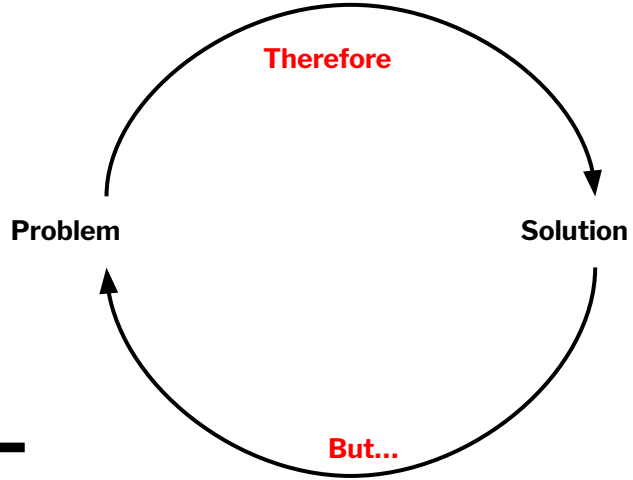


Simplify IT

The art and science towards simpler IT solutions

Author : Maikel Mardjan
Status : Permanent discussion DRAFT
Sharing is Appreciated!

Introduction

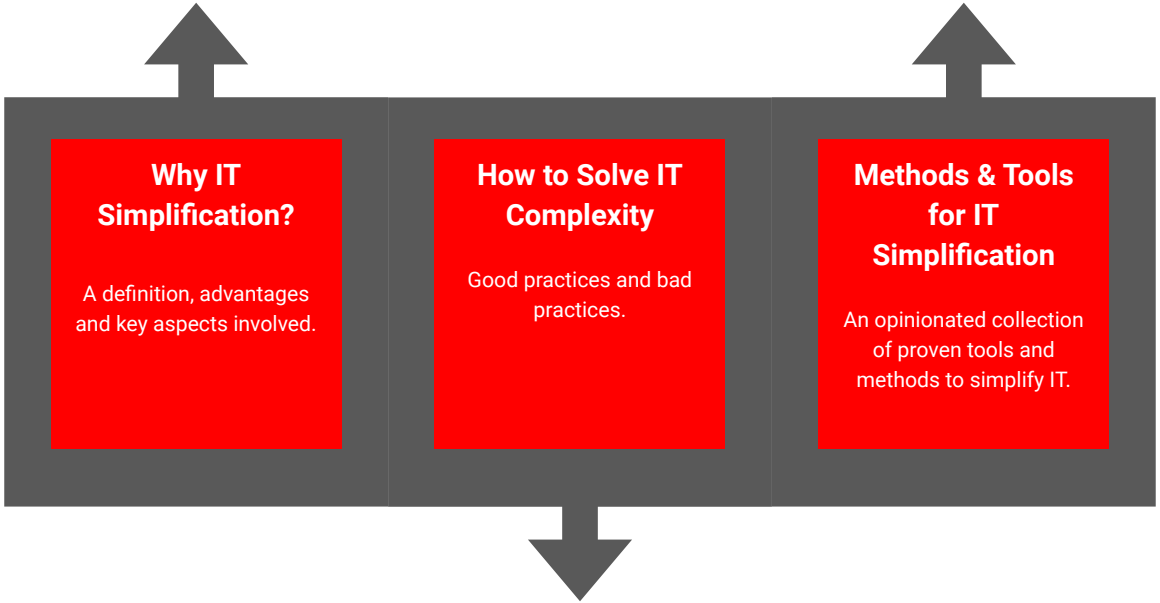
'*Simplify IT*' is about the art and science to simplify IT solutions. There is no such thing as 'simplicity science' that can be directly applied when dealing with IT problems. However, many insights of complexity science are applicable to solving real world IT problems. People love simple solutions.

However simple solutions are biased and context dependent. There are no generic, simple solutions for complex IT challenges. But there are ground rules for finding and implementing simple solutions.

The science of simplicity is interdisciplinary. It has roots at mathematics, computer science, social sciences, business science, natural science and philosophy.

Nobody ever asks for a complex IT solution. We all love simple IT solutions.

Outline



Why IT Simplification

Every company depends on IT. The dependency of IT is mostly visible when IT is not working. IT is just like air. You only notice your strong dependency when it is no longer there.

When your business growths, somewhere in time your valuable IT landscape, systems and applications, have become complex. Often too complex to control.

Alarming signals of IT complexity are:

- Difficulty in implementing new automatic business interfaces (APIs).
- A rising number of stability, cybersecurity, privacy, or even safety incidents.
- An increasing number for valuable resources is needed to maintain normal operations.
- New business products and ideas are impossible to realize and integrate within a reasonable timeframe.
- High switching costs. Switching costs are disadvantages and expenses when a new solution is needed.

IT Complexity is a slowly growing danger. Most IT (Information Technology) systems are nowadays very complex systems with many interacting components. An IT system is far more than software alone.

An IT system comprises more than just software. It is, at its core, a communication system that receives input and produces output. This system includes hardware, often remote components in a Cloud environment, sensors, software, networks, human activities, and automatically triggered processes. Additionally, all processes and activities necessary for operation and maintenance are integral parts of the system.

Complex IT and simple IT: A definition

An unified definition of what ‘Complex IT’ is does not exist.

The view on what complex IT is context dependent and is depended of human views.

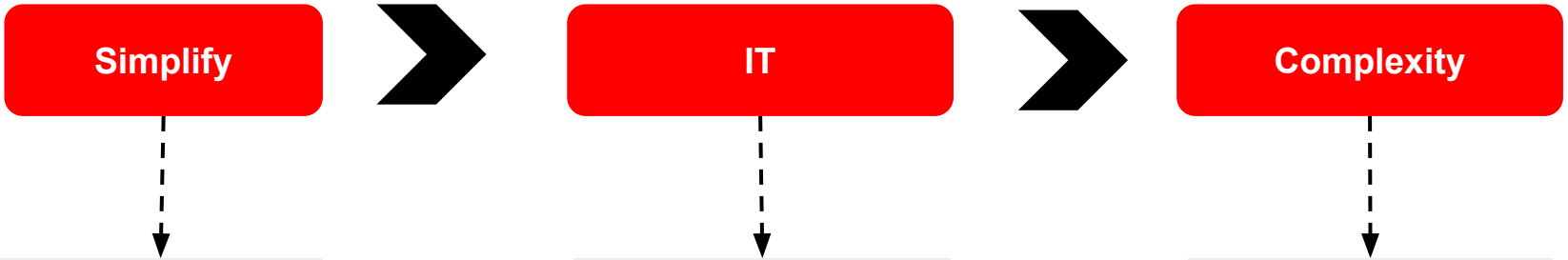
Simple IT solutions refer to the use of IT technology to solve problems and improve efficiency in a straightforward and uncomplicated manner.

