, an analysis or a solution outline not resulting from these activities performed by a professional Analyst, should be considered as unreliable.

Building upon it is building on quicksand.

- Learning the systems, processes, structures and their environment.
- Diagnosis: Identify gaps, problems, needs, obstacles, issues, opportunities
- Align demands, intentions, expectations with the NECESSITY

It is not about what people wants (“not mainly”, “not always”, ... ; What people want does not always solve the problem or contribute positively to the whole organisation).

Systems Analysis is not a discipline that aims to please people, but to conceive systems, structure, processes to solve problems.

The alignment of the expectations with the necessity is important in order to create motivation, aligned collaboration and to avoid disillusion.

- Align conflicting interests, demands, priorities, rules, ...
- Conceive a solution
- Get consensus about the solution

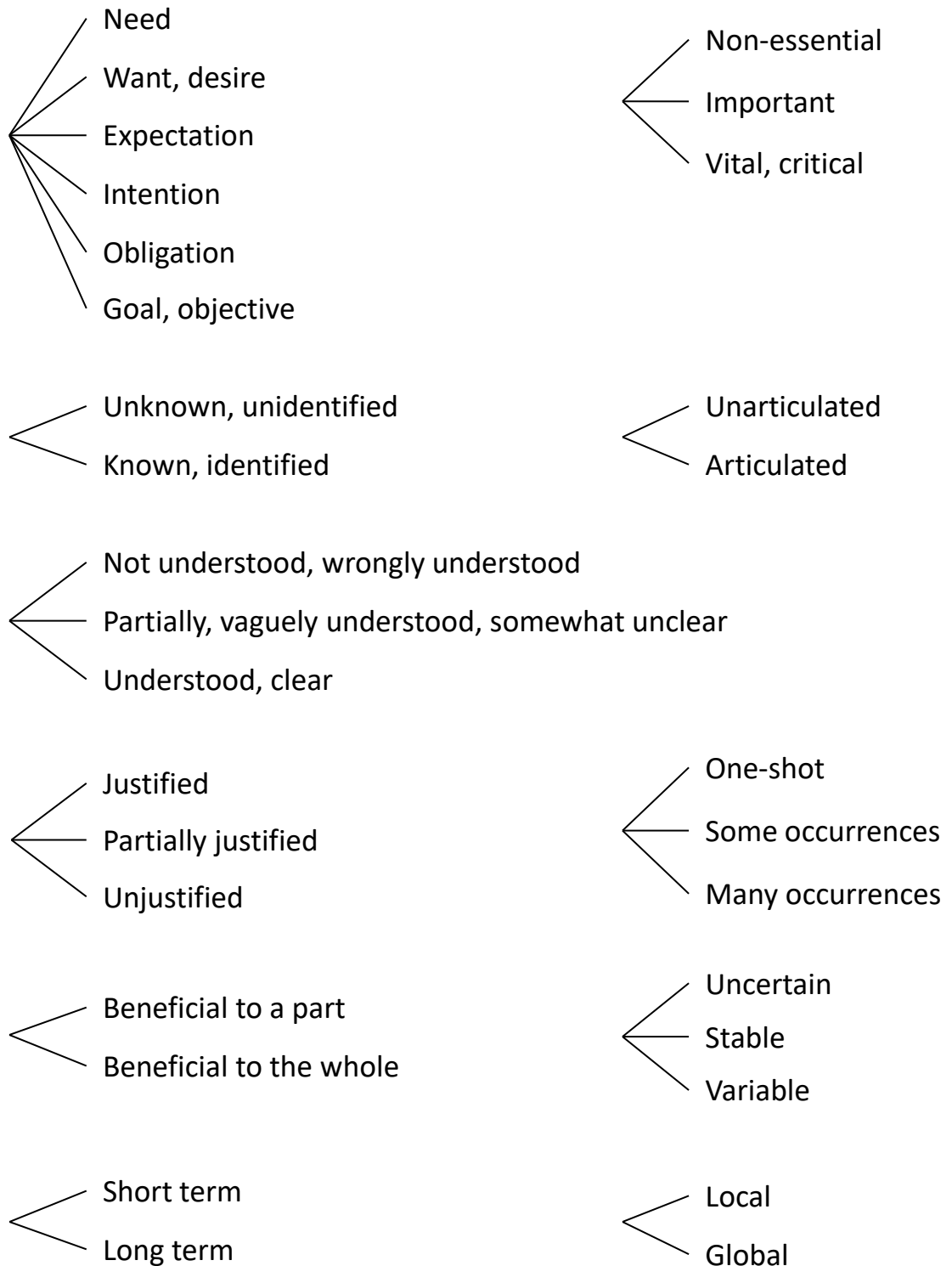
Analysis should be ideally) right and complete. Usually, a fairly right and fairly complete works also.

But an Analysis that aims to solve the wrong problem or which is insufficiently based on the reality, on the truth, like superficial insight, is likely to do more harm than good.

Needs Analysis

15/03/2019

Cause of the need, driver, links with other needs, goals, objectives



Needs Analysis

30/10/2018

NEED

SOLUTION

"I need a drill"

Drill

Drill

Why?

"I need to make a hole in the wall"

A hole

Anything that makes a hole

Why?

"I want to hang a painting on the wall"

Hanging an painting

Any mechanism that fixes a painting on the wall

Why?

"To make to room cosy in order to relax" To relax

Anything for relaxation

Why?

"I am stressed."

Not being stressed

Anything to decrease stress

What stresses you ?

Work? Relationship? Family? Conflicts? Problems? Uncertainty? ...

Not being stressed

Remove the real causes of the stress

Questioning can continue to get to the deeper cause

Wrong solutions: Drill, hole, painting, making a room cosy, relax

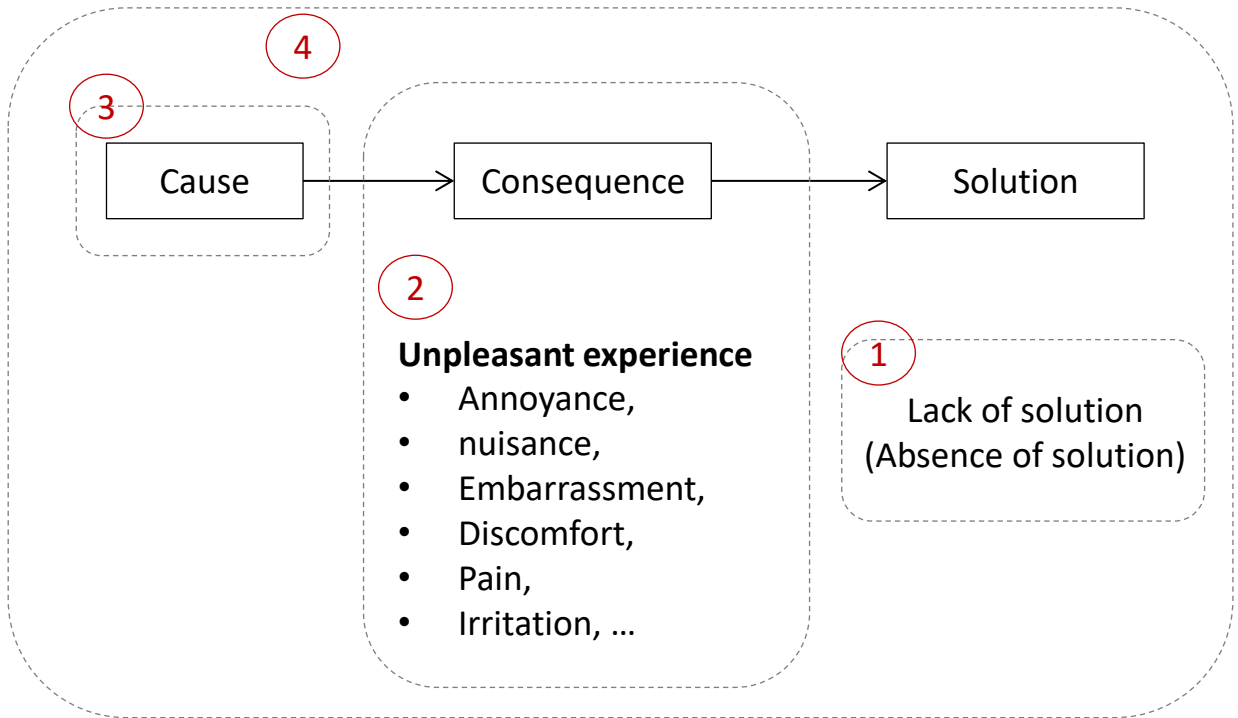
Bad diagnosis: Stress (=consequence)

Right diagnosis: Cause of stress

More 'needs' and solutions appear.

What is the Problem?

30/10/2018



Is a problem ...

- 1) the lack of something, the missing solution, ...
“we need a database”; “we need a system”; “we must have (object, product, solution>”
- 2) getting rid of the unpleasant experience, unpleasant feeling, irritation, inability, ...
- 3) eliminating the cause
- 4) the whole concept

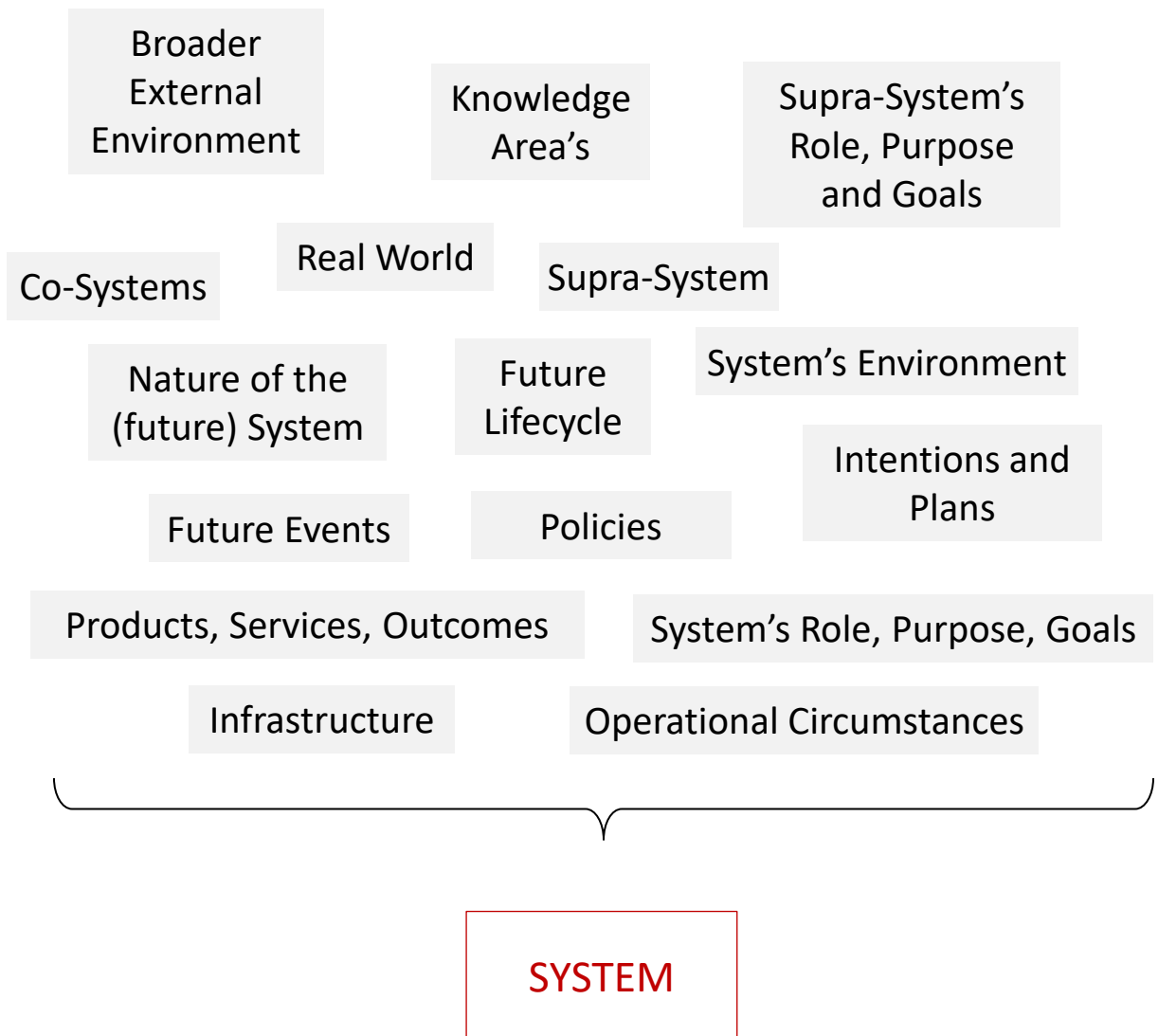
- What do the stakeholders ask?
- What do they want?
- What is the motive of the demand?
- What is their expectations?
- What is their demand about?
- What is required?

Are the answers related to 1 or 2

→ Need for more questions

Main Areas Determining a System

31/12/2020



Future Lifecycle and events concerns the lifecycle and events that are commonly or likely to occur, which can be expected and for which the system has to be able to deal with in the best possible or acceptable way.

The external environment may include other organisations and their systems, inter-organisational agreements and collaborations, law systems, etc.

This list might not be exhaustive. But it gives a pretty good idea about what to consider.

What to Analyse



Unknown needs and unarticulated needs

Personal insights

Personal Perspectives
Insights
Conflicting views
Belief systems
Assumptions

Articulates needs, partially articulated needs, known but not expressed needs, unjustified needs and expectations, ...

What the stakeholder, the customers or the end-users want

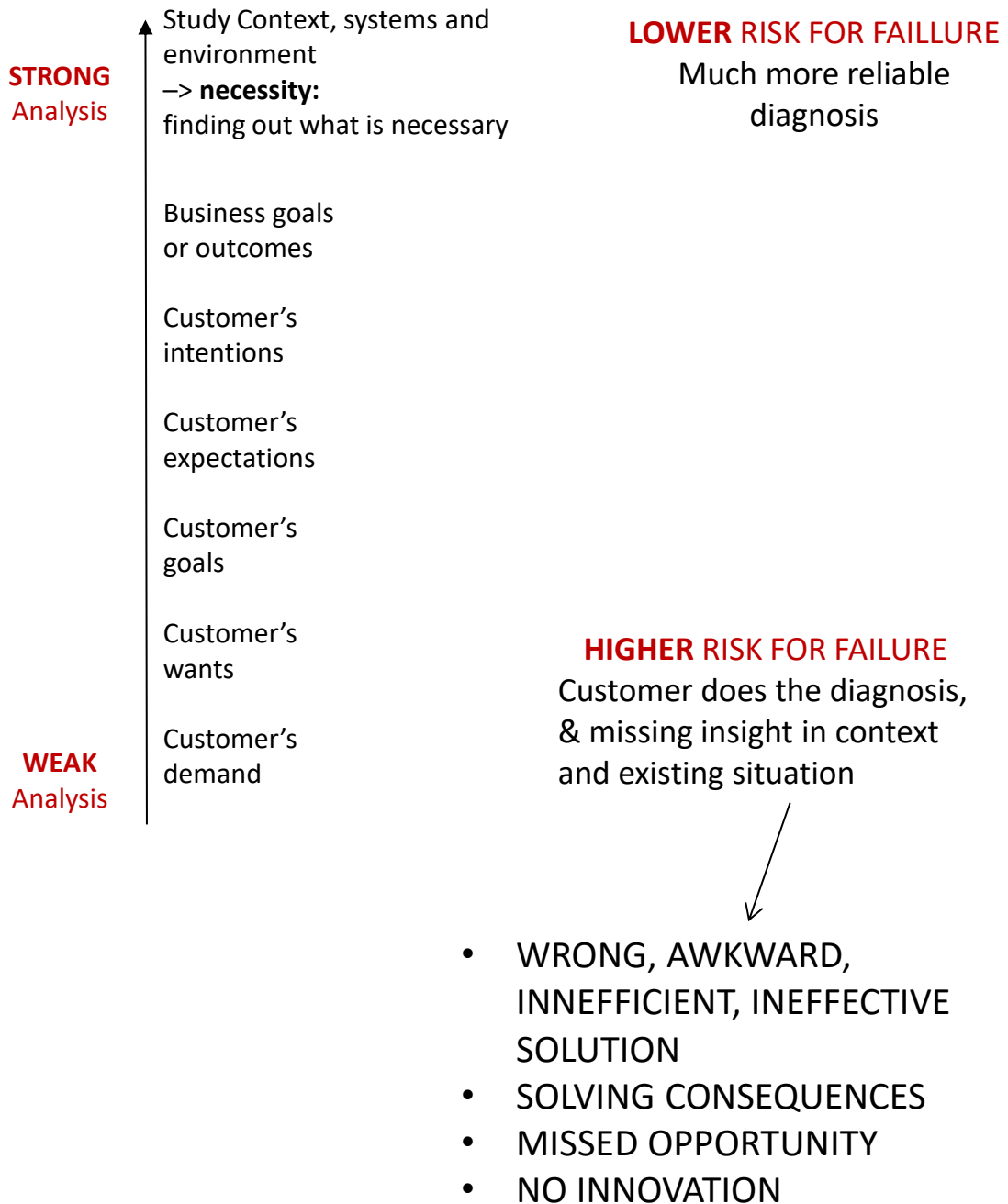
Opinions
Subjective diagnosis
Ideas, Suggestions
Disapprovals
Dislikes

Demand
Wants
Desires
Preferences

* : for concerned business domains, business stakeholders, ...

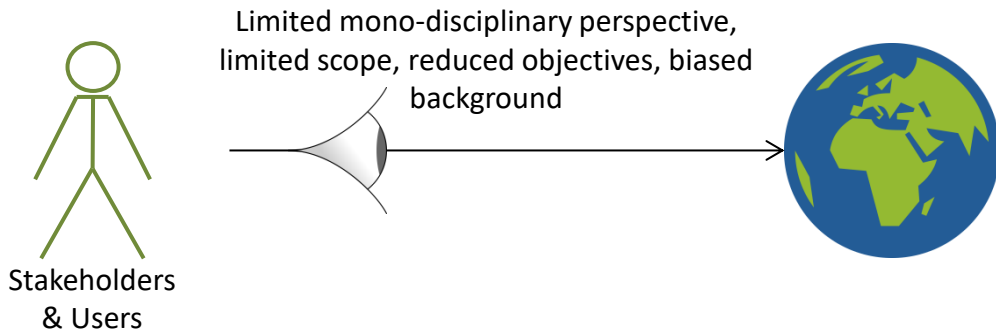
Weak & Strong Analysis

10/01/2019



Seeking to know “only what we have to know” to conceive a solution is a principle that doesn’t work, let alone being told what to implement.

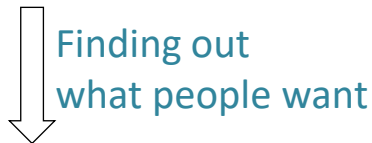
Wants vs Reality



Satisfying People



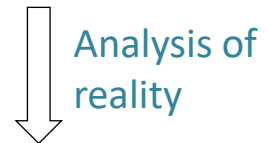
Solving Real Problems



- Pain points, annoyances
- Wants & Desires
- Preferences
- Choices
- Dislikes
- Demands
- Expectations

Trying to please others; being dependent of other people's thinking, desires, expectations, , ...

