

Architecting for Business Value

by *Gerrit Muller* USN-NISE and TNO-ESI

e-mail: gaudisite@gmail.com

www.gaudisite.nl

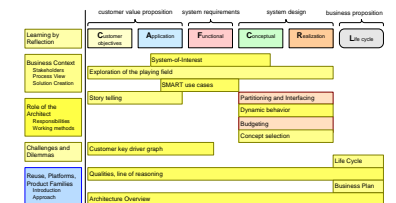
Abstract

The course Systems Architecting Fundamentals is a step toward a MOOC-based course. The number of slides is limited to fit into video recordings of 7 minutes or less.

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

January 27, 2026
status: preliminary
draft
version: 0.6



Architecting for Business Value; Introduction

by *Gerrit Muller* [TNO-ESI, University of South-Eastern Norway]

e-mail: gaudisite@gmail.com

www.gaudisite.nl

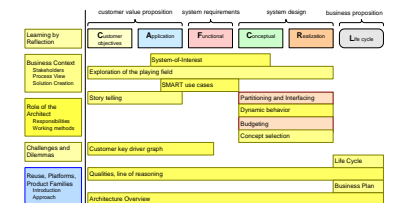
Abstract

This presentation introduces the ideas behind the course Architecting for Business Value. The course positions the architecting process in the context of the business processes. It explains the role of architects. The core of the course is the CAFCR+ architecture framework.

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

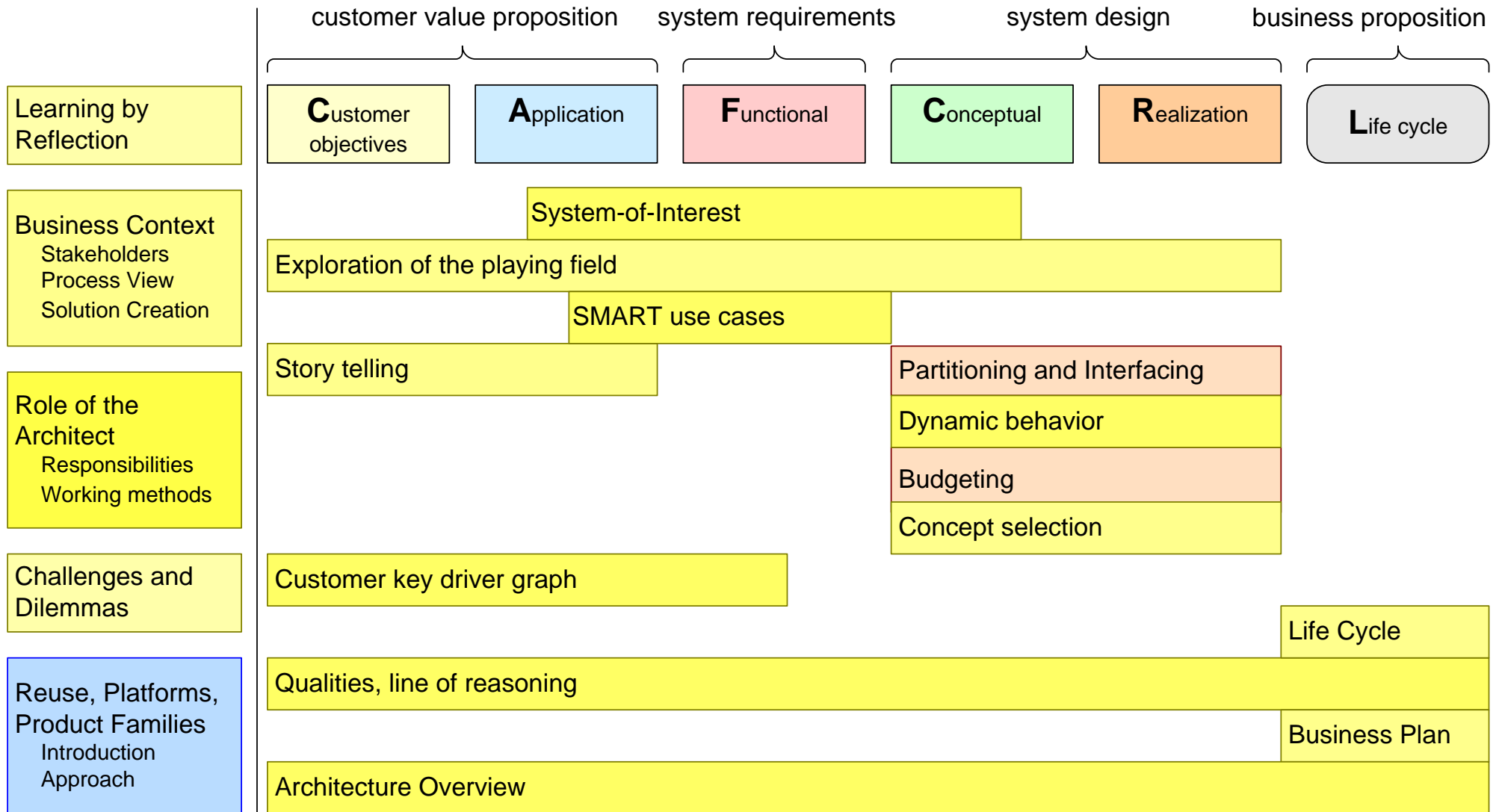
January 27, 2026
status: planned
version: 0