IT complexity describes the complexity of IT landscapes, driven by the interdependent variety of its elements and the unsettling dynamic of continuous development. Variety and dynamic result in a diffusing perception that again lead to rising IT complexity.

Complexity is not ‘soft’ but is tangible and measurable:

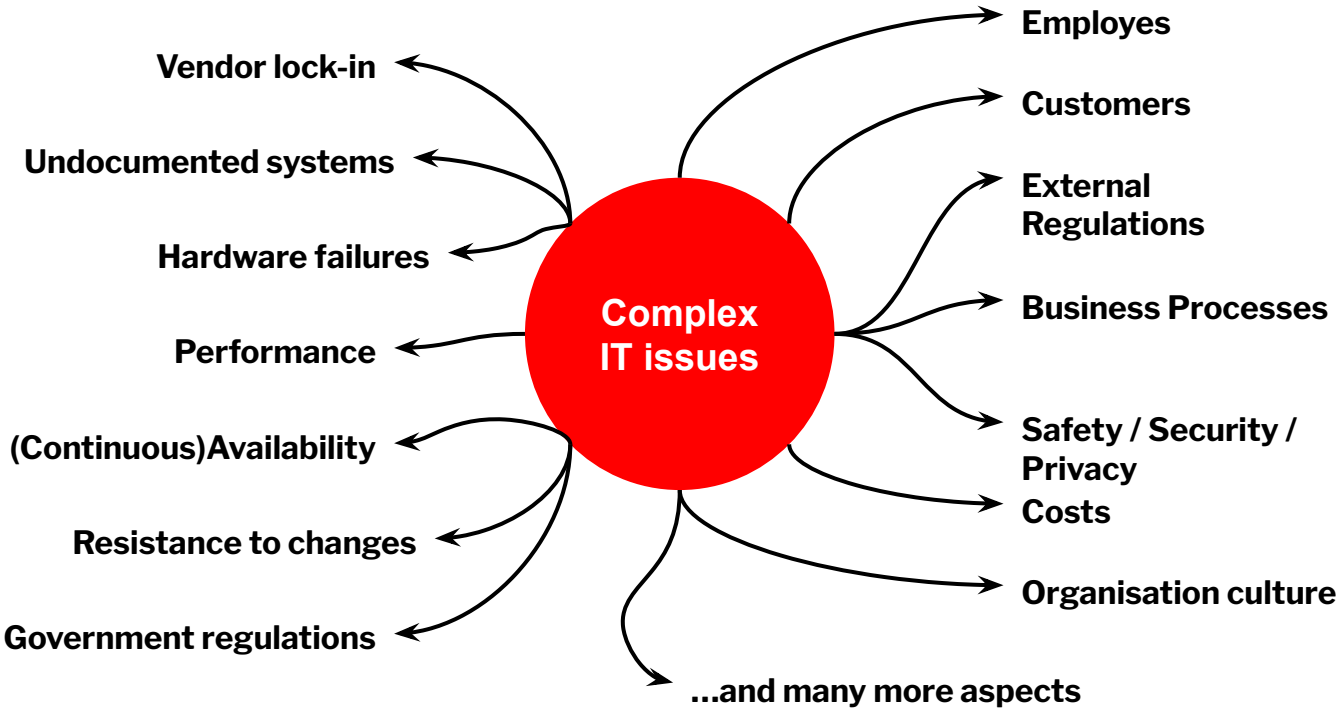
- Time needed for technical changes.
- Time needed for business process changes.
- Resources and cost needed for maintenance.
- Number of hardware and software elements and interdependencies.
- Number of internal and external stakeholders involved and/or effected by changes.

Simple is not easy



IT Simplicity requires bravery.
What are you willing to sacrifice to achieve simplicity?

Complexity of IT issues



Complex IT problems are multidimensional. Whenever you try to reduce complex IT problems you are on the verge of making a dramatic mistake because a simple solution can make the problem even worse.

Advantages of lowering IT complexity

- Lower cost. This applies to Capex and Opex costs.
- Faster decision making.
- Greater transparency.
- Easier alignment between business and IT.
- Shorten development times for new projects and changes.
- Higher quality IT products and IT steering processes.
- Reduce business risks.

IT simplification is challenging but in the end always profitable. But it is not easy or simple to do. All issues involved when striving to a simpler IT solutions or IT landscape require value judgments. Value judgements are biased. Always. This is not bad as long as open discussions are possible to qualify arguments. Experience is needed, as well as deep technical insights of both technology and deep insight on theories on business management principles.

Is reducing IT complexity really needed?

In essence IT complexity doesn't have to be bad . We all love complex technology as long as we have enough trust in the systems.

Trust often means that we or someone else can control and manage the risks involved. E.g. airplanes, medical systems, high speed railway systems and nowadays cars full of complex software.

Dealing with complexity is a matter of trust and accepting risks.

However when IT complexity causes severe problems or has huge risk profiles, it is time to take action.