Subjective diagnostic: Subjective, emotionally influenced, from single perspective, distorted, partial, vague, biased by belief system (eg. View on IT)



- What the situation, the reality, requires
- What is possible & impossible

Thinking critically
Being autonomous
Taking responsibility
Doing what you decided is necessary
Suits the situation
More objective diagnostic & view
Performed by competent analyst

**Serving Clients REALLY
by FINDING OUT what is NECESSARY and
solving the REAL Problems with REAL Solutions**

Notes:

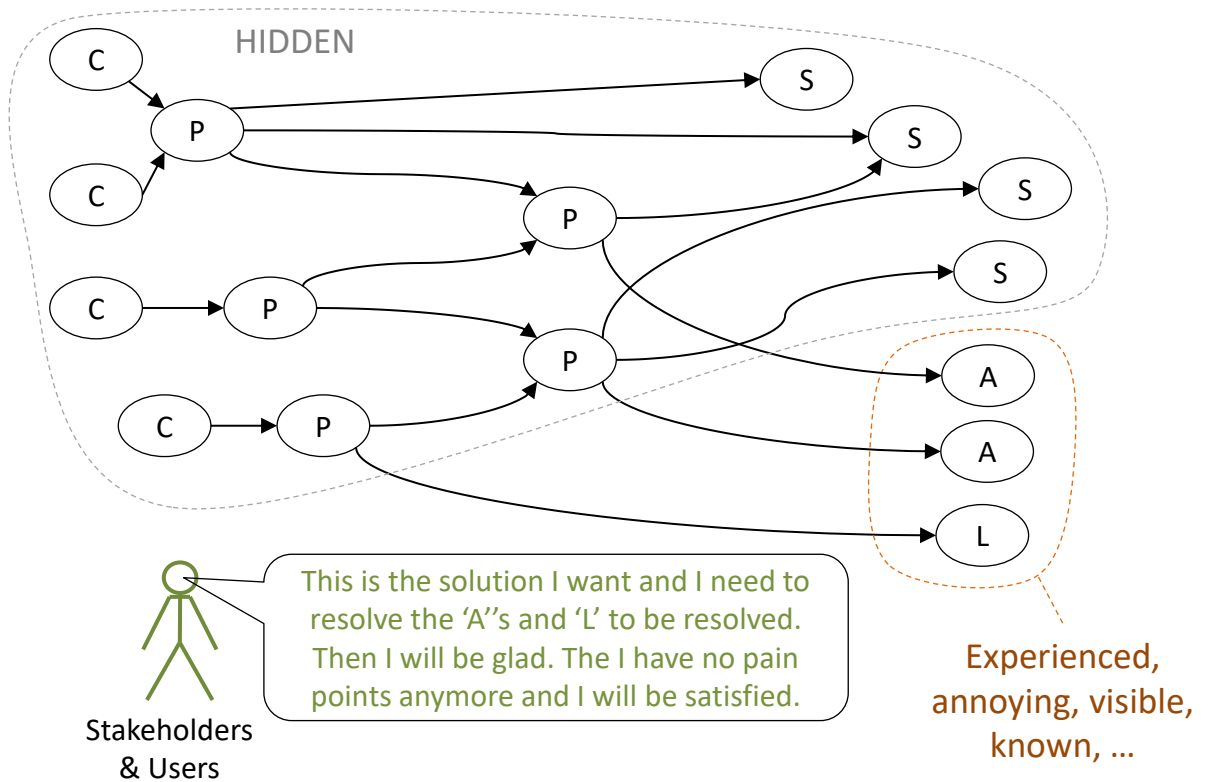
Doing what is necessary may, in the end, suite and satisfy people much more. Doing what is necessary, does not mean people choices should be ignored. It's more a matter of having different priorities.

Wants vs Reality

30/04/2019

(P) : Problem (underlying / hidden / root)
(C) : Circumstance(s)
(S) : Symptom or consequence

(A) : Annoyance (experienced consequence)
(L) : Limitation, obstacle



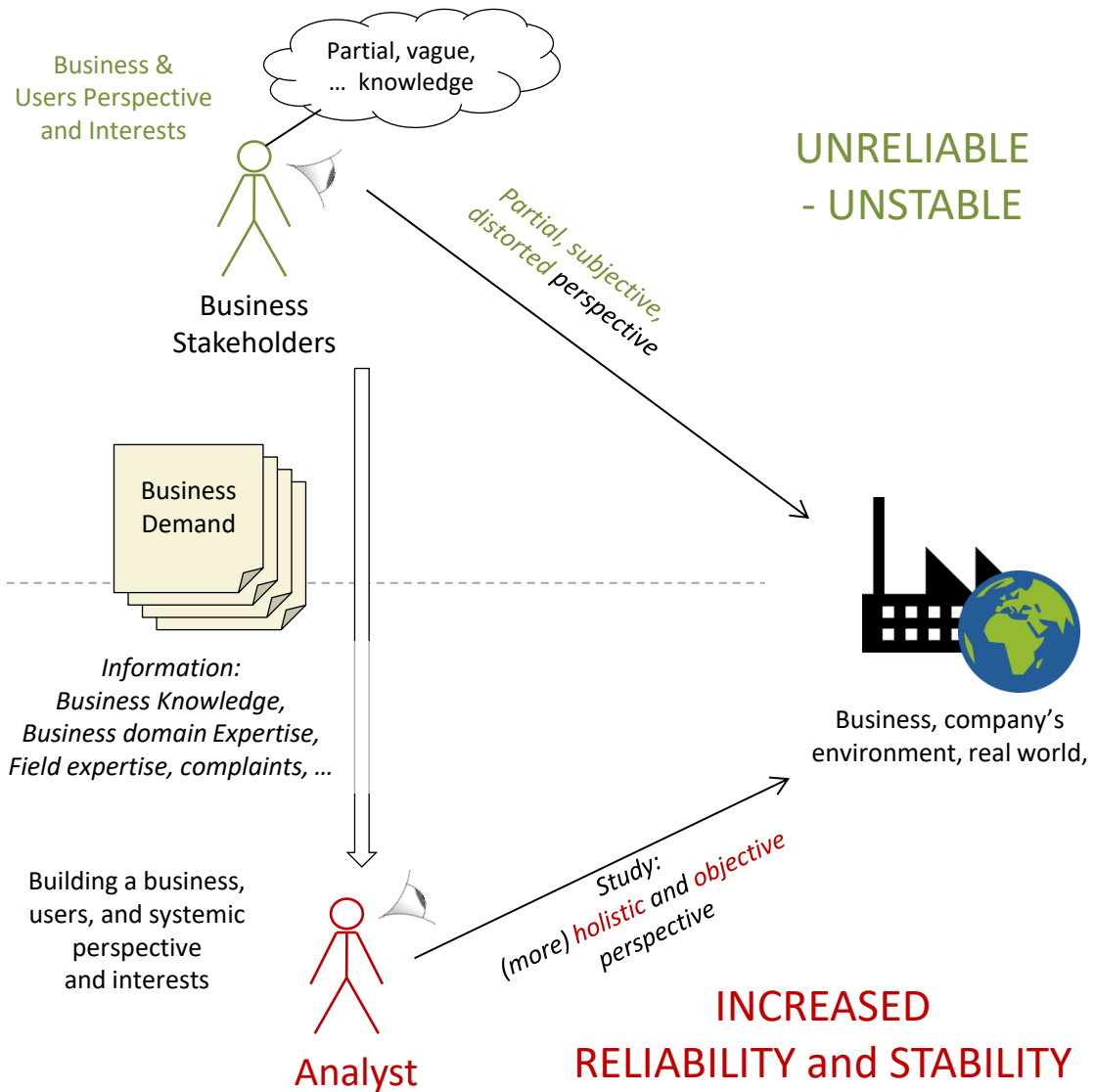
- From a **people's perspective** : pain points will be eliminated.
- From a **systemic perspective** : Problems still exists.

If you were the client

- Would you be satisfied if you got the solution you wanted, but it doesn't solve the annoyances and limitations or it creates new ones?
- What if the (P) continue to create negative consequences?
- Would you choose to get a solution to the annoyances and limitations or would you prefer to see all the problems to be resolved?

In the end, people prefer all problems to be resolved, even the one's they don't know of rather than having their solution and still having problems.

Reflect: What if doctors would treat only based on the patient's diagnosis, what he or she asks? What if they would treat only symptoms?



REALITY MATTERS

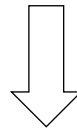
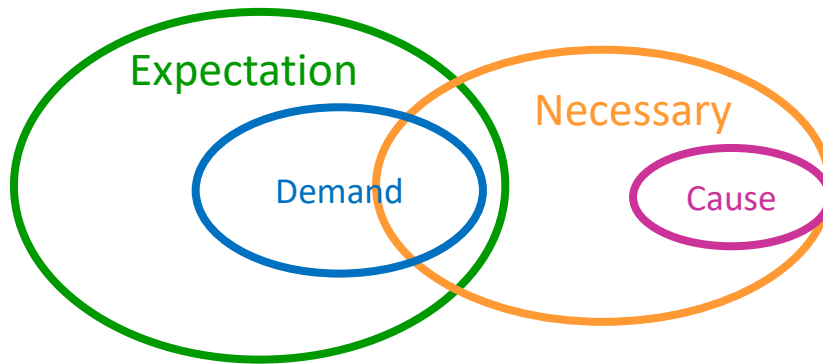


What has to be changed in reality and will truly solve the problem/needs and meet the company's systemic requirements:

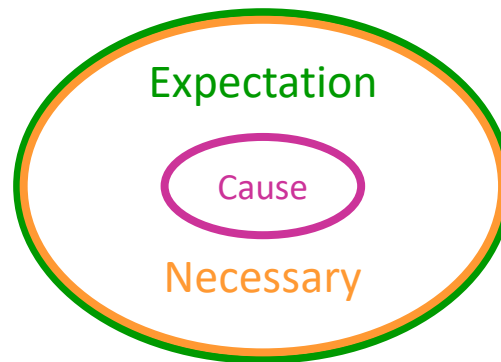


Human – Reality Alignment

30/10/2018



Analysis aligns cause, necessity and expectation

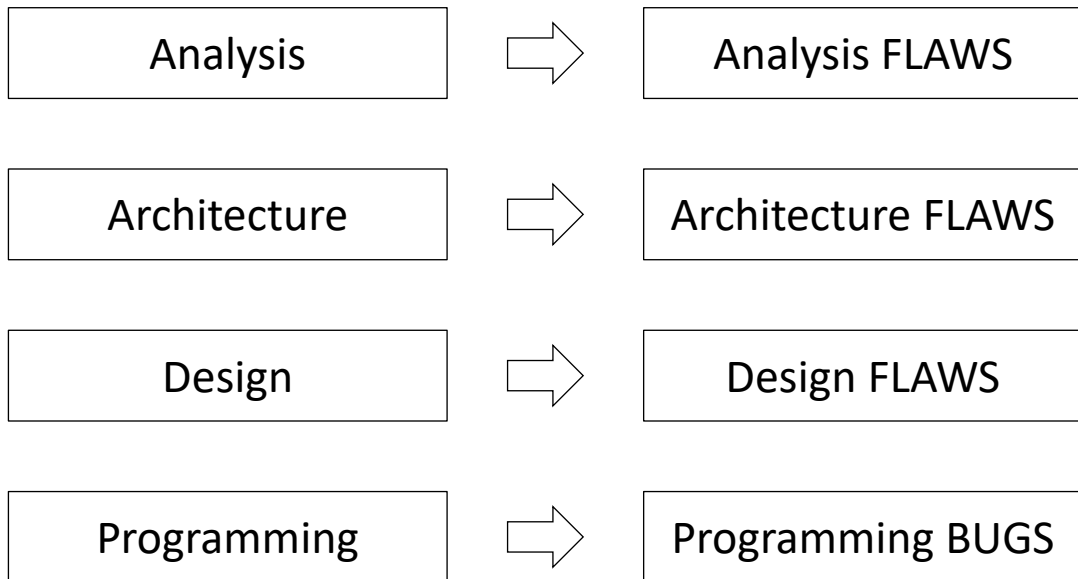


Causes of different views and different expectations

- Different knowledge
- Partial understanding of the whole problem or solution, its obstacles, its constraints and implications
- Different perspectives
- Different goals and priorities
- Different solutions in mind

Four possible solutions (ordered from best to worse)

1. Listening to the explanations and arguments to create a final global insight – merging the knowledge and perspectives
2. Finding a solution that meets all different perspectives, interests, goals and priorities
3. Making a compromise among the different parties to come to a final solution, hopefully the best solution
4. Choosing the solution of the most authoritarian party



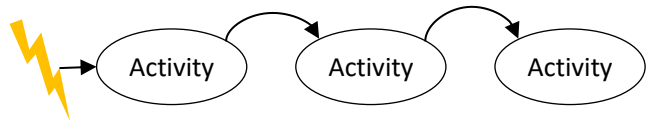
We know how to test/detect programming bugs.

But how to detect (on time)

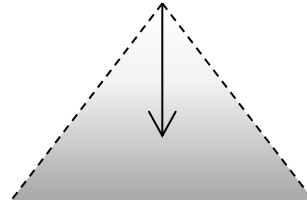
- Analysis flaws?
- Architecture flaws?
- Design flaws?

If we don't detect them, and handle them as simple 'changes', how can we improve?

1. Follow the Flow - Chronological



2. Top Down

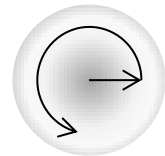
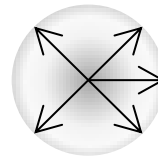


3. Begin with the end in mind - Goal oriented



4. Radial Expansion

Starting at a point (centre, core) then moving towards the periphery



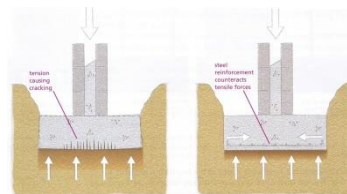
5. Build based on / around stable aspects

Starting with clear, definitive, stable elements



6. Foundation First. Then building on top.

Atomic elements, most basic aspects



Note:

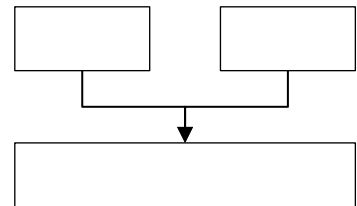
“Top down” means to look first at the broader picture, a drone-view. “Top” does not point towards the top organisational level (management).

7. Greenfield Approach or Reengineering

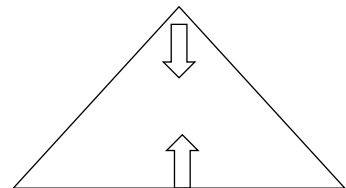
8. Approach for Adaptation of Existing System(s)

9. Aspect-based

Study specific aspects, domains, perspectives one by one. Then putting it all together when conceiving the solution. Example: process perspective and information perspective of a software application.

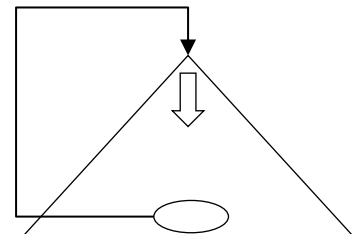


10. Mixed Top-down and Bottom Up

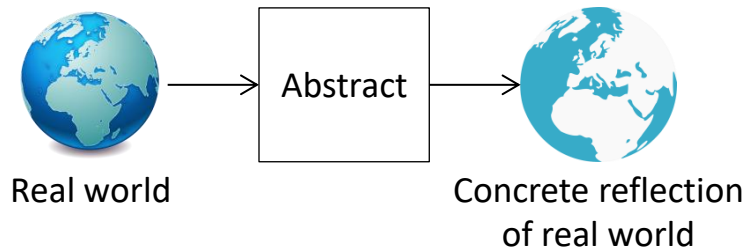


11. Initiate Bottom Up and Go Top-down

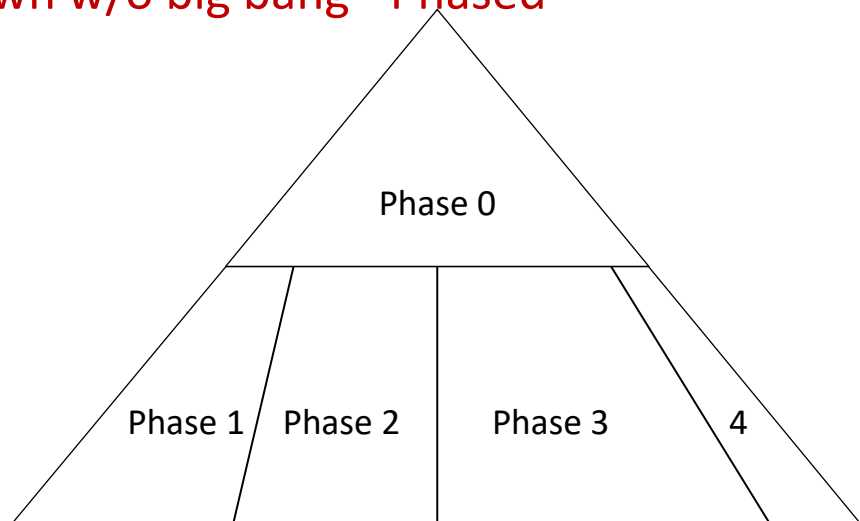
Detecting, identifying, study, ... something at the bottom. Bring the subject to the top. Work with it downwards by taking this new subject, intentions, plans, insight, ... into account. While the detection and start is below, the top-down process facilitates the integration and alignment.



12. Concrete to Abstract



13. Top-down w/o big bang - Phased



Phase 0 provide an overall framework, organisation (and architecture). It shapes the whole initiative.

Then (sub-)projects can be launched one after another as phase 2, 3 and so on. These phases can overlap. They can be executed in parallel or they can be delayed if necessary. This offer a greater flexibility.

14. Follow the Structures

Take an existing structure. Define parts of it. Treat part by part.
Example: cleaning up a house room by room

Notes:

An approach for **chaotic and undocumented environments** can be to make first inventories. Get rid of the obvious waste. Then compartmentalise accordingly to previous approaches. Prioritise them and then clean up one compartment after another.

Sub-Optimal TACTICS (from a logical perspective)

1. Risky Parts First

A project can fail on a difficult and risky part. This part is dealt with early in the project. If it fails, the project can be stopped or an alternative solution can be searched. If this difficulty is postponed until later in the project and it fails, then time and resources have been wasted and the lost investment will be larger.

2. Quick Wins First

Quick wins show usefulness and may create buy-in and motivation. The quick wins may correspond to simple issues that can easily be solved (see easy parts first). Or, it may be motivated by short term results orientation rather than by a system/long term ambition.

3. Following Established Priorities

Priorities may be defined by using different criteria and with different motives. How have these priorities been established? For what motives? Who defines the process? Based on what competencies? Or what aspects have to be taken into account and which have been underestimated or neglected?

4. Easy Parts First

Postponing the more difficult part can be driven by fear of complexity. This is not a good sign.

5. Known Parts First

Postponing the lesser well-known parts can be driven by fear of the unknown. Risks and true challenge may lay in the unknown (or lesser known) area's.

'sub-optimal' doesn't mean 'bad'. It means that the risk is higher to have a lesser good solution from the perspective of concept, logic, effectiveness, ...
For practical reasons, they might be the best choice. One needs to understand the risks and drawbacks of each.

Sub-Optimal TACTICS

6. Bottom - Up

...

7. Follow the Boss, the Demand, the Sponsor, ...

...

8. Smallest First

...

9. Highest Business Value First

...

10. Organic expansion based on Usage and User's perspective

1. Need-driven
2. Problem-driven
3. Short-term-goal-driven
4. Result-driven
5. Demand-driven
6. Priority-driven (based on usage)
7. Opportunity-driven

WORST

Reactive, symptomatic and organic growth guided by persons having only superficial knowledge of information, systems, information /software systems and systems engineering (engineering of systems).

Analysts investigate and think about the environment, system, problem from different perspectives:



Objectives - Plans - Intentions



Causes - Consequences



Opportunities



Priorities – Importance - Criticality



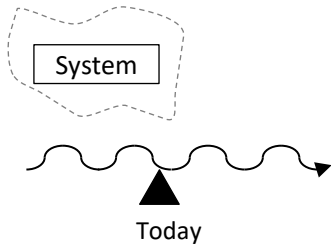
Value
(not just stakeholder's appreciation)



Stakeholders
(their domain, objectives, ...)



Necessity – Preferences & Desires

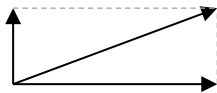


System and System Environment

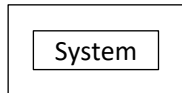
Evolution (past, present, future)



Short and Long Term



Forces - Tendencies



Hierarchy of Systems



Whole - Parts

Capability and Attributes



Different levels of details



Structure & Mechanisms
(behaviours, dynamics, processes, ...)

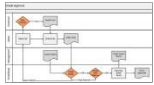
True Nature of Things



Interactions - Influences



Flows of Energy, Information,
Matter, Money



Abstract - Concrete



Lifecycles of systems, information, products, services, documents, ...



Mainstream - Exceptions



Balance & Harmony



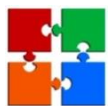
Effectivity - Efficiency



Flexibility - Evolvability



Usage – Maintenance – Manageability



Integration

(all levels, aspects and perspectives)



Risks & Security



Various other perspectives: Company culture, values, vision, organisation, policies, geographical locations, innovation, progress, speed, strengths and weaknesses, viability, ...



Cost

Three Key Perspectives of Thinking

15/03/2019

Thinking in terms of ...

1) A WORLD OF SYSTEMS

- Supra-systems, systems of systems, systems and sub-systems,
- Systemology, Systems Thinking, Systems Design, Systems Behaviour, ...

2) MULTI-DIMENSIONAL THINKING

Purpose, role, meaning, intentions, reasons, context, implications, knowledge areas, skills, perspectives, subjective versus objectivity, usefulness, location, focus, alternative usages, appropriateness, similarities, commonalities, differences, cohesiveness, coupling, characteristics, parts – whole - greater whole, object versus environment, structures, systems, processes, lifecycles, flows, concrete and abstractions, mechanisms, balance, harmony, action – reaction, nature of things, cause-effect, different levels of detail, evolution, small amounts versus larger amounts (numbers, repetitions,...), what-if's, strengths and weaknesses, opportunities and limitations, obstacles, importance, criticality, options, generalisation versus specific case, mainstream and exceptions, forces, interactions, influences, relations, certainty (probability), precision versus vagueness, approximation, variability, classification, priorities, order, cost, risk, value, timing, circumstances, and so on. (see previous slides)

All these aspects can be applied iteratively.

3) INFORMATION EXPLOITATION

- Presence, Availability
 - Creation, capturing, gathering, dissemination
- Value and Quality
 - Includes relations (connections, links, ...)
- Processability
 - Format, structure, organisation
- Usage, Innovation

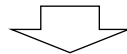
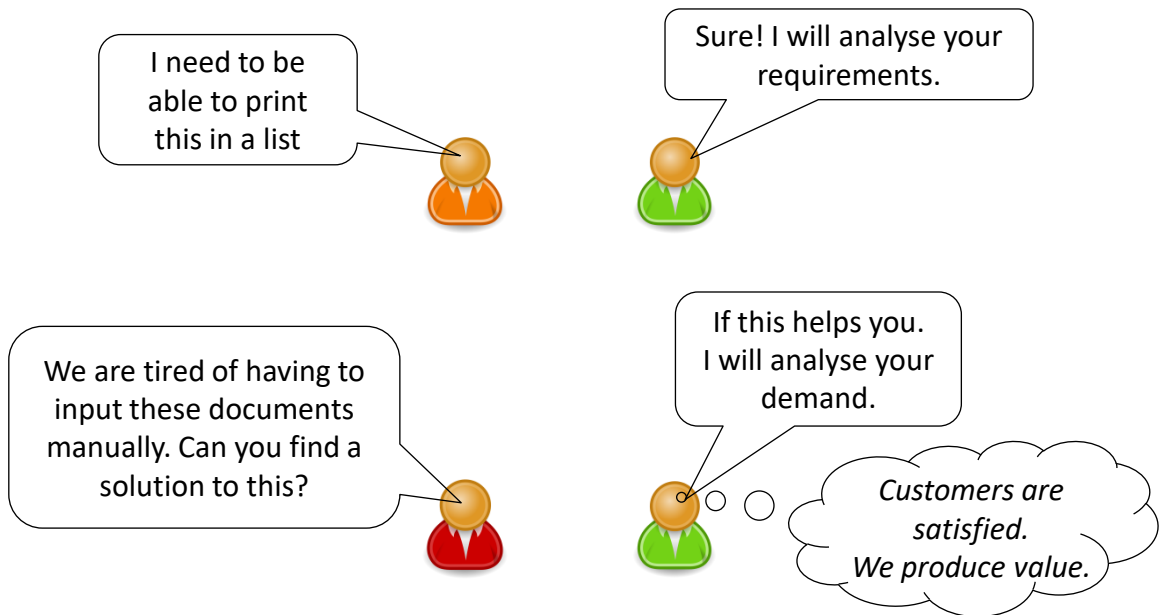
- Business Model
- Business Canvas
- Business Strategy Model
- Five-Forces Model
- Business Domain Model
- Value Network
- Value Chain Model
- Value Stream
- (System) Context diagram
- Organisation chart
- Goal Model
- Stakeholder Map
- Product / Feature Roadmap
- Business Function Model
- Business Activity Model
- Business Scenario Model
- Business Role Definition
- Business Function/Process Models:
 - Process Model / Swim-lane
 - Role Activity Diagram
 - Data Flow Diagram
 - Process Flow Diagram
 - Workflow diagram
 - Use Case Model
 - Flow Chart
 - Function Hierarchy
- Functional Dependency Diagram
- Sequence Diagram
- Business Event Model
- Interaction Diagram
- Timing Diagram
- State Transition Diagram
- Information/Data/Database Models:
 - Information Architecture
 - Logical Data model
 - Conceptual Data Model
 - Entity Relation Diagram
 - Physical Data Model
 - Class Model
 - Business Object Model
- Implementation Diagram
- Component Diagram
- Composite Structure Diagram
- Deployment Diagram
- Physical Model
- System Architecture Diagram

Assumption:

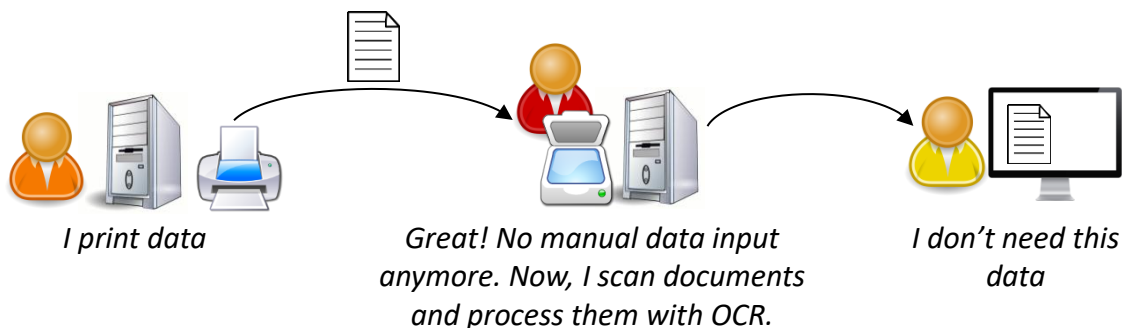
THE DEMAND DESCRIBES THE PROBLEM

AND/OR THE REQUIRED SOLUTION

You ask, I deliver – Your wish is my command
Giving Users What They Want

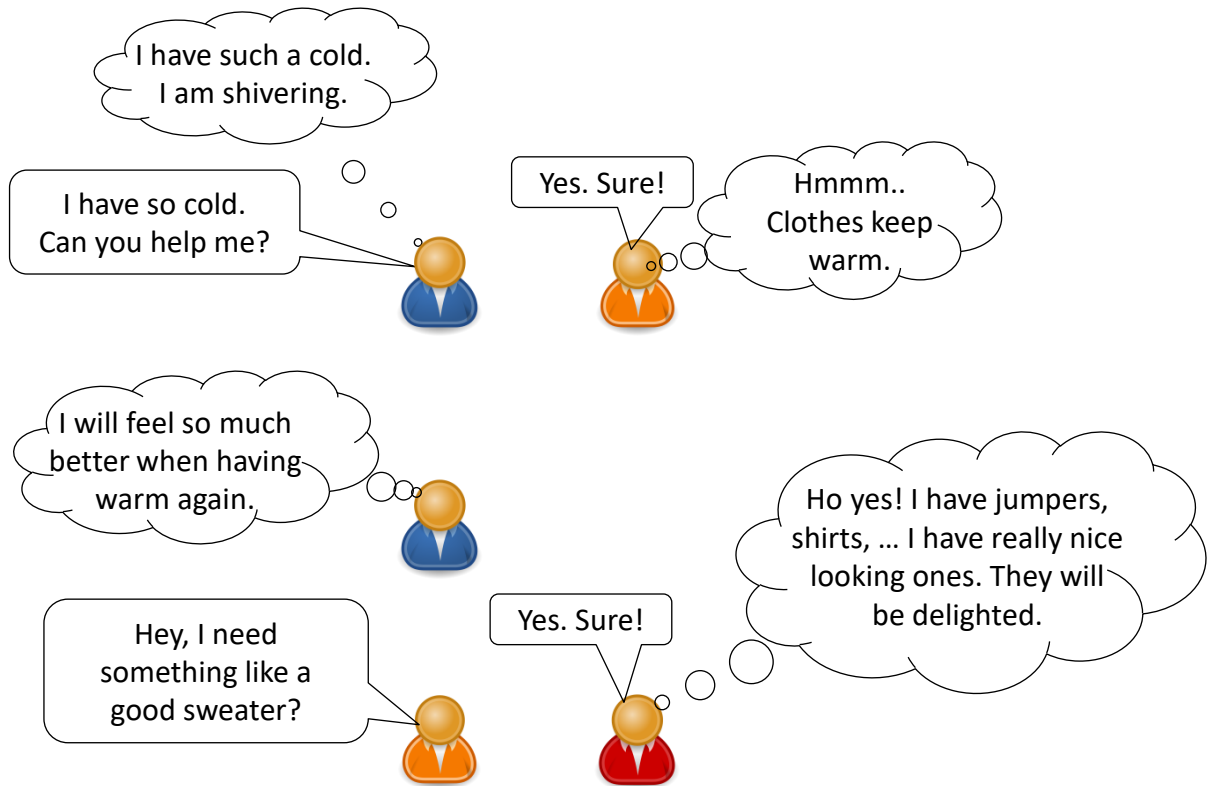


Improved Process



Traps:

- Missing the detection, diagnosis, learning, evaluation
- Trusting the demand, confusing the demand or complain with the diagnosis; assuming a diagnosis has been done.
- Limited scope, local thinking
- No analysis is done, only a refinement, translation of the demand



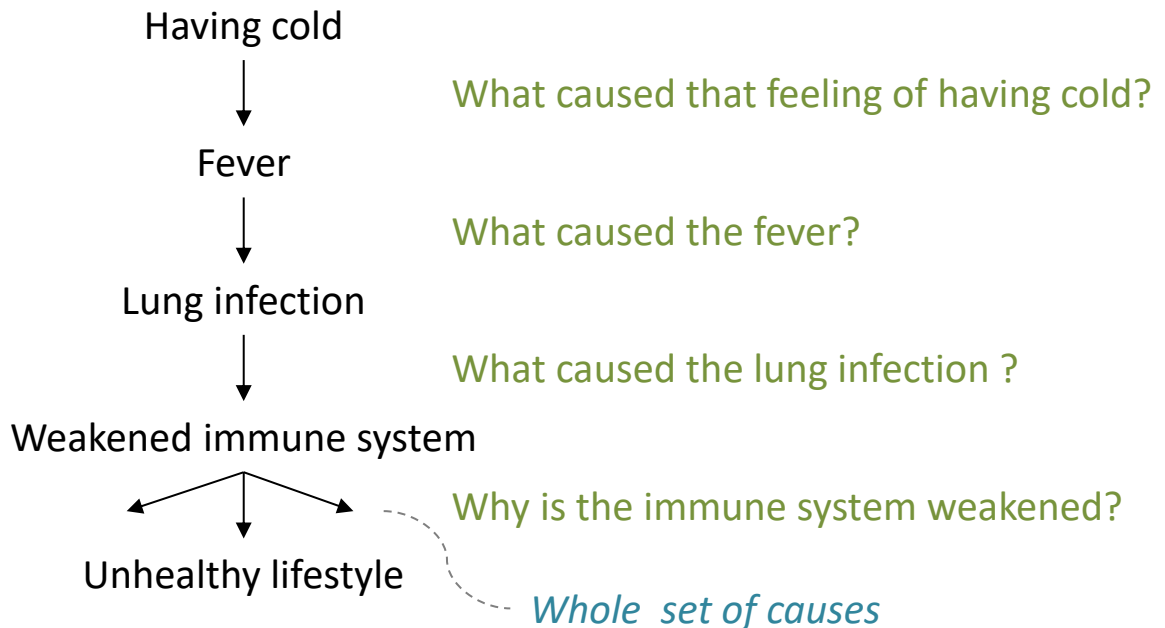
Problem statement:	Someone feels cold
Demand:	Something to get warm
Expectation:	Feeling fine once problem is solved
Need:	Something to get warm
Objective:	Feeling warm (normal)
Requirement:	Must keep person warm or deliver warmth
Solved:	When demand is satisfied, when person feels warm (normal) again.

Does this look right for you?

How About Alternatives?

- Go into the sun
- Going inside the house
- Switch the heating on
- Take a warm bath or hot shower
- Get a blanket
- Drink a hot drink
- ...

How about questioning?

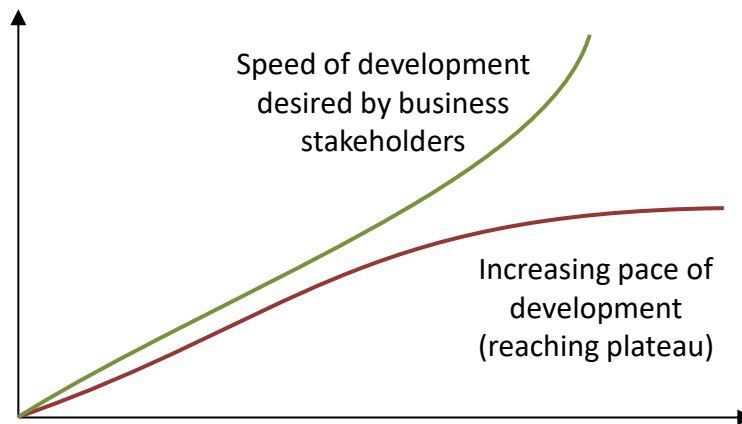


The real problem can be solved and real solutions can be found.

Questions provide a totally different view
on the problem
and on the problematic situation.

TRAP: No LIMIT

10/01/2019



The Business Stakeholders have **NO LIMIT** in their demand of:

Maximising

- Control
- Pace of development
- Adaptability
- Number of features
- Instant result, knowledge, ...

Minimising

- Cost
- Risks
- Effort

Maximising benefits,
Easier and lesser effort
And masking the own
shortcomings and limits

Informatics (or IT) seeks maximal time, resources, autonomy and reward.

Notes:

The business world is dynamic and competitive. Speed is important.

If the business community changed a decision or a choice, or if a mistake has been made, then IT has to be able to correct this as quickly as possible. This puts pressure on IT-people. The IT-community has to adapt. This is a symptomatic 'solution'. How about preventing bad choices and decisions being made, reducing mistakes, wrong problems being 'solved' or inappropriate solutions being demanded? The solution for this is a true Analysis.