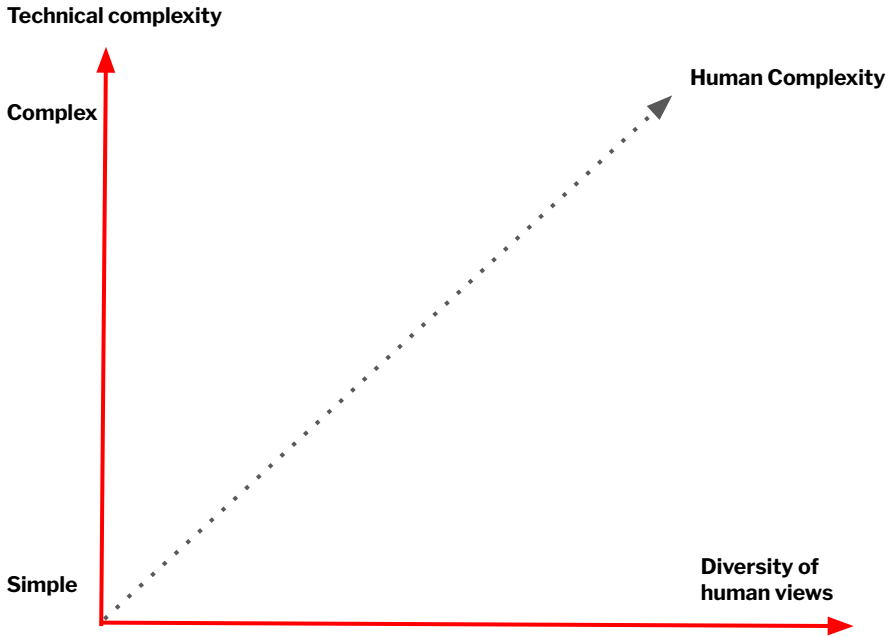
Simple or Complex IT problems?

Complexity within IT means a large number of elements (software, hardware, applications, systems, APIs) and many internal and external interactions.

Complex IT problems are known for multiple dimensions, like costs, divergence of values, dynamic non-linear behaviour and continues change.

Simplifying IT means dealing with humans with different *background*, *values* and *goals*.

People make technical IT solutions. But people make creating good and simple IT solutions hard!



“If you think technology can solve your problems, you don’t understand the problems and you don’t understand the technology.”

– Bruce Schneier, American cryptographer and information security author.

Methods & Tools for IT Simplification

Reducing your organization IT complexity is a known *severe* challenge. Using proven open scientific methods & tools that have been applied with success in the past will always help. Action is best done using systematic steps.

Barriers for finding simple IT solutions

Common barriers for finding simple IT solutions are:

- We humans are very bad at dealing and predicting non-linear behaviour with systems.
- The believe that your business problem is so unique that use or reuse of existing simple solutions is not possible.
- We are brainwashed by expensive IT consultancy and analyst companies on solutions that seems to have only major benefits but the negative drawbacks are seldom clear in advantage. The list of common falsehoods is long. To name a few:
    - Cloud solutions are cheaper.
    - Cloud security is always better than on premise.
    - Avoid using open source software solutions.
    - Centralized systems are always better.

There is never an absolute good and wrong. Scientific evidence, if available at all, is only provided for some specific use-cases. You should always look at your context and analyse your real problem and the root-cause(s) before jumping towards solutions. Distrust general statements about simple IT solutions. Simple solutions for complex business IT problems seldom exist.

What type of tools do really help?

IT complexity problems are very context specific. But on a higher level there are very good scientific frameworks for dealing and solving complex problems. More than 70 years of scientific research on complex problem solving, also known as problem solving methods, gives us a solid foundation on how to approach and solve IT complexity problems.

Refuse to use the latest hyped agile method, stay away from a magic (TM) tool from a consultancy firm. If a tool or method sounds too good to be true, it's just not true. But worse: You will end up with an even worse situation than before.

Reducing complexity within IT landscapes is managing trade offs:

- Fast versus slow.
- Expensive versus cheap.
- Robustness versus flexibility.
- Quality aspects incorporated versus purely functional.

There is no magic tool or method: There are however many *bad* practices, *failed* holy grails, but also some good principles to follow!

Method: Rethinking

Thinking different is crucial for simplifying IT. Consider rethinking the following aspects:

Centralized	vs	Decentralized	Centralized systems are conceptually simple and easier to analyse, build, operate and maintain. However decentralized systems are more robust. From a security and privacy perspective decentralized systems should always be taken into consideration.
Isolation	vs	Interaction	Isolation components and minimizing interactions is a standard way to manage complexity in large systems. But isolation comes with a price. More components can results in a more complicated system that is harder to keep stable and to maintain.
Top down	vs	Bottom-up	Most architects and managers use unconsciously a top down approach. A lot of knowledge for finding good and simple solutions is available by people who do the daily hands-on activities. But intuition and the natural resistance of humans to change should not be underestimated.
Analysis	vs	Computation	With the wide availability of good and efficient methods to perform calculations on assumptions, computation is key to finding best fit solutions. Quantify assumptions and qualitative aspects into a model and do some calculations. Qualitative analyse alone is not enough when challenging solutions
Design	vs	Reuse	Almost all business problems have been solved. Finding existing solutions starts with a good analyse of the problem you want to solve. An IT problem is never an IT problem, but always a business problem.
Objective	vs	Subjective	There is no objective truth on what a simple IT solution is. The truth about what a simple IT solution is depends on the people involved and the weights that are put on the different aspects.

Method: Balance advantages and disadvantages

Large IT companies have created many options that seem to reduce complexity. However when IT complexity is hidden you have to make sure the advantages are larger than the disadvantages. Using more software means introduction of more hard and soft dependencies, not only technical, but also dependencies with external vendors. Think e.g. on extra effort on security management and software license management. Advantages and disadvantages are never black and white. It depends on your current context and your future business strategy plans.

Advantages	Typical IT 'solution'	Disadvantages
Predictability,stability, efficiency and fixed operational costs.	**Software defined**	Increases dependencies of software experts, increases number of software building blocks and technical interfaces. Requires CI/CD automation which can become complex too.
Reduces resource costs, can reduce security risks, can increase productivity for developers.	**Virtualization**	Increases number of software building blocks, extra knowledge required.
No on premise software / hardware needed, use of standardized services, stability and scalability.	**Cloud**	Cost, security and increase vendor lock-ins. for operation with an external service provider.
Shorter realisation time when business and tool-users work together.	**Nocode / low-code**	Cost and increased vendor lock-ins. Business flexibility limited by tool capabilities.
Can increase business flexibility when a correct architecture is developed with business stakeholders.	**SOA / microservices**	Complex dependencies are transferred not removed. There is nothing wrong with a monolith IT system if it works good.