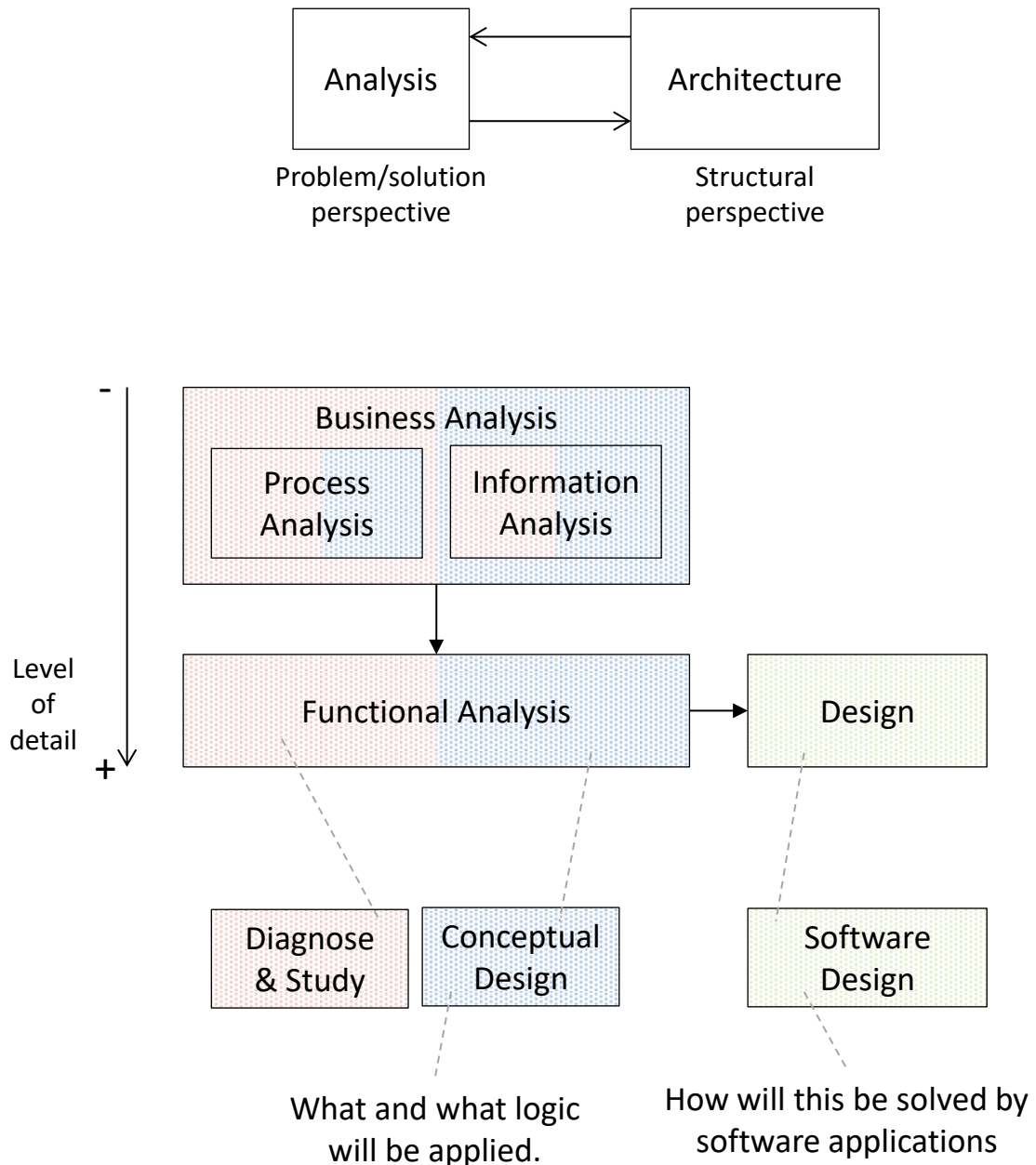
Confusions

- Understanding the demand \neq Understanding the problem
- Understanding requirements \neq Understanding the situation, the context, ...
- Understanding requirements \neq Understanding the required solution

Some Dangerous Assumptions^{30/10/2018}

- The complaints or the demand tell us what the problem is.
- The complaints must be resolved.
- The demand or requirements tell us what is needed.
- A demand or requirements is based on an analysis.
- A demand or requirements is reliable.

Relation Systems Analysis - Architecture



Notes:

If a diagnosis has been performed at the level of BA, then the FA doesn't have to do a diagnosis. However, it is possible that the FA refines the diagnosis, new information about the problem changes the diagnosis or new problems are detected (like dead bodies in the closets).

Managing The Stakeholders Contact Data



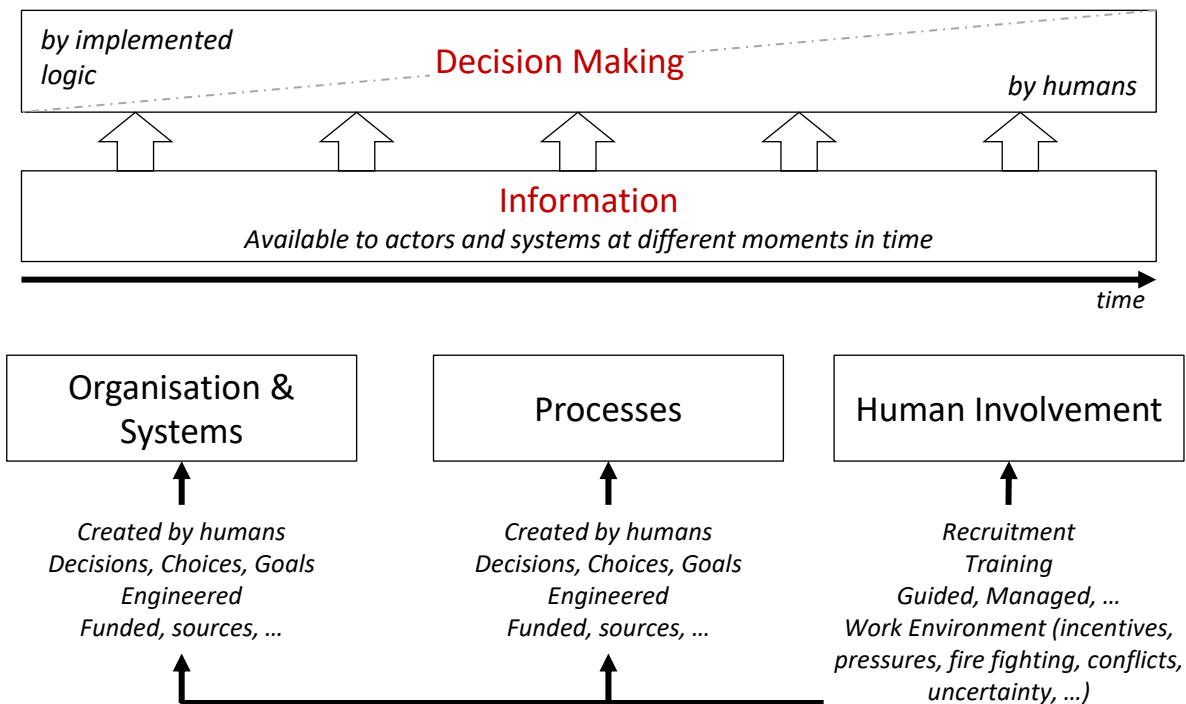
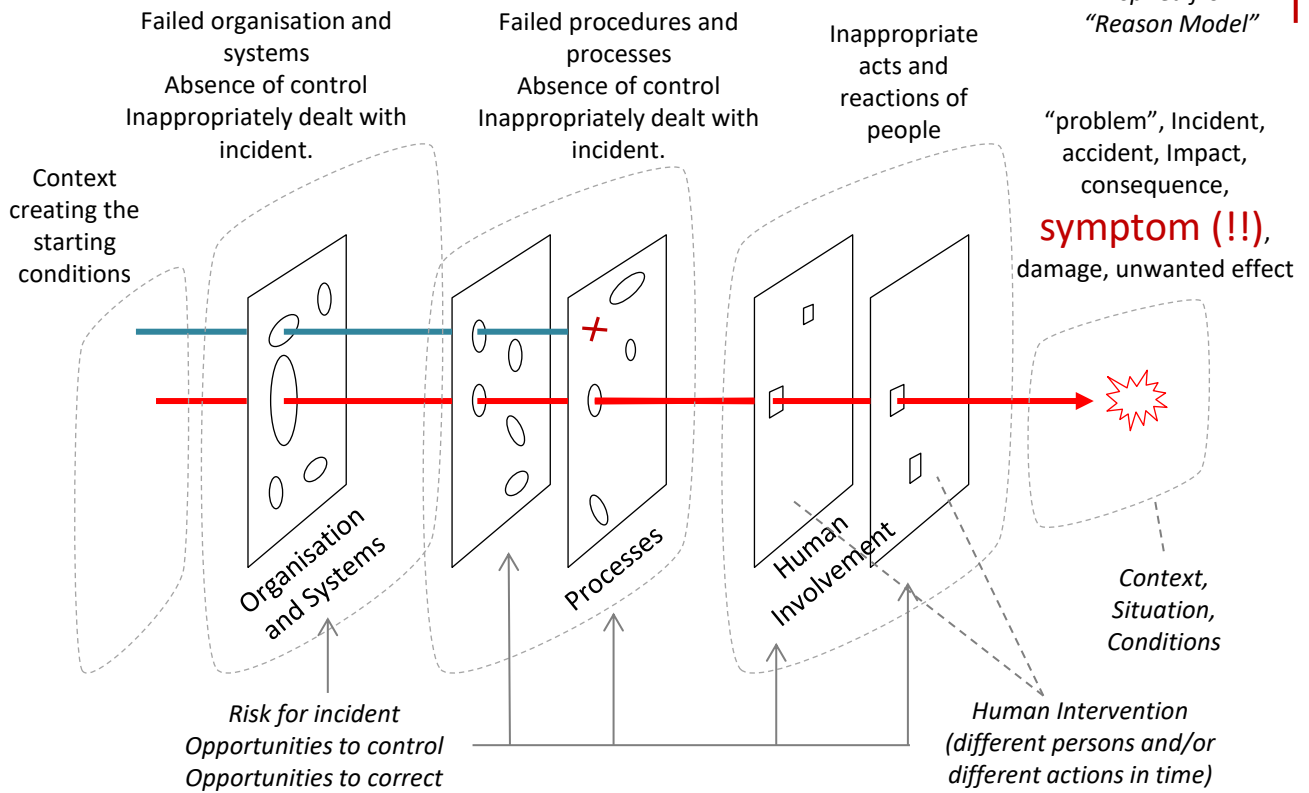
“Managing”

- Their vision
- Their understanding of the situation
- Their perspective
- Their objectives (linked to their department, service or team)
- Their priorities related to their jobs
- Their function and responsibilities
- The level of control they seek to exercise
- Their level of freedom and autonomy they have
- Their level of required flexibility
- The degree of collaboration among business units or with the project
- Their role, importance and contribution within the overall organisation
- Their agreements and disagreements
- Their support to the project and solution
- Conflicting views
- Their expectations (explicit and unexpressed, unawareness)
- Their apprehensions and fears
- ...

Incident / Symptom Investigation

10/01/2020

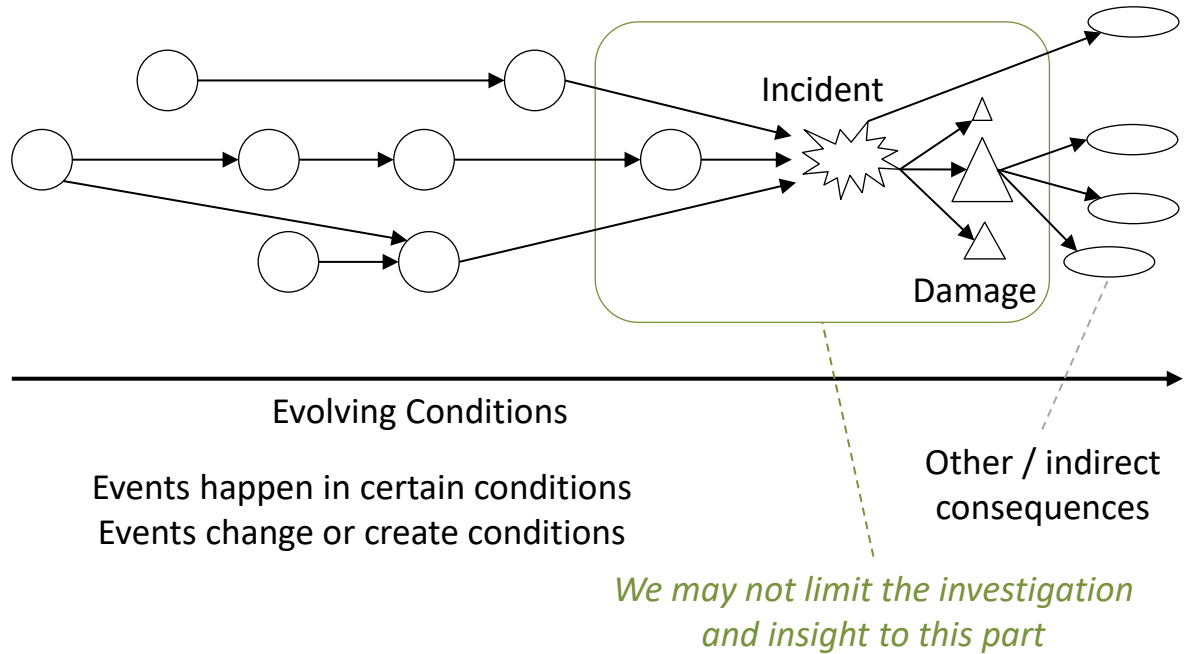
Inspired from
"Reason Model"



Example: If a system or an organisation failed, why did it fail ? Is the engineering approach right?
Who took the decisions? Why did engineers fail ? Do we have competent engineers?

Investigation of the Chronology of the Events

Chain of Events

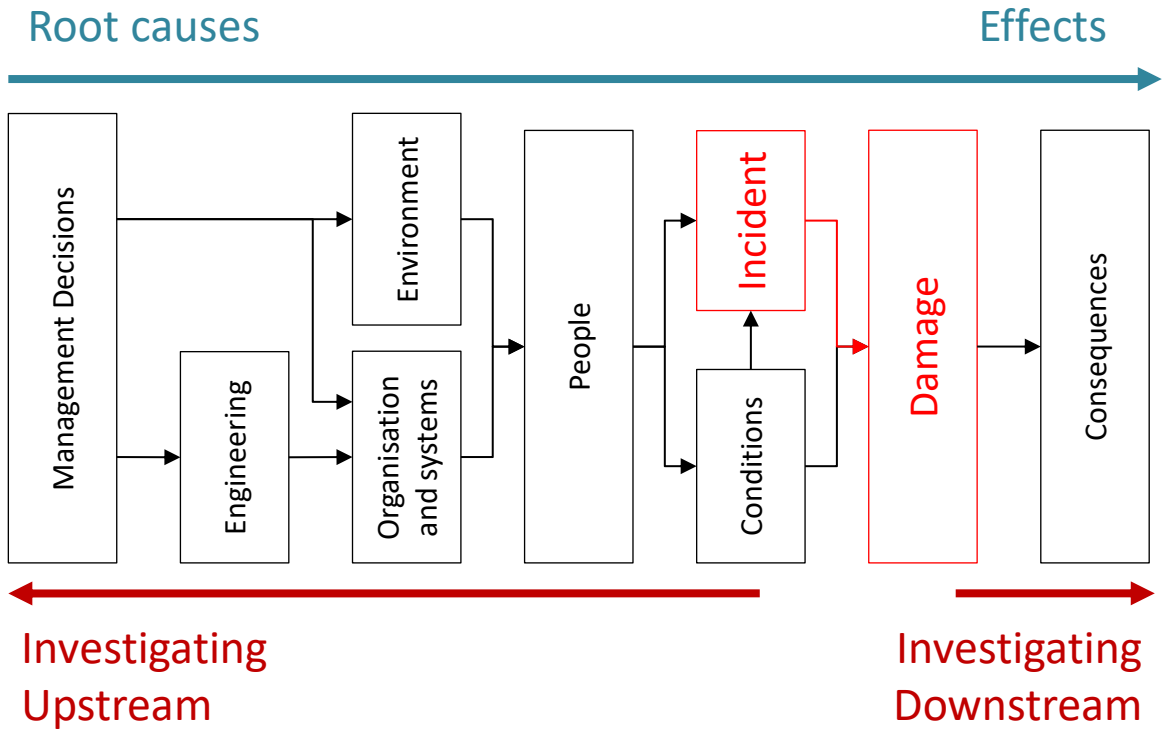


We may also have a **network-model of causes and factors** or link them to the events-model.

Major Common Event Elements:

- Time, duration
- Present Information, knowledge
- Decision
- Motivation, intention, goal, expectation
- Action or process
- Changes something
- Performed by person, organisation, physical system, software system
- Producing a result
- Involves energy, matter, information, money
- Mechanism
- Trigger
- Conditions in which the event happened (starting, during, after event ended)

Investigation of the System (Involved Actors, Elements and Areas)



This diagram may not be suitable for all cases. It depicts a way of thinking. Once understood, it is easy to devise an approach of investigation based on an adapted model.

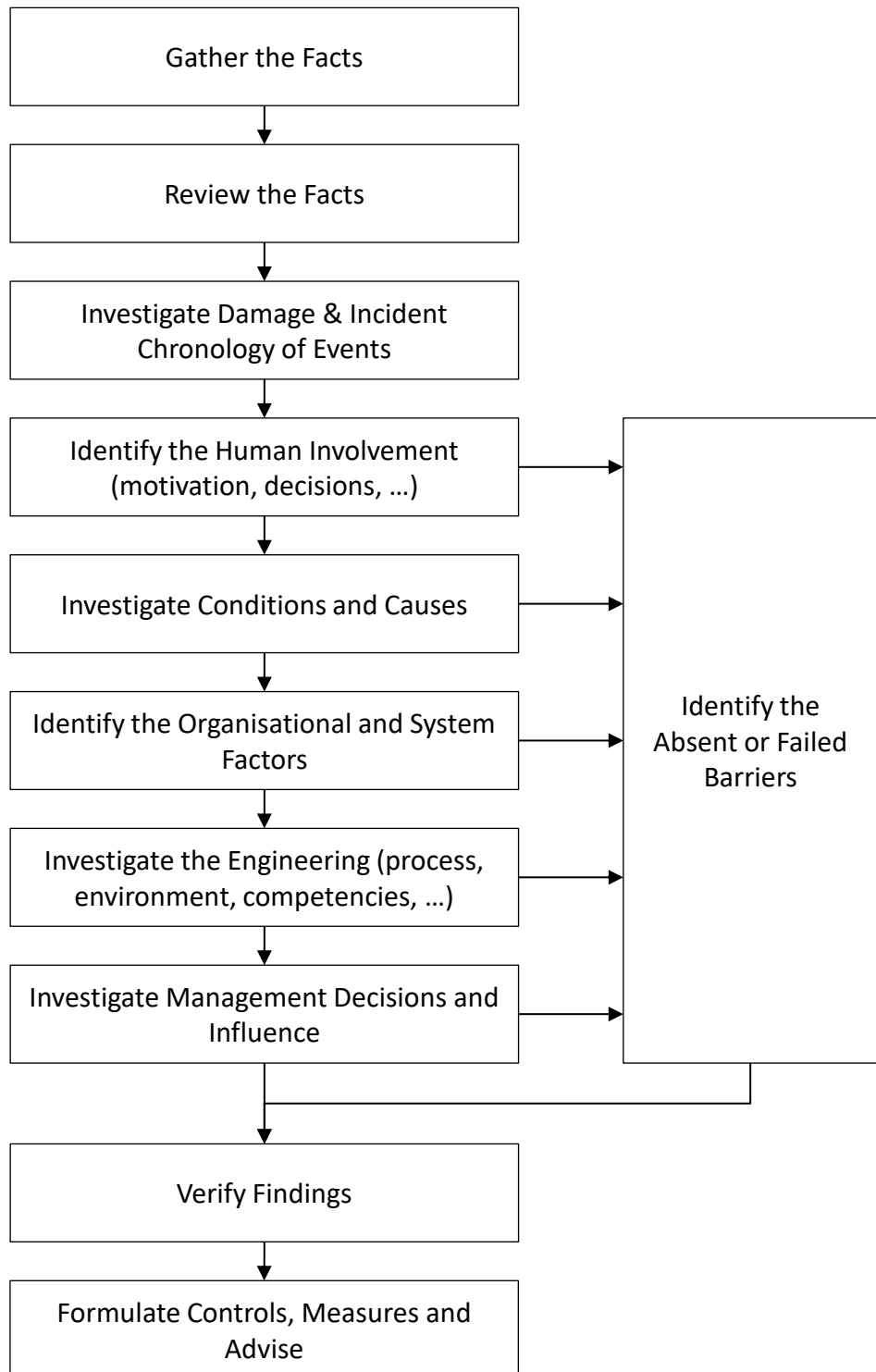
Notes:

See page “Barrier Functions in a System”

Incident Investigation

Systemic Incident Analysis Approach

10/01/2020



Notes:

See page "Barrier Functions in a System"

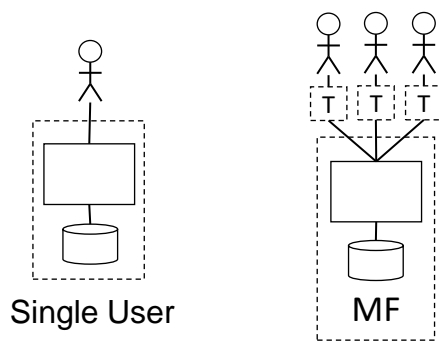


- ARCHITECTURE -



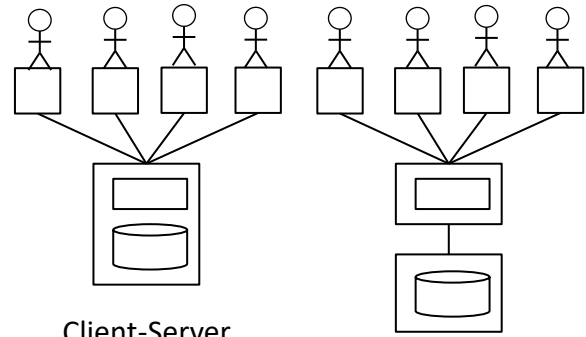
A Few Types of Architectures

30/10/2018



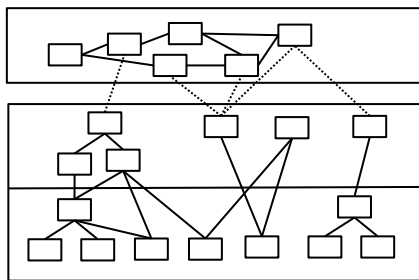
Single User

MF

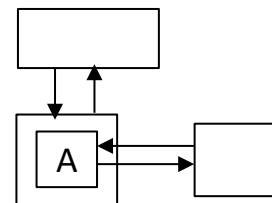


Client-Server

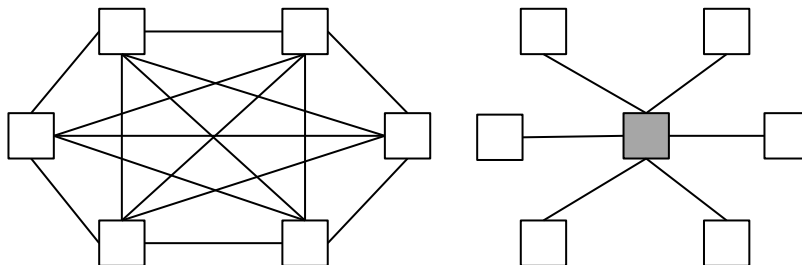
3-Tier



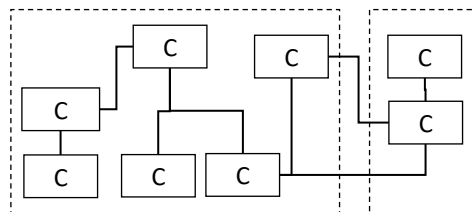
Service-oriented



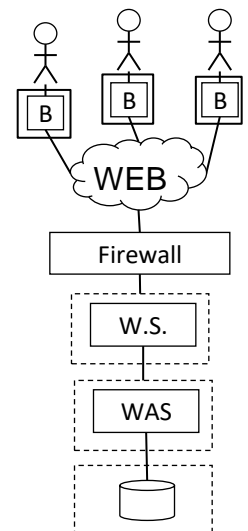
Agent Based System



Peer – to - Peer

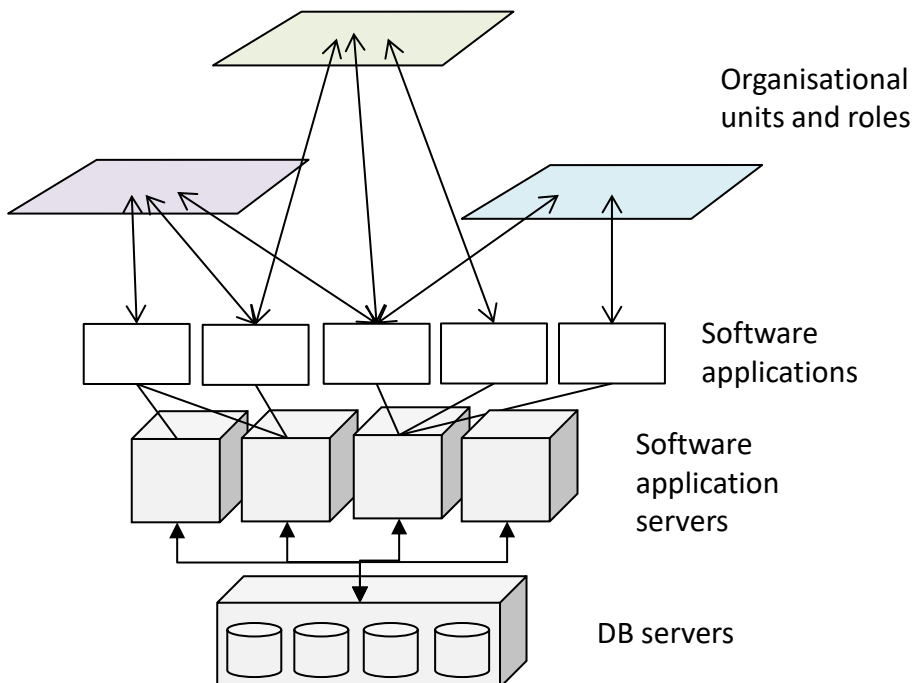
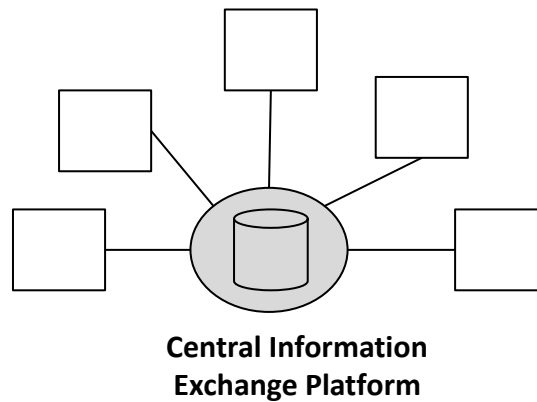
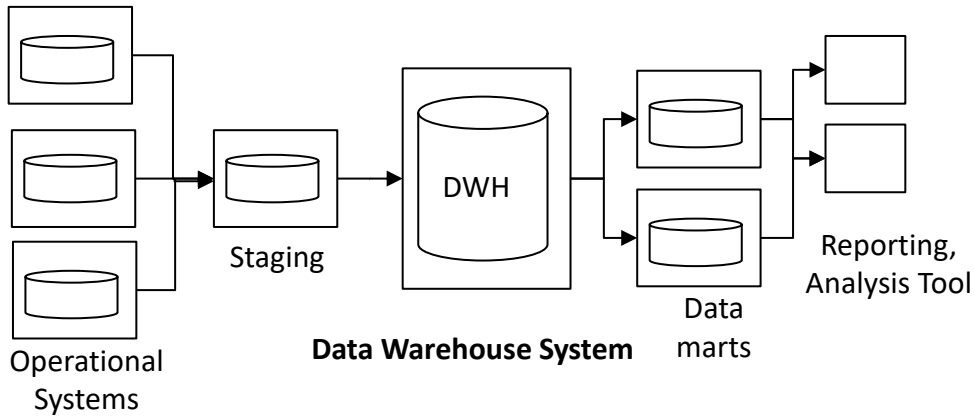


Component-based



A Few Types of Architectures

30/10/2018



“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.”

Melvin Conway

This is actually an observation (and not a law) that organisations have the tendency to design systems that reflecting the organisational structure.

Understandably, copying the structure of the organisation into the system is the first thing that may come to the mind of the designers and architects.

However, *sometimes, it is better* that the architecture matches the organisation. The organisation itself is the supra system. And *sometimes, it is not the best* thing to do.

The important lesson is to study the communication structure, communication flows of an organisation.

An alternative way of organising systems is the information architecture.

Factors Determining the System's Architecture

Indicative List of Factors (not exhaustive)

- Number of actors
- Geographical spread of actors
- Geographical spread of usage of information (capture, processing, usage)
- Other existing systems and infrastructure
- Functional decomposition/organisation
- Different involved independent parties
- Distribution of responsibilities among the involved parties (sponsors, stakeholders, users)
- Nature of information (type, structure, subject matters, ..)
- Communication structure and channels
- Degree of connectedness of information
- Required systems qualities (flexibility, evolvability, reusability, scalability,..)
- Emphasis on workflow, on processes, on data, on security, on system's qualities
- Degree of expected resilience
- Degree of scalability
- Degree of dynamism in information change
- Degree of spread and fragmentation of information
- Push or pull philosophy (in information distribution)
- Nature of processes (formal, ad-hoc, cohesive, event-driven fragments of processes)
- Degree of desired integration
- Usage of information
- Ownership of information
- Degree of control over systems, processes and information
- Degree of sharing information (desired, achievable)
- Required level of standardisation and compliancy to some laws, policies, agreements or rules
- Amount of information
- Future plans, intentions, ...



- DESIGN -



People have **virtues** and **character flaws**.

As an individual, it has limited effect on the surrounding.

Systems are meant to obtain a greater effect. They are designed to amplify the capabilities and thus the power of humans.

Systems may amplify the virtues. But the system can also be abused by people. Then it will amplify some character flaws.

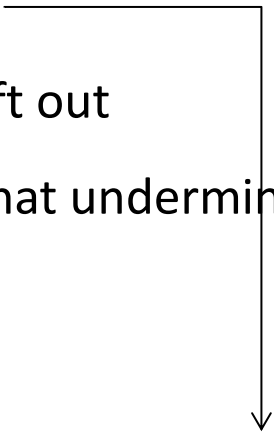
ALL Systems have to be designed

- to amplify the virtues
- while mitigating human flaws.

+ Thinking **carefully** about **ALL** the **upsides**, **downsides**, **implications** and **consequences**.

Systems designers, architects and organisation developers have to pay much attention to this.

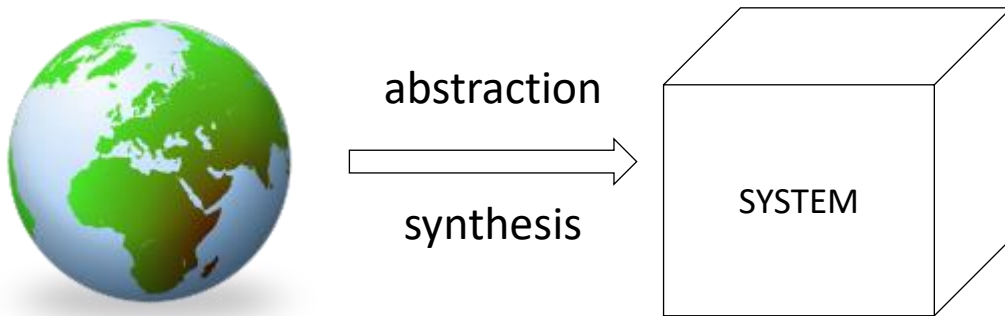
- In line with the **purpose**
- **Simple**, no unnecessary complexity and no oversimplification
- Easily **adaptable**
- **Expandable** (horizontal)
- Can be built on top (vertical)
- Reflects **reality**
- Nothing can be left out
- No mechanisms that undermine the system



Requires
excellent abstract thinking skills

System's design has to reflect the real world (reality)

(as much as possible)



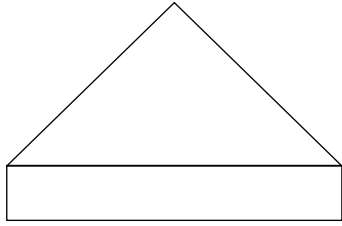
Understanding the world, the reality, the “what is”, the objective reality (versus partial subjective perspectives)

Analyst's key abilities

- Learning and understanding the reality
- Discerning the signal from the noise, the essence from the details, the truth from the distortions
- Abstraction and Synthesis skills

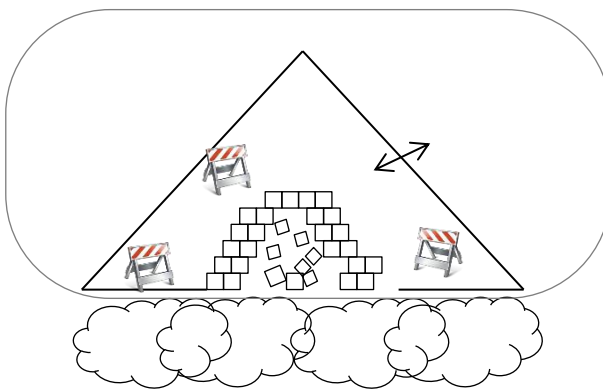
Major Types of Design Flaws

30/10/2018



Solid foundation, harmony, balance, stability, ...

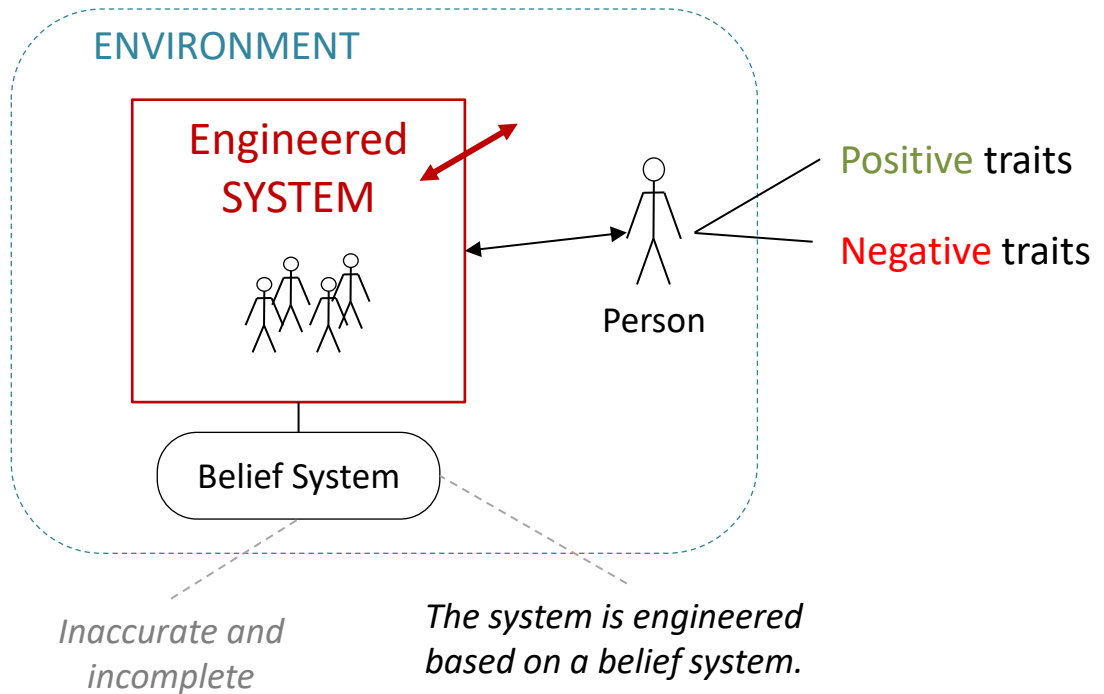
Major types of design flaws



1. Solving symptoms, local problems
2. Lack of controls
3. Inappropriate structures
4. Incomplete solution
5. Imbalances
6. Internal built-in limitations preventing a full exploitation
7. Internal mechanisms undermining the system (may create imbalances)
8. Inappropriate internal organisation
9. False beliefs, assumptions, principles, values, ... (foundation)

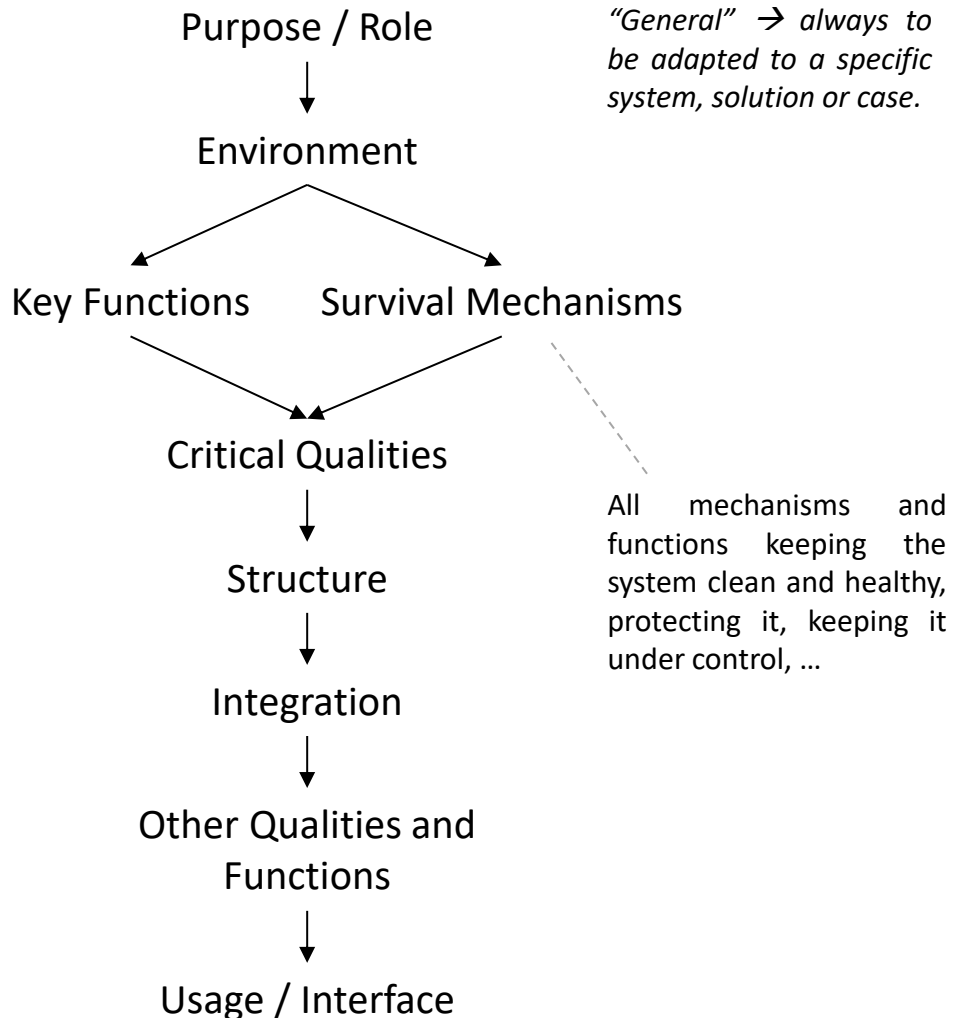
Major Sources of Poor Design

17/09/2019



1. Wrong diagnosis (trying to solve consequences, symptoms)
2. Not understanding the problem
3. Not understanding the environment, the context, the circumstances surrounding the problem
4. Wrong priorities
5. Not respecting purposes and goals
6. Not understanding systems and information
7. Inaccurate, distorted, incomplete belief system
8. Not taking into account the negative traits of people. People can be part of the system (solution) or they can use the system.
9. Not understanding or underestimating the implications, consequences, risks and limitations of the future solution

General hierarchy of aspects and priorities for designing a system



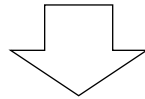
The higher in the hierarchy, the more it impacts (the core of) the design.