Don't solve problems you don't have. Be suspicious of IT hypes!

Complicated IT should not be confused with complex IT.
Prevent complicated IT from growing to complex IT.

Method: Simplify business processes

A simple business process leads to simpler supporting IT systems. However simplifying a business process is not simple. Too often complexity has grown by implementing countless software features to avoid hard choices to keep a business process simple and straightforward.

Method	Short description on IT impact
Remove options	Options in business processes and business activities create multiple complicating factors in software. E.g. a standard procedure for sending invoices, use one method for digital interaction with customers instead of many.
STP (Straight Through Processing)	Increased efficiency, no manual intervention needed. Requires handling of errors when automatic processing is not possible.
Standardization	Standardization of business processes results in less complicated systems and procedures.

Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better

— Edsger Dijkstra

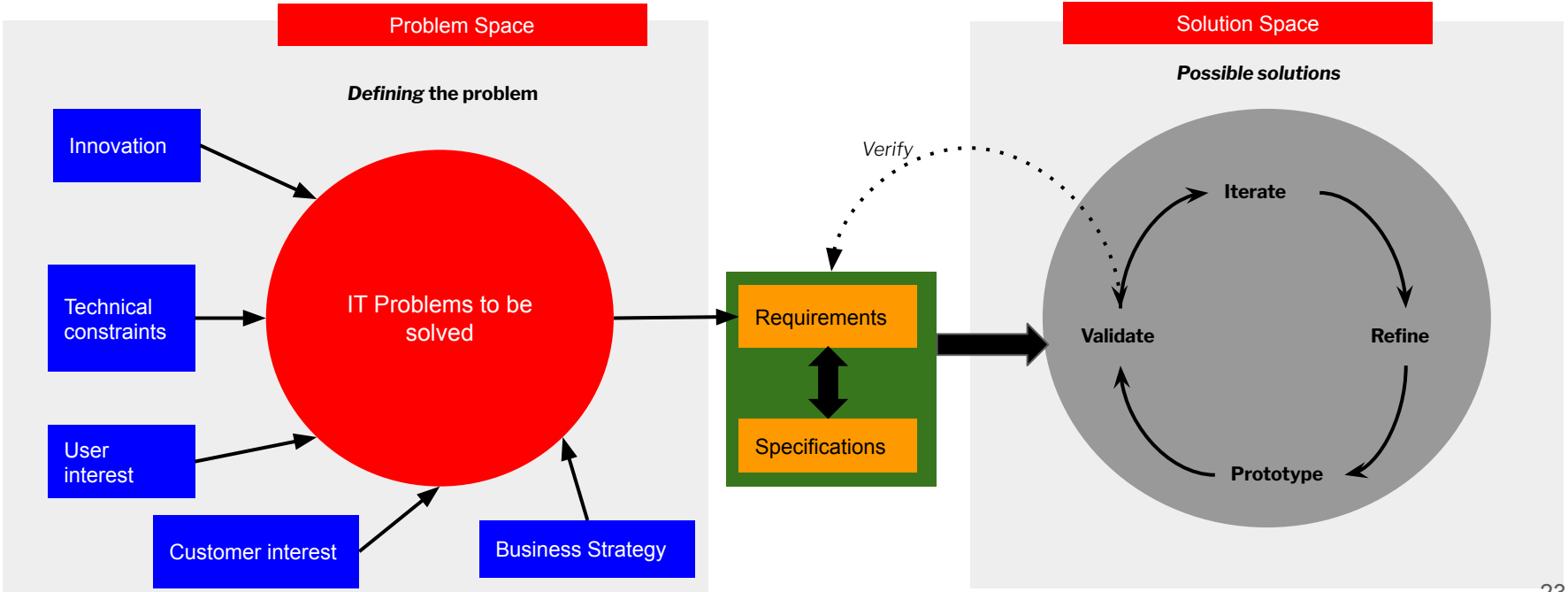
Method: Reuse software building blocks

Reusing a software building block is often better than creating or buying a new software building block with overlapping capabilities. However when it comes to simplifying IT reusing software means that there is unfortunately not one best way. Important factors that are context dependent play a major role. E.g. available resources, license cost (if applicable), maintenance effort, required business flexibility etc.

Method	Short description on IT impact
Copy software	Using a 'copy' means that the software and knowledge is reused, but dependencies on software level are not increased. This method is applicable if having new independent software objects is preferred but dependencies between business process should be minimized.
Share software objects	Sharing a software building block means it is shared during runtime. This can create extra software dependencies that should be managed. Often usable for a complicated building block that requires deep knowledge for operating.
Documentation	Documentation of building blocks is key to make reuse possible. Minimal APIs, dependencies and design decisions should be documented to make reuse possible.
Use a framework (middleware)	Using a generic framework for generic application functionality (e.g. UI, messaging or logging) means less specific business logic is needed.

Method: Create specifications in an iterative process

To solve complex IT problems connecting the problem space with possible solutions can be done in an iterative process. Within this process create specifications in a collaborative way in multiple iterations. Whenever possible create prototypes or 'Proof Of Concepts' of possible solutions. Every solutions has drawbacks. However creating software prototypes is check and ultime flexible. Make use of this core property of software to find the good fit. For IT you need agreement specifications that are based on requirements.



Method: Use a proven architecture approach

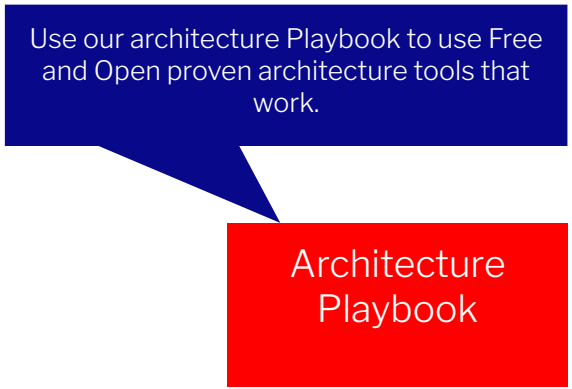
Architecture alone will not solve your problem. However having no architecture at all will certainly not solve your IT complexity problem. A good architecture saves time, money and prevents major failures. Advantages of using an architecture approach are:

- Sustainability
- Cohesion
- Maintainability

Many proven architecture methods exist. If possible, use always an international open method that is not (™) and the method is really open. There is no perfect architecture method. To simplify IT using a proven architecture methods and tools is a must. Develop IT systems without a business IT architecture is a high risk.