Notes

Usage comes last because usage is determined by the interface. This interface is not the system. It is a layer on a system allowing users or other systems to interact with the system. One system can have different interfaces. An interface can be complex; it can be a system on its own.

Example of different possible priorities:

System	Top Most Key Aspects
• Military submarine	under water, resistance to water pressure
• Stealth fighter jet	air/flying, aerodynamics, stealth aspect, agility
• Race car	road, speed, road holding
• Nuclear power plant	security, energy production and transportation



System's key aspects
define
the design process

A second factor defining the design process :

The interactions of some factors
upon each other

For example, the strength of a structure influence the weight. The weight influences the performance. This may require a some kinds of adjustments (iterations) until a right and balanced solution is found.

Notes

This is

Hierarchy of aspects / Priorities for designing a system

PROCESS

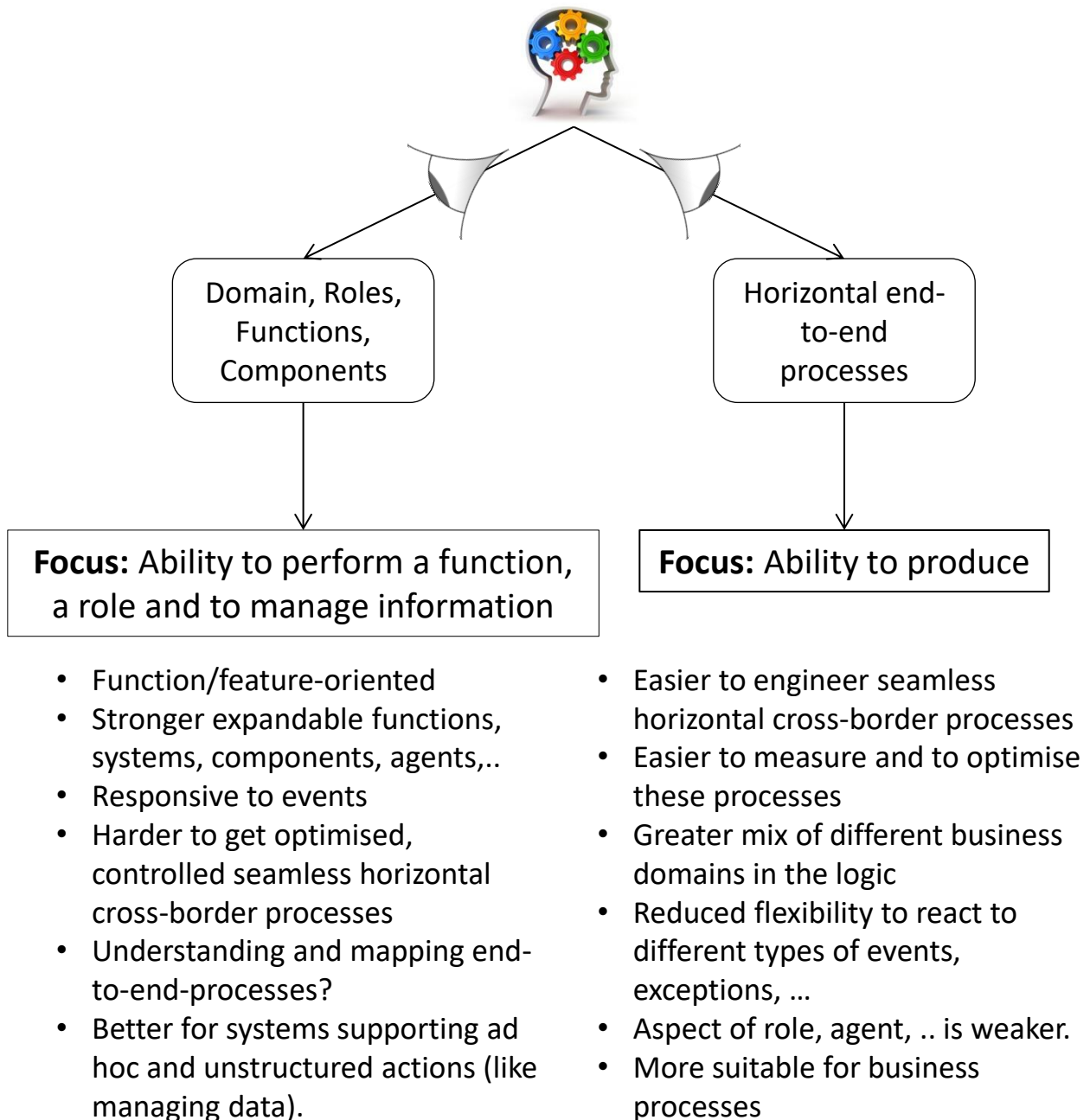
1. What is the top most important aspect?
2. Repeat the question to obtain the hierarchy. Some aspects may be equally important.
3. Design the system accordingly to the obtained hierarchy.

The obtained hierarchy serves as overall guideline for the systems design approach. However, it is not a strict hierarchy. And trade offs decisions may have to be taken.

Notes

This is

Two mind-sets / perspectives to consider when engineering processes



1. Preparation

- Identify Trigger
- Identify Outcome (all products, changes, ...)
- Existing Process Exists?
- Identify start of process, end(s) of process (across the entire organisation)
- Identify Stakeholders
- Gather information
- Determine the established performance goals

2. Modelling of the HL Business Process

- Identify order and add:
 - main stages / phases
 - the different obligatory intermediary results across the process
 - the main business decisions
 - the purpose and intentions
 - the concerned locations
 - the pre-conditions
 - the criteria
 - the end-criteria
- Record specific key business expertise to take into account during the detailed modelling
- Verify the process

3. Detailed Business Process Modelling

- Model the process
- Main case
- Exceptions – Exception handling
- Assign executors (ppl, machines, software apps/ computer)
- Check inter-process interactions
 - Ex.: Delivery process & Client's address change
- Check & define user access authorisations
- Add controls
- Add measures
- List involved information entities
- Are they present in the information architecture and databases)?
- Check the states and the events triggering state changes of these entities for effects on the process or if the process changes the states as noted?



- PROGRAMMING -



Internal Organisation of Program Logic

30/10/2018

Unreadable, confusing, incoherent, unorganised code
increases effort to adapt and risks


Organisation of code increases clarity

- For understanding
- For reducing risks
- For people needing to maintain the code
- For re-use
- To avoid increasing the entropy
- Increases maintainability and evolvability
- Lowers cost and increase the longevity of the system

A few methods:

- **By component, package, library**
 - Loosely coupled externally and with a high internal cohesion
 - Often about a same subject
- **By Layer**
 - Higher layers have lesser details and use code from the lower and more detailed layers.
- **By Kind (By type, by subject, purpose, ...)**
- **By Technology**
- **Or a mix of the previous methods**

Check the “SOLID principles”



Many assume knowledge of
programming languages and
technologies suffice
to be a programmer.

As a consequence

Software Engineering

and

Programming Techniques

are ignored by many



Types of Programming

- Structured programming
- Procedural-oriented programming
- Functional programming
- Logic programming
- Event-driven programming
- Object Oriented programming
- Scripting

Programming Techniques

- User Interface Design
- Interface design
- Multi-user
- Multi-language, multi- ...
- Distributed computing
- Data formatting
- Patterns
- Logging, tracing
- Messaging
- Buffering techniques
- Commenting
- Sorting Algorithms
- Linking, Chain algorithms
- Encryption and decryption techniques
- Graphics, Sound
- Fuzzy logic
- Transaction
- Recursively
- Back tracking
- Dead lock
- Immutability
- Lambda
- Safe Call
- Closures
- Concurrency
- ...

Environments

- Apps
- Web
- Mainframes
- ...

Is code optimisation overdoing?

An optimisation reduced the processing time of a transaction by 0.1 second.

# Transactions	Gain
10.000	16,6 minutes
1.000.000	27,7 hours

Since computers work extremely fast, 1 hour of computer time represents a lot of computer time and computer work.

A company has more than one computers, running several applications day and night, executing many different transactions and operations.

The global optimisation is more than code optimisation. An efficient data architecture, usage of performant technologies, optimised systems, and correctly configured databases are also part of the optimisation.



- TESTING -



TESTING ACTIVITIES

Synonyms: Verify, Control, Check, Test, Feedback, Evaluate, Review

Testing activities include all kinds of verifications (even those done as part of the analysis), simulations, trials, prototype, Proof of Concept, reviews, ...

These activities happen always during the whole course of the project, from the very beginning, regardless of the project phase.

The **objective of testing activities** is to know the obtained information is correct, the understanding is correct, the decisions are right, a concept will work, the design is fine, the work is well done, and so on.

Software can only be tested once it exists, after it has been programmed. But many other things can be tested earlier.

ORGANISED TESTS

Specific tests can be organised during the course of the project (regardless of the project phase)

- Proof of concept
- Simulation
- Reviews
- Validation
- Unit tests
- Integration tests
- Data quality tests

TESTING PHASE at SDLC-level

The Test phase is the project phase that focusses on executing a global set of tests right to make sure the product can be released and no bugs make it to the operational level.

The Test Phase

is **NOT** the phase in which
the **first tests** are performed
to **see if they did a good job.**

The Test Phase

is the phase in which
the **last, the ultimate, tests** are performed
to ensure the product is ready for the production environment.

Tests provide a feedback from which can be learned and correct. This learning is about corrections and adjustments. More learning can be done by techniques such as **Proof of Concepts**, **Prototypes** and **Pilot Products**. This kind of test methods 'tests' ideas. These and other explorative methods don't replace Analysis. They are a part of Analysis.

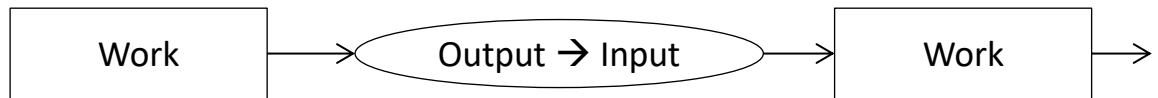
"TESTING PHASE" in the WBS

Testing activities ≠ Project Test Phase

The project's WBS's often contains a branch called "Test Phase" or "Testing".

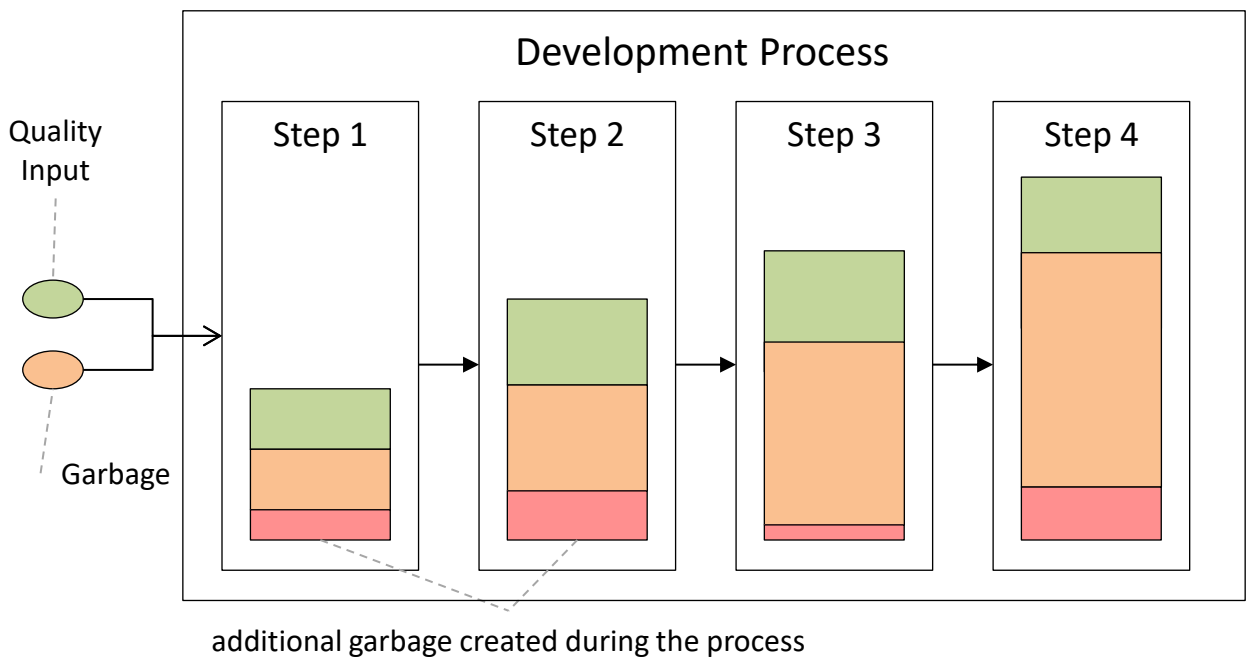
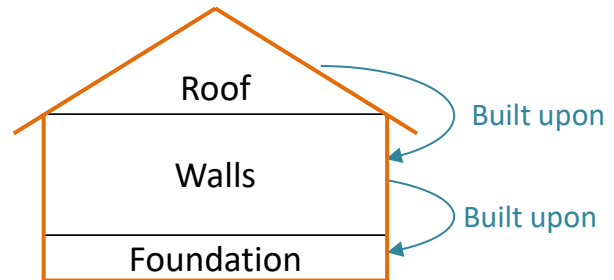
This branch contains the planned activities of the "Test-phase". However, this Testing section of the WBS also include a lot of other testing activities which are executed outside the "Test phase".¹

GIGO-prone process



Project = Progressive Elaboration Process

A house is built upon its foundation. Its roof rests upon the walls.



while the project is on track and deliver the demanded product within scope, on time and on budget.

Notes:

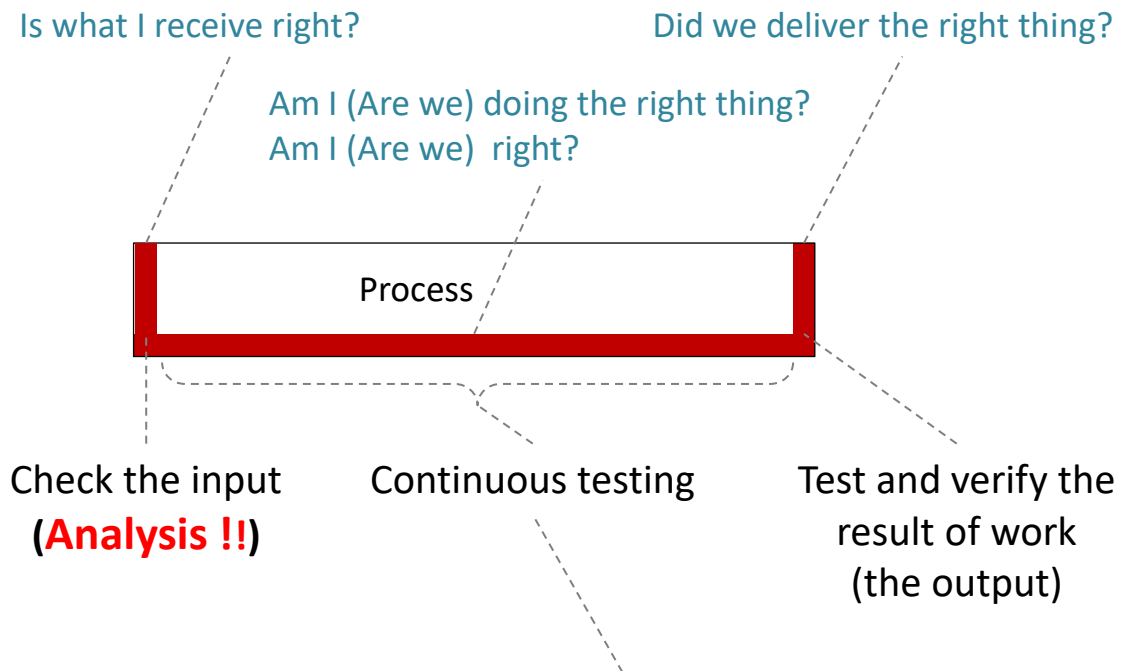
All work requires an input. And often this input is the result of previous work. Garbage may not be inputted in the project or be created by the project and to be dragged all the way downstream through the process.

Avoiding GIGO in Projects

30/10/2018

No GIGO in project process

No creation of garbage



Principle: **Test at the end of a each unit of work**

For example:

- A problem has been diagnosed → test the cause
- An area has been analysed → test the understanding
- A part of the software has been designed → test the design part
- A set of screens has been designed → test the screens
- A component has been developed → test the component
- An integration has been done → test the integrated components
- An artefact has been produced → test the artefact

The Test phase in a project SDLC is the final complete ensemble of tests before the product is released to production environment.

The goal of this ultimate large scale testing is to verify whether the product is ready for production environment.

- The parts of the product function as expected. These products can be integrated. The whole product meets the requirements, and has to be able to meet the goal (or to contribute as expected). Although, at this stage, there has to be already a great confidence in these aspects.
- The product has to function as expected.
- Ensuring the product is ready for being released in the operational environment. It is the last chance before going into operations to detect and remove bugs.
- Idea: Maybe we should have a
 - *“Regular Testing”*: normal testing of the software being developed. This testing occurs as soon as it is possible (unit testing, integration tests, deployment tests, ..)
 - *“Pre-release test phase”*: final tests aimed to get the approval for deployment and release to the operational environment. This phase follows the *Regular Testing phase*.
 - This does not eliminate the need of testing thorough the whole course of the project.

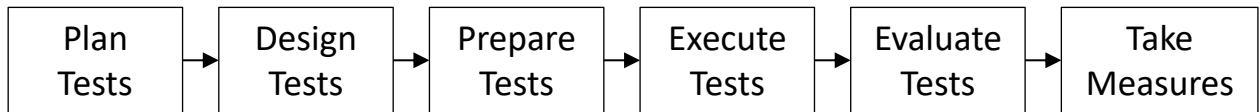
Notes:

Wrong assumptions:

- It's during that phase that the first tests happen. The project has to wait until the test phase to do tests
- Testing happens only during the test phase.

In a WBS, all tests activities, regardless of their phase, can be gathered in a “Test”-branch of the WBS. (A WBS-branch is often confused with a project phase.)