The essence of creating an architecture is creating a model to solve problems. Never model everything. Visualisation always helps.

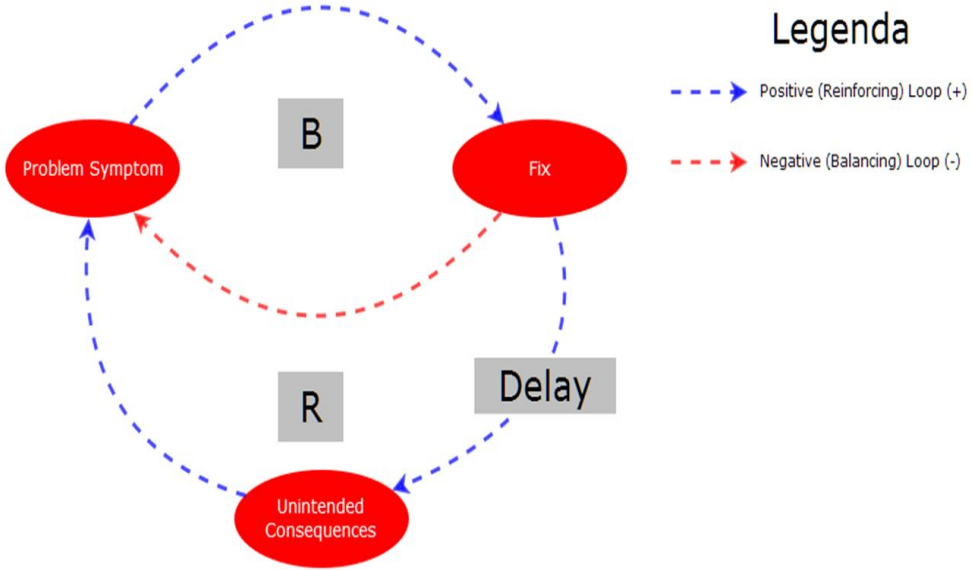
Complex IT systems require multiple different architecture views (models) that describe the structure and working. Without architecture visualisations there would not be no great buildings or very complicated IT systems that are maintainable and flexible.



Tool: Cause and Effect diagrams

Using causal loop diagrams helps you manage and solve complexity issues. Causal Loop Diagrams (CLDs) helps with visualizing and understanding complex systems and relationships. To be effective for solving complex IT problems you should discuss and create CLD with your key problem stakeholders. CLDs promote discussion and mutual understanding. A key aspect with all solutions is that there are always side effects. Also CLDs make clear how other stakeholders see relationships.

Use our friction free Causal Loop Diagram Tool without cost to create your CLD now!



Creating CLDs brings clarity on how people see IT complexity problems and possible fixes.

CLDs have the advantage to stimulate deeper discussions on ‘how’ and ‘why’. Hidden assumptions of different stakeholders are made explicit.

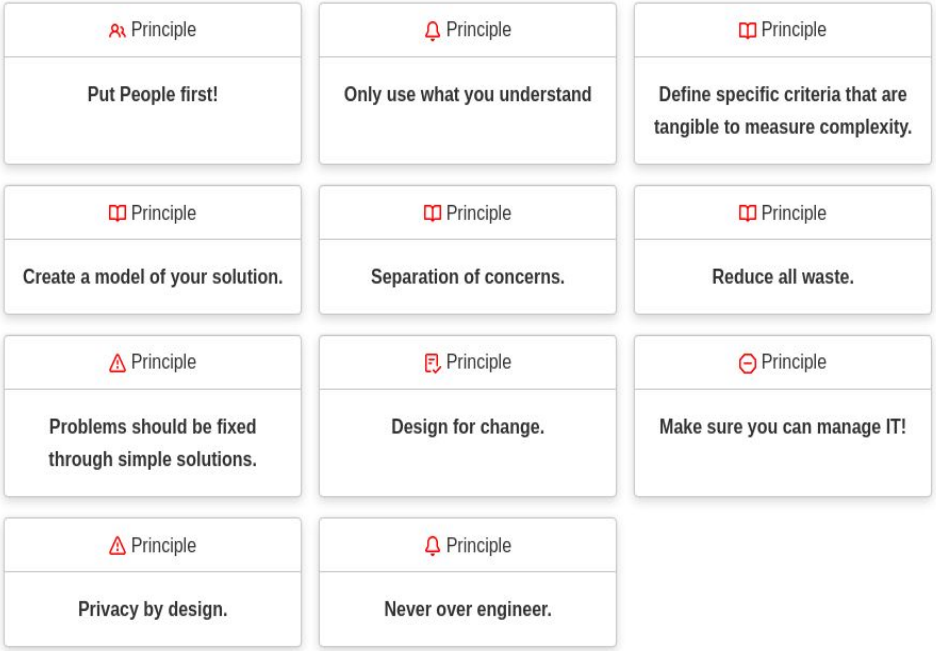
Tool: Use principles to steer development

Principles help to prevent complicated IT systems growing towards unmanageable complex IT landscapes. Principles are statements of direction that govern selections and implementations. Principles provide as a foundation for decision making.

It is important to make a clear differentiation between standards, requirements, and principles:

- *Standards* are “musts”. They must be met for compliance (e.g. for law, security or safety standards).
- *Requirements* articulate specific needs that must be met by a specific solution.
- *Principles*, on the other hand, are more general and serve as a framework for making choices by providing guidance about the preferred outcome of a decision in a given context.

The 0CX (zero-complexity) standard defines architecture and design principles to prevent complexity for new products.



Use our free 0Complexity principles Guide to Simplify IT now!

In order to solve complex IT problems, we need adequate knowledge. Different problems require different models of the same reality, without any one being the "true" representation.

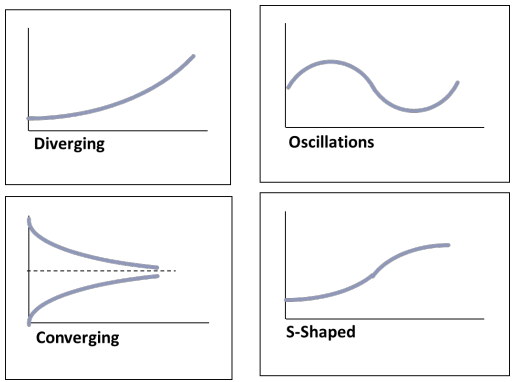
Tool: System Dynamics

System Dynamics (SD) is a method to understand, manage or simplify complex IT systems characterized by:

- nonlinear dynamics
- feedback
- time delays
- soft factors
- interdisciplinary aspects

Key tools within System Dynamics are stocks-flows models,feedback or feedforward loops, and time delays. We humans are *very bad* in predicting non-linear behaviour and curves. With simple (FOSS) software you can create a quick crucial overview to solve complex IT problems.

Feedback loops are linked to specific kinds of basic behaviour patterns:



Finding the archetypes (feedback loops) in non-linear systems is very hard. (=for us humans). So use a tool!

Our recommendation:

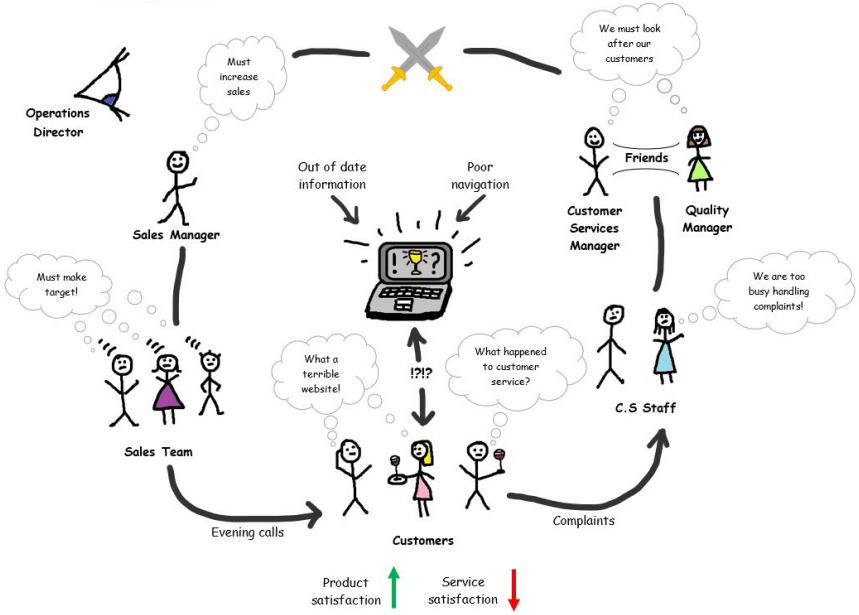
Use Insight Maker to create rich pictures and causal loop diagrams. Then create shareable simulation models, all right in your browser, for free.

Tool: Rich Pictures

Description: Rich pictures provide are a tool to use learn about complex IT problems before diving into discussions on solutions. Key is to get more insights and learn on all aspects involved. Rich pictures are great simple tool to use for ill-defined problems since it is just drawing without rules. Rich pictures follow no commonly agreed syntax and can consist of symbols, cartoons, sketches or some words to make the picture easier to understand.

How to use it: Some guidelines for creating Rich Pictures when solving complex IT problems.

- Use some form of structure.
- Do something with a process that is deeply involved with the problem.
- Incorporate objections or concerns. Use for example,a question mark or text balloons.
- Stick closely to the language and imagery that resonate within the organization for which the illustration is intended.
- Use text, figures, sketches, or colours as you see fit to create a visual representation.
- Draw by hand if possible. This facilitates quicker discussions and conveys the essence more rapidly. It also helps maintain complexity and encourages others to join in more readily.



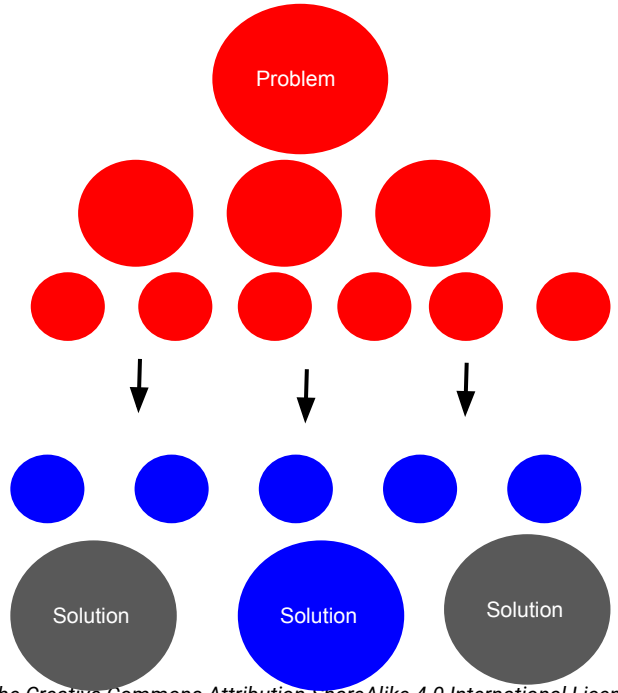
Tool: Mental models

What is it: Mental models are explanations of how something works in the real world. They are representations of the surrounding world, the relationships between its various parts and a person's intuitive perception about their own acts and their consequences.

How to use it: Take your problem and 1) break it down into its fundamental parts and 2) reconfigure those parts to build a solution.

Some core mental models:

- The Map is Not the Territory. The map of reality is not reality. Even the best maps are imperfect.
- Circle of Competence. We all have blind spots.
- Reasoning from first principles. A way to reverse engineer complicated situations.
- Thought experiments. Evaluate the potential consequences and re-examine to make better decisions.
- Inversion. Instead of thinking of ideal solutions, think of bad solutions and ask yourself, "How might we avoid these?"