Main steps:

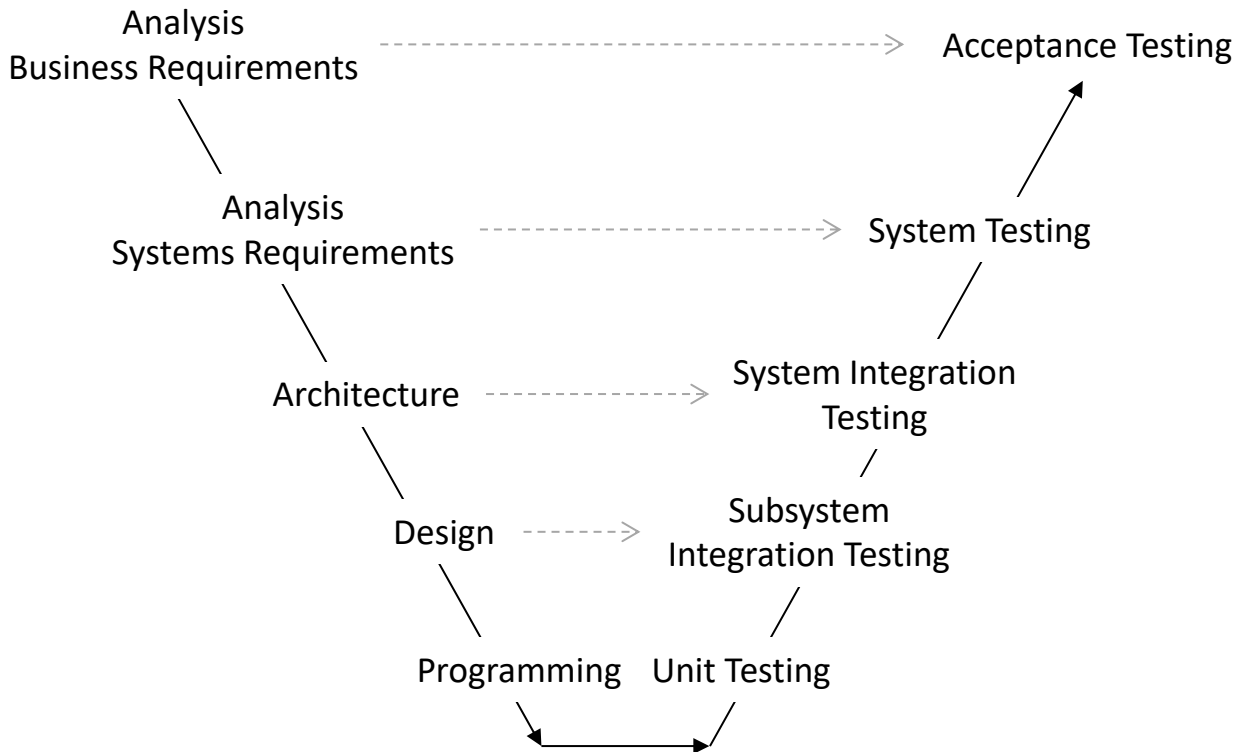


Following activities can be added

- Reviewing test scenarios
- Validating test scenarios
- Acquiring test tools
- Developing test scripts
- Establishing test environment check list
- Preparing test environment
- Testing the test environment
- Test the tests
- Preparing test data
- Training the testers
- Controlling test execution
- Analysing test results
- Analysing defects
- Test Environment Clean-up
- Elaborating a Test Report
- ...

Related:

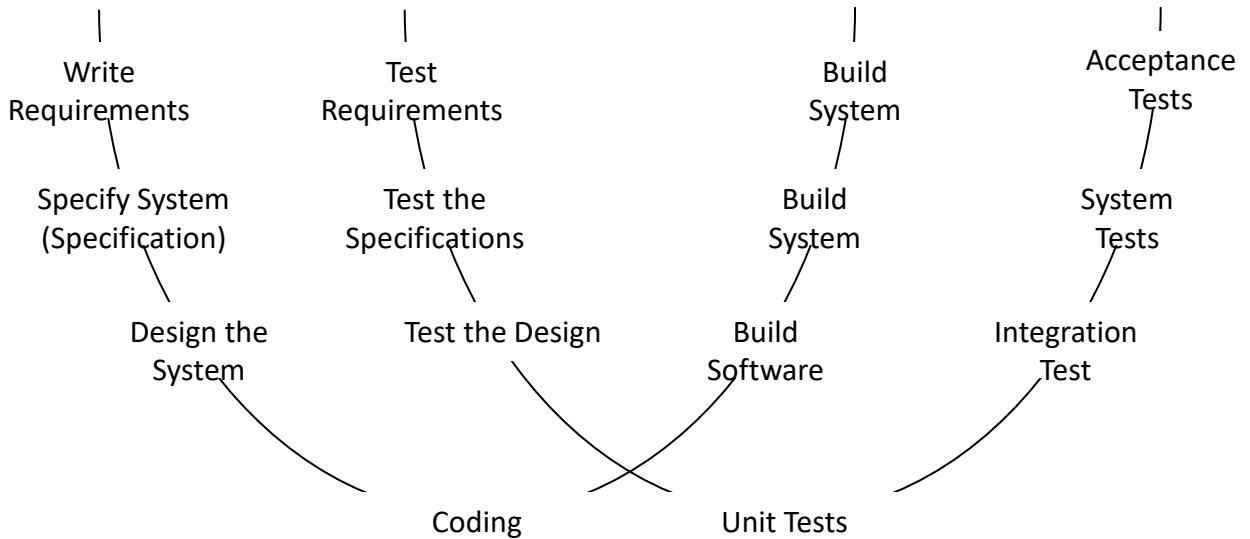
- | | |
|---|-----------------------------------|
| - Acceptance Criteria | - Tests incidents |
| - KPI's | - Defect communication |
| - Metrics | - Review Matrix |
| - Validation Matrix | - Test Data, Test Data Management |
| - Test Log | - Test automation |
| - Test Strategy | - Cycle Runs, Manual Runs |
| - Test Model ((network)model of subsequent tests to be carried out) | - Evaluation meetings |
| - Bug reporting | - Learned lessons |



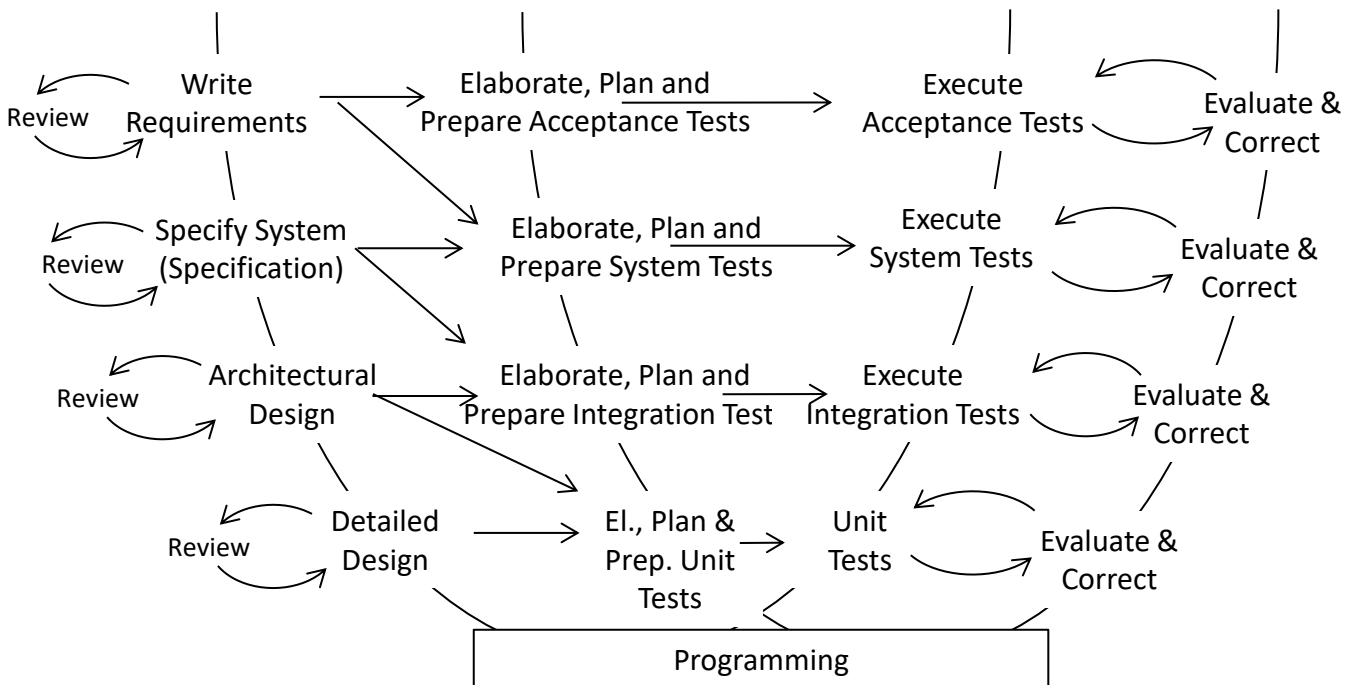
Notes:

V-model shows the order in which some tests are being performed (and why)

Variant 1



Variant 2



Notes:

- Improved V-model - Many variants of the model exist
- Easily to be incorporated in Waterfall-type approaches
- Understand the idea
- Activities related to testing start way earlier than at the end of Programming phase.

Testing is **NOT** limited to the Testing phase

Various forms of testing (including checking, verifying and reviewing) are done during the whole course of the project. Specific tests (like proof of concept, simulation, unit tests, integration tests, data tests) can be organised thorough the project. The Test phase is only an ensemble of tests aiming to prevent faulty software to make it to the production environment. It is the ultimate chance to catch bugs before reaching the operational phase.

Testing Phase is **NOT** a “**Proof of Concept**”

The goal of the Test phase is not to find out if the product works or not. Other techniques, like a “Proof of Concept” serves this purpose. When a product reaches the test phase, there must be a fair amount of confidence that the product is ready for production. The Test phase is a final test that should detect the “last” issues and bugs and, once corrected, to provide confirmation of its readiness for release.

Testing Phase is **NOT** a “**Product Discovery Phase**”

Business stakeholders and representatives often see the product for the first time during the Test phase. They should have seen and tried the software product (or any product) during the design activities (prototype, screens mock-ups, ..) and during the programming/building (even partially, non-working software applications can be showed). Major flaws have to be discovered early. The goal of such reviews is not to capture latest preferences and new ideas. If they are important, they should be treated as real changes (depending on their importance and impact).

Testing Phase is **NOT** a “**Learning Phase**”

Learning happens all the time. However, one of the major goals of the Analysis is to learn (not just to be told). Then, a lot should be learned when presenting artefacts, models or the product in construction to the stakeholders. Finally, some learning happens also once the product is in production, leading to adjustments.

ACCEPTANCE TESTING Testing to verify a product meets customer specified requirements. A customer usually does this type of testing on a product that is developed externally.

BLACK BOX TESTING Testing without knowledge of the internal workings of the item being tested. Tests are usually functional.

CHAOS TESTING Testing by deliberately injecting failure into their services and systems (tools: "Chaos Monkeys").

COMPATIBILITY TESTING Testing to ensure compatibility of an application or Web site with different browsers, OSs, and hardware platforms. Compatibility testing can be performed manually or can be driven by an automated functional or regression test suite.

CONFORMANCE TESTING Verifying implementation conformance to industry standards. Producing tests for the behaviour of an implementation to be sure it provides the portability, interoperability, and/or compatibility a standard defines.

FUNCTIONAL TESTING Validating an application or Web site conforms to its specifications and correctly performs all its required functions. This entails a series of tests which perform a feature by feature validation of behaviour, using a wide range of normal and erroneous input data. This can involve testing of the product's user interface, APIs, database management, security, installation, networking, etc. testing can be performed on an automated or manual basis using black box or white box methodologies.

INTEGRATION TESTING Testing in which modules are combined and tested as a group. Modules are typically code modules, individual applications, client and server applications on a network, etc. Integration Testing follows unit testing and precedes system testing

LOAD TESTING Load testing is a generic term covering Performance Testing and Stress Testing.

PERFORMANCE TESTING Performance testing can be applied to understand your application or web site's scalability, or to benchmark the performance in an environment of third party products such as servers and middleware for potential purchase. This sort of testing is particularly useful to identify performance bottlenecks in high use applications. Performance testing generally involves an automated test suite as this allows easy simulation of a variety of normal, peak, and exceptional load conditions.

REGRESSION TESTING Similar in scope to a functional test, a regression test allows a consistent, repeatable validation of each new release of a product or Web site. Such testing ensures reported product defects have been corrected for each new release and that no new quality problems were introduced in the maintenance process. Though regression testing can be performed manually an automated test suite is often used to reduce the time and resources needed to perform the required testing.

SMOKE TESTING A quick-and-dirty test that the major functions of a piece of software work without bothering with finer details. Originated in the hardware testing practice of turning on a new piece of hardware for the first time and considering it a success if it does not catch on fire

STRESS TESTING Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements to determine the load under which it fails and how. A graceful degradation under load leading to non-catastrophic failure is the desired result. Often Stress Testing is performed using the same process as Performance Testing but employing a very high level of simulated load.

SYSTEM TESTING Testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

UNIT TESTING Functional and reliability testing in an Engineering environment. Producing tests for the behaviour of components of a product to ensure their correct behaviour prior to system integration.

WHITE BOX TESTING Testing based on an analysis of internal workings and structure of a piece of software. Includes techniques such as Branch Testing and Path Testing. Also known as Structural Testing and Glass Box Testing

(Src: various websites on the web publish this list.)



DEPLOYMENT





- TECHNIQUES -



- 5W1H
- 5Why's
- MoSCoW method
- Business Case
- CATWOE
- Critical Success Factors
- de Bono's 6 Thinking Hats
- Force Field Analysis
- Heptalysis
- Interviews
- Ishikawa
- KAOS (Knowledge Analysis in autOmated Specification)
- Mind Maps
- MOST analysis
- Net Present Value
- Job Observation
- PESTLE/PESTELI
- Proof of Concept
- Prototyping
- Questionnaires
- Requirements Analysis
- Requirements Engineering
- Rich picture
- SCRS
- Soft Systems Methodology
- Stakeholders Analysis
- Storyboarding
- SWOT
- Task Demonstration
- Use Cases
- Theme / Epic / User Stories
- WBS
- Workshops
- CRC
- ...

Models

ERD

Finite State Machine

Languages

UML

BPMN

Specification and Description Language (SDL)

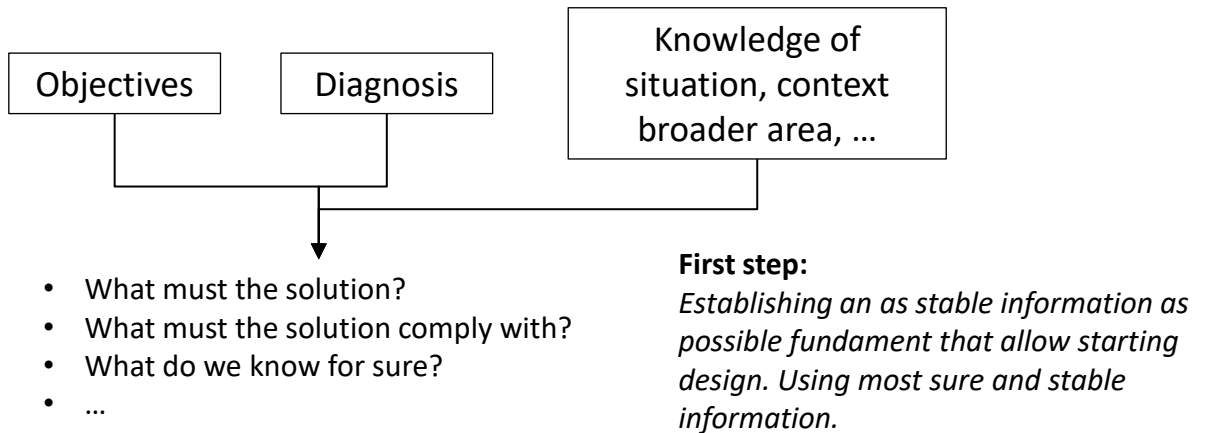
User Requirement Notation (URN)

Use Case Maps (UCM, a URN language))

Goal-oriented Requirements Language (GRL)

Systems Modelling Language (SysML)

Requirements



Requirements give an indication about the future solution to the design process. These requirements will be implemented in the final solution. Thinking about what the solution should be able to do and comply with, is solution-oriented thinking. On the other hand, requirements are deduced from and by the analysis activities. They are not created as a result of creativity.

A **requirement** is a statement expressing a function, a key aspect, attribute, principle, characteristic, capability, constraint or limitation that is mandatory to reach the objective, solve the problem or satisfy the need. The future solution must implement it or comply with it. A requirement is related to a necessity.

The Analyst works with a **set of requirements**.

The Analyst extracts requirements **through investigations, interviews of stakeholders and analysis**. The requirements are **the result of an Analysis, of an investigation**.

The Set of Requirements is an artefact serving as a **criteria**.
The usage of this set of requirements is a **method**.

Notes:

Techniques: Volere, Planguage, EARS (Easy Approach to Requirements Syntax)

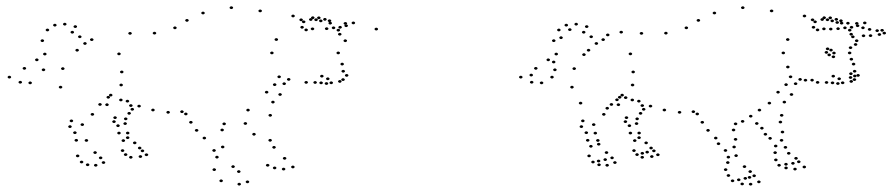
The Requirements technique is a Problem Solving technique

The following metaphor illustrates the functioning of requirements in the problem solving. Each dot can be seen as a requirement.

Elicited (not gathered) requirements provide some information about what has to be incorporated in the future solution.



The more requirements are elicited, the more become known about the future solution.

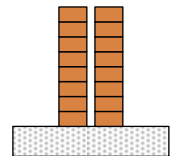
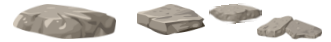


Sometimes, in the end, a pattern showing the future solution may emerge (in the illustration: a bull) from the set of requirements. The requirements help the mind to connect the dots and to fill in the blanks. If no pattern emerges, it is up to the Analyst to conceive a solution. They also help the Analyst excluding very early on solutions that do not meet the requirements. In any case, requirements help conceiving a solution.

After requirements have been elicited, the solution has still to be defined (conceived). Requirements facilitate this task.

Three Main Purposes

- To facilitate the process of conception of the solution by providing key elements that must be incorporated in this solution. It is the first information we know about the future solution.
- Purpose of **foundation** for further thinking. They form the stable basis for creative thinking. (stability of requirements!)
- To keep the conception of a solution in the right direction. Ensuring that the important aspects which have been identified are taken into account in the design, or, preventing to forget or ignored some of them. This can be verified by requirements-based tests.



Requirements can be optional

- 1) There might be several ways to reach a same objective. Each way may have its own sub-set of requirements. Depending on the chosen way, the other sub-set requirements won't be necessary.
- 2) An overall solution may tackle several issues. Some issues can be optional. As a result the requirements linked to these issues, become optional as well.