Tool: Soft Systems Methodology

Description: Soft systems methodology (SSM) is an organised way of tackling perceived problematical (social) situations. It is action-oriented. It organises thinking about such situations so that action to bring about improvement can be taken.

How to use it: The SSM stages are intended to be iterated and have no strict order.

1. The problem situation unstructured
2. The problem situation structured.
3. Define Root definitions of relevant systems.
4. Create Conceptual models.
5. Comparison of stage 4 and stage 2 outcomes. So compare models and reality.
6. Identify feasible and desirable changes
7. Take action to implement changes the decrease observed IT complexity. Transformation must be judges by three ‘E’s:
   - ➔ 1) Efficacy (does the it work?)
   - ➔ 2) Efficiency (resources used) and
   - ➔ 3) Effectiveness (Meeting the longer term aim?)

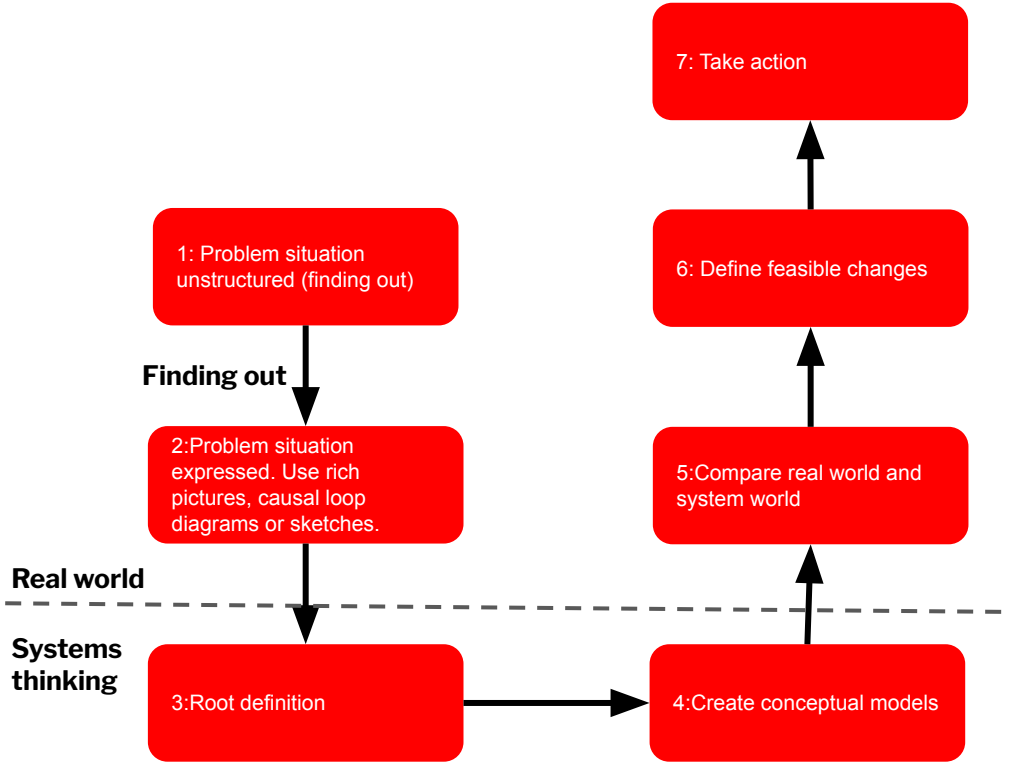
Tool: Soft systems methodology (2)

The root definition can be created by using the mnemonic CATWOE:

- Customer(s) (those who benefit)
- Actors (those who perform)
- Transformation process (f X ? Y)
- World View (point of view on the process)
- Owner (who owns the systems)
- Environmental constraints.

Key SSM features:

- Goal driven.
- Iterative process.
- Process is more important than the outcome.
- SSM is done by the people in the organization, along with a facilitator



Tool: Option matrix

Description: An option matrix, also known as a decision matrix or criteria matrix, is a tool to evaluate and compare options against a set of criteria.

How to use it: Creating an option matrix requires 3 steps:

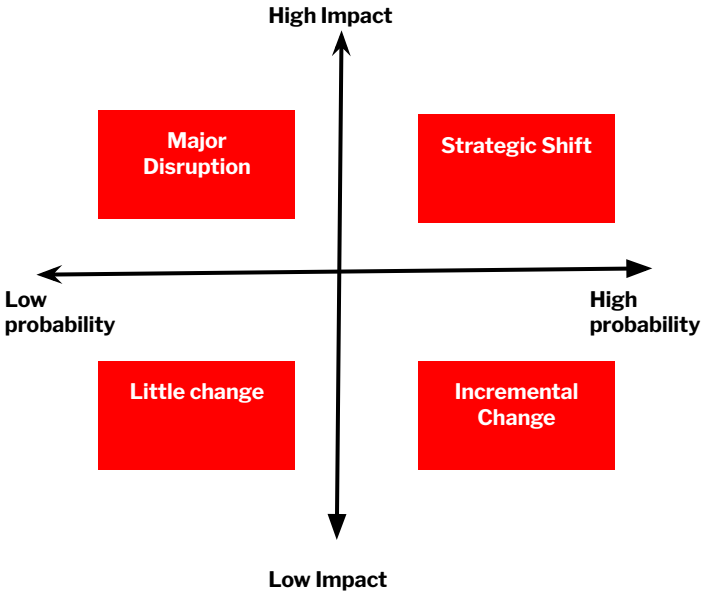
1- Reformulate your problem from a binary choice to a high to low scale.

2- Introduce a second variable that also has a scale from high to low.

3- Draw the matrix. And put options in every quadrant of the matrix.

Now evaluate each quadrant of the matrix to get deeper insight in each scenario. In fact each matrix quadrant represents a scenario.

Do not fall in love with the matrix. It is good to create a couple of matrices with different axis to evaluate what other scenarios exist when key variables on the axis are changed.



Tool: RACI matrix

Description: A RACI matrix is a tool that defines roles and responsibilities as Responsible, Accountable, Consulted, and Informed in a process.

How to use it: A RACI matrix helps to find the needed stakeholders for simplifying IT.

The key responsibility roles in the RACI matrix are:

- *Responsible* : Do the work to complete the task.
- *Accountable*: Must sign off (approve) work that responsible provides .
- *Consulted*: Those whose opinions are sought, typically subject-matter experts, and with whom there is two-way communication.
- *Informed*: Those who are kept up-to-date on progress, often only on completion of the task or deliverable, and with whom there is just one-way communication.

Task	IT Manager	Business Manager	Engineer (process or software)	Architect	CIO
Redesign process	C	R	R	C	C/A
Remove software dependency	I	I	R	C / I	C
Simplify UI interface	R	C	R	C	I
Implement changes	R	A	R	I	A

Creating a RACI matrix brings clarity of roles and responsibilities and facilitates efficient decision making when solving IT problems in larger organisations.

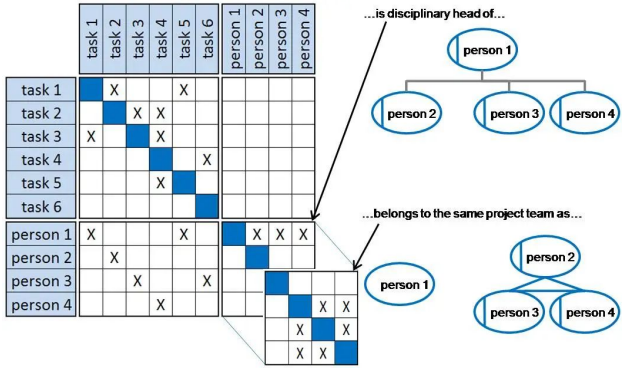
Tool: Design Structure Matrix (DSM)

Description: The Design Structure Matrix (DSM) is a simple tool to perform both the analysis and the management of complex IT systems. It enables the user to model, visualize, and analyze the dependencies among the entities of any system and derive suggestions for the improvement or synthesis of a system.

How to use it: The DSM method can be used in various ways. Some simple steps to create a Design Structure Matrix:

1. Define the system and its scope. What is the system that you are trying to represent? What are the boundaries of the system?
2. List all the system elements. What are the individual components or parts of the system?
3. Study the information flow between system elements. How do the elements of the system interact with each other?
4. Complete the matrix to represent the information flow. Use a 1 to indicate a dependency between two elements, and a 0 to indicate no dependency.
5. Give the matrix to the engineers and managers to comment on and use. The DSM can be used to identify critical elements, analyze robustness, find bottlenecks, and optimize the system.

More information on DSM methods and tools.



Tool: Heat Mapping

Description: A heatmap is a visualization technique that shows range of different perspectives.

How to use it: To use a heat map to manage IT complexity, you need to identify the factors that contribute to the complexity in your organisation. These factors can be technology diversity, data volume and variety, and organizational structure and culture. Next select perspectives e.g.:

- Maturity.
- Effectiveness.
- Performance.
- Maintenance effort.
- Cost (opex or capex).

Different colors of each capability gives a quick visual overview on a capability map. To keep it simple you should use the colors like a stoplight metaphor (red, yellow or green). Heatmaps are useful for quickly and easily understanding complex data, identifying patterns and trends, and comparing different categories or values. Be careful when presenting heatmaps since they can also be misleading since context is important and business heatmaps are never neutral but biased.

Example heat map:

Technology maturity	**Software maintenance**	**Security risks**	**Open Standards**
Cost performance	**Opex**	**Capex**	**Hidden cost**
Effectiveness	**Organisational culture**	**Organisation structure**	**Innovative**

About

Maikel Mardjan:

- Architecture & Design of complex IT systems.
- 25+ years work experience in the IT Industry.
- Master (MSc) Business Studies of University of Groningen.
- Master degree (MSc) Electrical Engineering, of Delft University of Technology.
- ...still likes to do hands-on programming (C/C++, Java, Python, PHP,JS,GOlang etc) to learn, make and break things!

I love solving complex IT challenges.

THE Open GROUP
TOGAF™ 9 Certified

TU Delft Delft University of Technology

CISSP® Certified Information Systems Security Professional

university of groningen

Appendix: More tools and method

The number of scientifically substantiated methods and tools that can be applied to simplify IT landscapes is enormous. Most tools and methods that are derived from a system approach have a good balance between hard aspects (like technology and software) and soft aspects (like humans and business goals). The following tools & methods can also be usable to simplify IT:

- Triz: Theory And Innovative Problem Solving - An approach for systematic inventive problem solving.
- 8D Problem Solving Process: Structured problem-solving methodology designed to find out the root causes of a problem
- Lean Thinking: A methodology that aims to eliminate waste and increase efficiency in processes and operations.
- Pareto Analysis: A technique for prioritizing problems or tasks based on the principle that a small number of causes are responsible for the majority of effects.
- Agile methods (like SAFe, LeSS, Scrum-at-Scale, DAD, and the Spotify model): Steering to solve or avoid IT complexity is still no guarantee.

I am always curious to learn about new problem-solving methods that are applicable to solving complex IT problems.

Disclaimer: No warranties are given. Solving IT Complexity is not simple and depends on you!

The text in this publication is written by a human.

Trademarks, trade names, product names and logos appearing in this publication are the property of their respective owners.

Feedback

Please send me a DM on X at @maikelmardjan or send an email to simple [at] nocomplexity [dot] com if you have any feedback on this slidedeck. I like to extend this publication with your favourite tools or methods to simplify IT complexity.

This Simplify IT slide deck is created to be used, shared and expanded with your input!

So I created this slide deck using the Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license.

We need to solve and prevent IT complexity problems. Too many projects fail, become unmanageable or introduce severe safety & security risks.

You are free to:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material for any purpose, even commercially.

If you do use or reuse content of this slidedeck I would appreciate attribution.

The simplest way to give attribution to this work is by including the following line in your derived publication:

Attribution Suggestion:

NoComplexity.com, SimplifyIT by Maikel Mardjan, licensed CC BY-SA 4.0.