- Requirements represent needed capabilities, constraints, limits or other characteristics.
- A requirement should state what a system must comply to, not or rarely how. However, this is likely to be in terms of business logic (in particular as business rules).
- Requirements without a right prior diagnosis are unusable.
- Requirements have to be feasible / attainable.
- Requirements should be deduced from the objectives, from the business knowledge and field expertise, from the business forecasts, from the supra-system and from the environment.
- A requirement is the result of an analysis, a deduction, an observation, a conclusion. If a requirement doesn't comply with this and is the product of creativity, then it is not a requirement but already a design statement. Such early design statements, as false requirement, narrow the range of solutions. This is not the role of requirements.
- An approach in which different subjects, like an area, a domain, an aspect or a level of detail, are subsequently analysed is motivated by the objective to elicit requirements about that subject. The assumption is often made that only requirements about the subject being analysed will be elicited. It can't be excluded that when analysing a subject A, requirements impacting subject B or impacting the solution or system as a whole.
- Requirements are by their very nature incomplete.
- The future solution has to comply with the ensemble of final requirements.
- The set of final requirements does not represent "THE SOLUTION".
- Requirements help to solve problems. It's a (possible) stage in the problem solving process. But it is not the magic bullet that replaces problem solving skills.
- Requirements are not a substitution for insight in the situation, in the context and in the broader area. Conceiving a solution based on a set of requirements without understanding the supra-system and the broader environment is a good receipt for disasters. It's better to have this insight and no requirements than to have requirements and no decent broader understanding.
- Business requirements are requirements concerning the business domain, field experience, business activities. It does not mean "owned by the business" or "expressed by the business".
- A future solution not complying with the final set of requirements, provided they are right, guarantees that this solution won't function properly or, at least, be significantly lesser valuable.
- Any solution meeting the set of final requirements doesn't guarantee the right problem is being tackled or contributing to the right goals. It doesn't guarantee the problem will be solved, the solution resulting from it being right and the best, the solution being valuable nor does it guarantee customer's satisfaction.
- A solution matching the requirements can still be refused.

- Requirements are **obligatory** or **alternatives** or **optional**.
 - **Alternative**: different solutions may lead to a same result
 - **Optional**: for expansions of a solution
- Requirements have to be **certain**.

The objective is to get requirements as stable and certain as possible. They have to be based on sound insight and be verified and validated.
- Requirements **can be changed**, **but** decent upfront analysis is done to prevent this. It is possible, yet unlikely to change.

- *Received (demanded)* 💣💀 ?
- *Assumed, guessed* 💣💀 ?
- *Imagined* 💣💀 ?

- Deduced from **objectives**
- Deduced from the **real situation**

Based on knowledge and insight
(obtained by Analysis)

Notes

‘Requirements’ describing what is asked or what someone wants, by definition, these requirements are unreliable. (Usable for SW-development?)

Giving what is asked for is uncertain and short term.

If requirements are nothing more than desired and uncertain statements of desires and assumptions, then the whole approach should be questioned as well as the understanding of ‘requirements’ as a technique.

Qualities

- Unitary, atomic
- Clear
- Complete
- Cohesive
- Consistent
- Non-conjugated
- Traceable
- Current
- Feasible
- Unambiguous
- **Necessary**
- Importance specified
- Verified
- Verifiable, testable

A requirement has first a temporary and uncertain status, like draft or preliminary. Later its status may become more definitive or final when it is found to be true, certain, accepted, validated or agreed (depending of the organisation).

Let's call these the "***final requirements***".

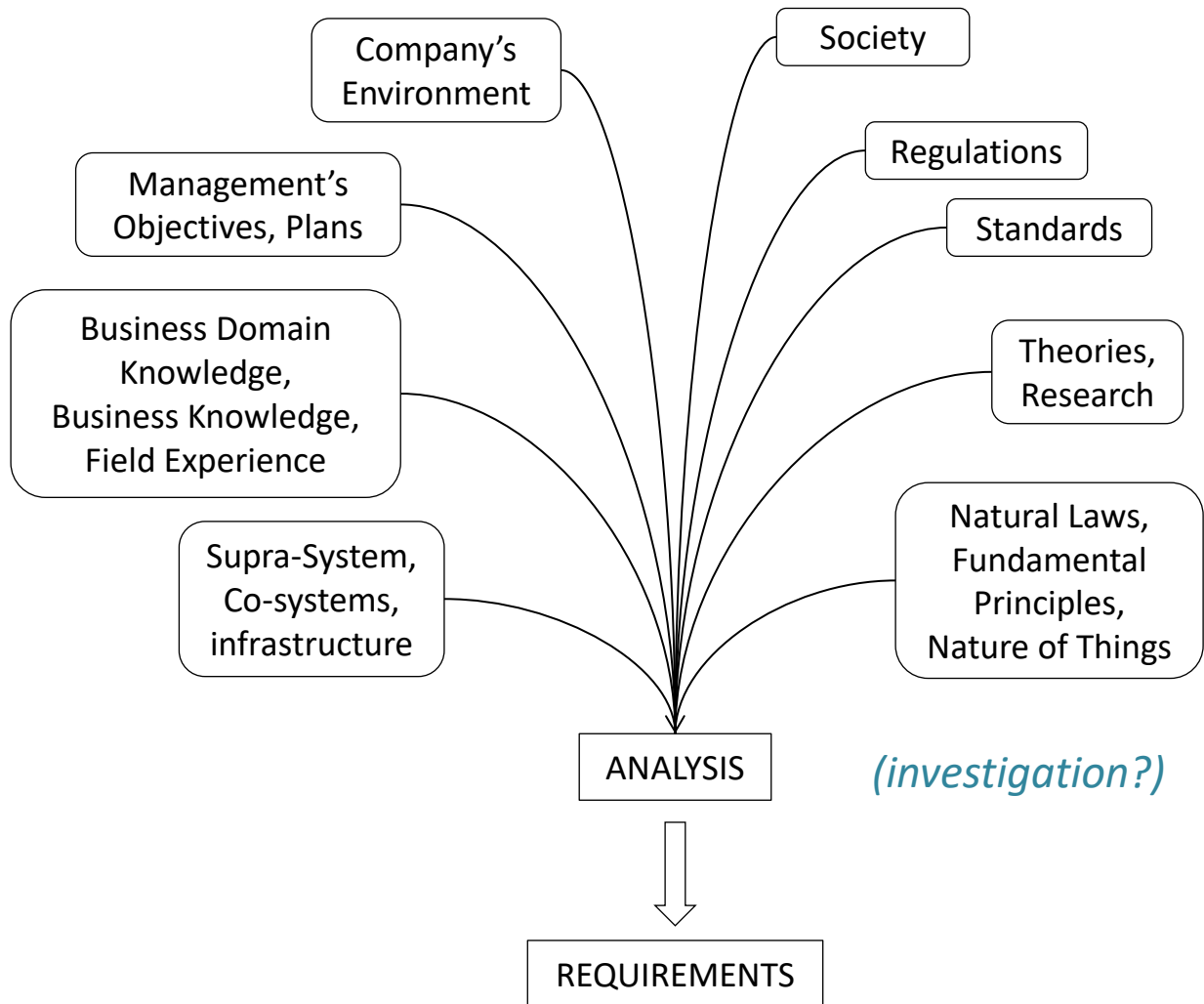
1. Business Requirements
2. Goal Requirements
3. Stakeholder Requirements
4. User Requirements
5. Program Requirements
6. Project Requirements
7. Process Requirements
8. Management Requirements
9. Enterprise Requirements
10. Environmental Requirements
11. Solution Requirements
12. Product Requirements
13. Architectural Requirements
14. System Requirements
15. Software Requirements
16. Change Requirements
17. Regulatory Requirements
18. Quality Requirements
19. Quality-of-service (non-functional) requirements
20. Functional Requirements
17. Non-functional Requirements
18. Usability Requirements
19. Indirect Requirements
20. High-Level Requirements
21. Detailed Requirements
22. Information Requirements
23. IT Requirements
24. Data Requirements
25. Design Requirements
26. (non-)Technical Requirements
27. Testability Requirements
28. Security Requirements
29. Documentation Requirements
30. Implementation Requirements
31. Transition Requirements
32. Requirements Specifications

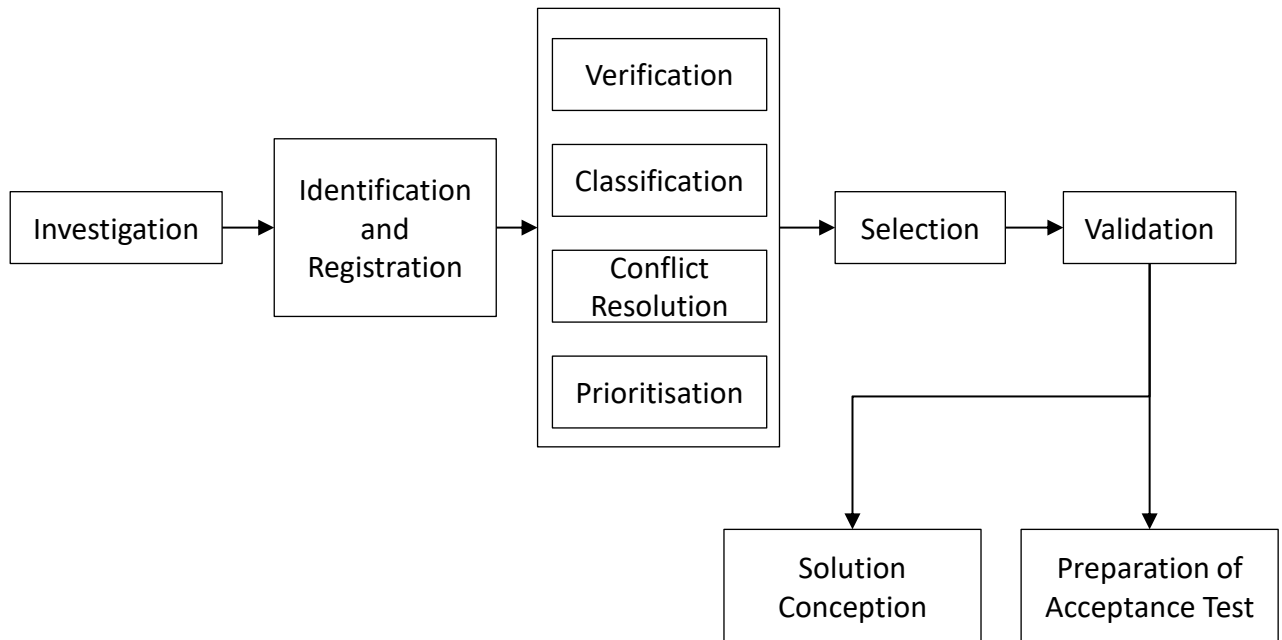


... and there are many more

Sources of Requirements

30/10/2018





Notes:

This model shows a general idea. It is not the only way of working and can be adapted, for example, the selection and validation can be swapped.

Requirements ??

- The new database must be online 24h/24h – 7d/7d.
- The software application must be able to print the customer's data.

These are NOT requirements

- What if the problem can be solved without new database?
- What if another software application would print the customer's data? Or what if the customer's data could be made available in another way?

Whether a new database is required, what software application should print or if printing is required are all design decisions.

**If a statement is based on a design decision,
then it is NOT a requirement.**

Correct requirements:

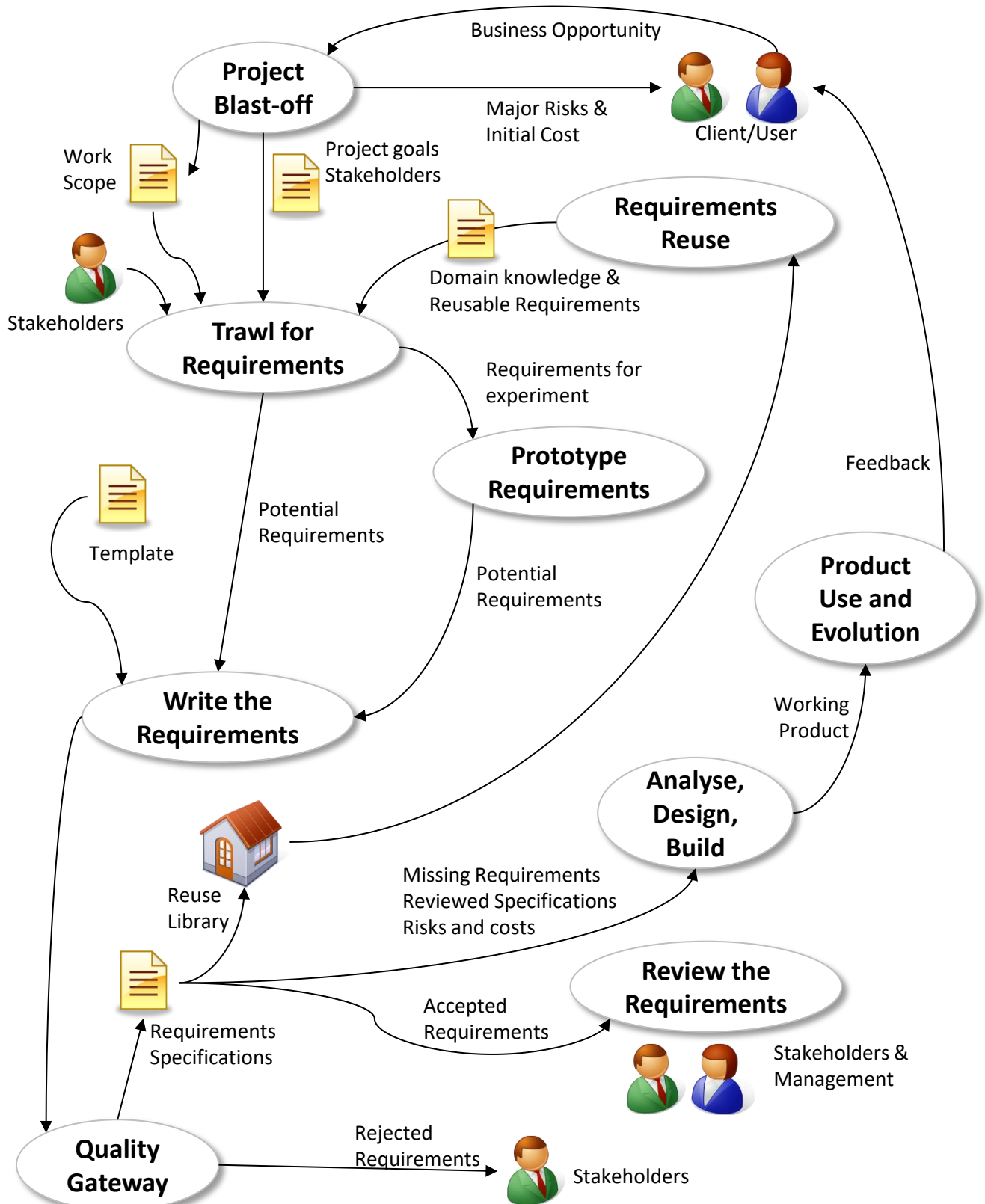
- The information should be available 24/h/24h – 7d/7d
- The customer's data should be made available to the customer.

Notes:

Requirements precede any design decisions. They are not the result of a design decision of any kind.

Requirements – Volere Process

30/10/2018



Requirements are the **result of an Analysis Process**.

Good requirements **result from the understanding** the environment, the situation, the business domain, the objectives and systems.



Requirements are **NEITHER GATHERED NOR RECEIVED !!**

Some requirements are obvious.



Requirements are detected through in depth investigation.

We need to dig deep to bring them to the surface. Some critical requirement are well hidden.

Obtaining bad requirement / bad practices:

- Gathering requirements
- Asked, received or expected from the business community, SME, ...
- Accept requirements at face value
- Deduced from “what the business wants”
- Asking the business community to confirm the requirements they provided

Bad requirements result from an insufficient insight in the context and the situation.

A “requirement” stated by the business community CAN be valid, but **it must be verified** by the Analyst checked with the reality, with what “is”, by **investigating the broader** context and see if this requirement matches and fits in this broader contexts.

WRONG PURPOSES of REQUIREMENTS

Requirements as a goal

“The goal is to meet the requirements.” This shows little care about whether the solution will solve the problem or need or not. Basically, we don't mind to create value. Meeting requirements is good enough. Or we assume that by meeting requirements, value is created. Requirements are not a goal. Meeting requirements is an important criteria among others.

To replace a diagnosis

Requirements are a **post-diagnosis activity**. Requirements result from investigation activities (Analysis). The diagnosis should have been done by the time requirements technique is started.

To understand the problem or solution

Understanding requirements is not the same as understanding the problem or the solution. Requirements only express elements of the solution. They don't describe the problem or context. Having requirements does not discharge the Analyst from understanding the situation and the context. Requirements may describe a fraction of the problem or solution. Requirements are incomplete and may be wrong. Understanding requirements is insufficient to build a suitable solution.

To describe the solution

Requirements describing a solution are called “specifications”. Seeking to have all the requirements is impossible and senseless.

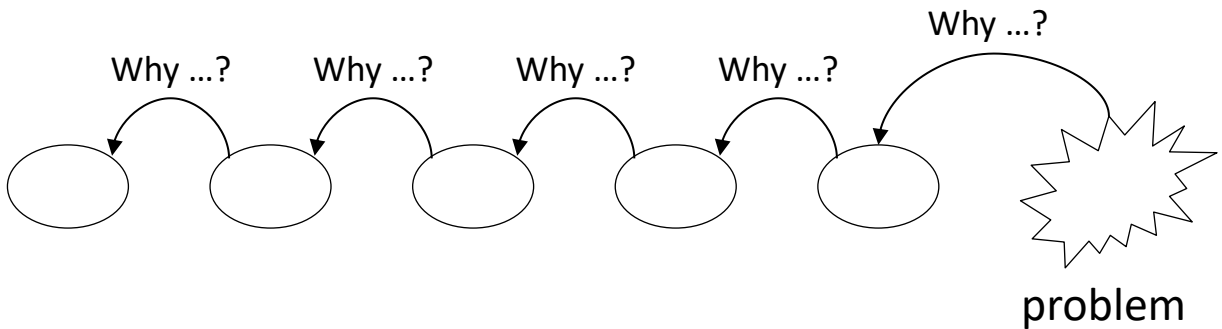
Questions

18/12/2018

1. Why?
2. What?
3. When?
4. Who?
5. Where?
6. How?
7. Which?
8. What kind of ?
9. What for?
10. Whose?
11. How long?
12. How much?
13. How old?
14. How come?
15. How far?
16. Why not?
17. What if?
18. Whom?

5 Why's

18/12/2018



Purpose: Finding the root cause

- The number of 5 shouldn't be taken literally. The idea is to repeat the question as many times as possible.
- The answer may consists of several reasons → branching
- It is simply a matter of answering the question based on the present knowledge. If necessary, it should lead to further investigation (new knowledge) of the question.

- *Who* was involved?
- *What* happened?
- *Where* did it take place?
- *When* did it take place?
- *Why* did that happen?
- *How* did it happen?



THE ANALYST'S RED BOOK

THE END