Course Nuggets

Introduction	Architecture Framework - Introduction, CAFCR+, Time Boxing, Iteration	CAFCR+ Iteration - Life Cycle
Learning by Reflection	Architecture Framework - Exploration of the playing field	CAFCR+ Iteration - Qualities, Line of Reasoning
Business Context - Stakeholders	CAFCR+ Iteration - Black Box; SMART, use case-based	CAFCR+ Iteration - Business Plan
System of Interest	CAFCR+ Iteration - Story telling	CAFCR+ Iteration - Architecture Overview, Threads of Reasoning
Business Context - Process View	CAFCR+ Iteration - Partitioning and Interfacing	
Business Context – Solution Creation	CAFCR+ Iteration - Dynamic Behavior	Reuse, Platforms, Product Families - Introduction
Role of the Architect - Deliveries, Responsibilities, Activities	CAFCR+ Iteration - Budgeting	Reuse, Platforms, Product Families - Approach
Role of the Architect – Viewpoint Hopping	CAFCR+ Iteration - Concept Selection	Wrap-up
Role of the Architect – T-shaped	CAFCR+ Iteration - Customer Key Driver Graph	
Challenges and Dilemmas		

NuggetMap



Architecting for Business Value; Assignments

by *Gerrit Muller* [TNO-ESI, University of South-Eastern Norway]

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

Abstract

All assignments of the course Architecting for Business Value.

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

January 27, 2026
status: planned
version: 0

logo
TBD

Determine the system of interest

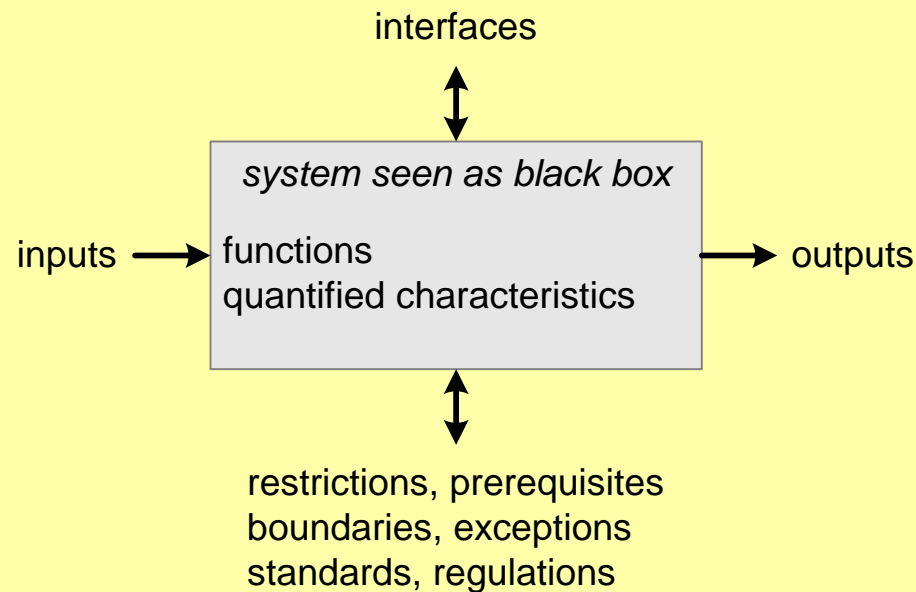
Define your organization

Determine an innovative change to be architected

Exercise SMART KPPs and Use CAse

Make specification overview with ~10 **SMART** Key Performance Parameters (or functions or interfaces)

determine at least one use case



use case
typical use with relevant
context data (quantified!)

- **S**pecific *quantified*
- **M**easurable *verifiable*
- **A**chievable (Attainable, Action oriented, Acceptable, Agreed-upon, Accountable)
- **R**ealistic (Relevant, Result-Oriented)
- **T**ime-bounded (Timely, Tangible, Traceable)

Exercise Story Telling

Create a story

as text + sketch or as cartoon

Use the criteria

be highly specific!

envision the future value proposition

Enjoy!

Exercise Block Diagram

Make a set of **block diagrams** capturing the **static parts** and **interfaces**.

Ensure coverage of the entire system, e.g. including service, training, production, etc.

Show both **hardware** and **software**

Good block diagrams have in the order of 10 to 20 blocks

Exercise Dynamic Behavior

Capture the **dynamic behavior** of the **internals** of your system in **multiple** diagrams.

Diagrams that capture dynamic behavior are among others:

- Functional flow (of control or information, material or goods, or energy)
- Activity or sequence diagrams (e.g. with “swimming lanes”)
- State diagrams




Make a **technical budget** for one of the **key performance parameters**.

- a good budget has 20 to 30 contributing elements
- elements should be balanced (remove or combine insignificant contributions)
- use the previously defined parts and dynamic behavior

Exercise Concept Selection

Make a **decision matrix** for one of the **concept selections**.

- define at least 3 concepts
- define 7 to 10 criteria for selection
- score the concepts against the criteria, for example using a scale from 1 to 5: 1 = very poor, 5 = very good
- recommend a concept with a rationale

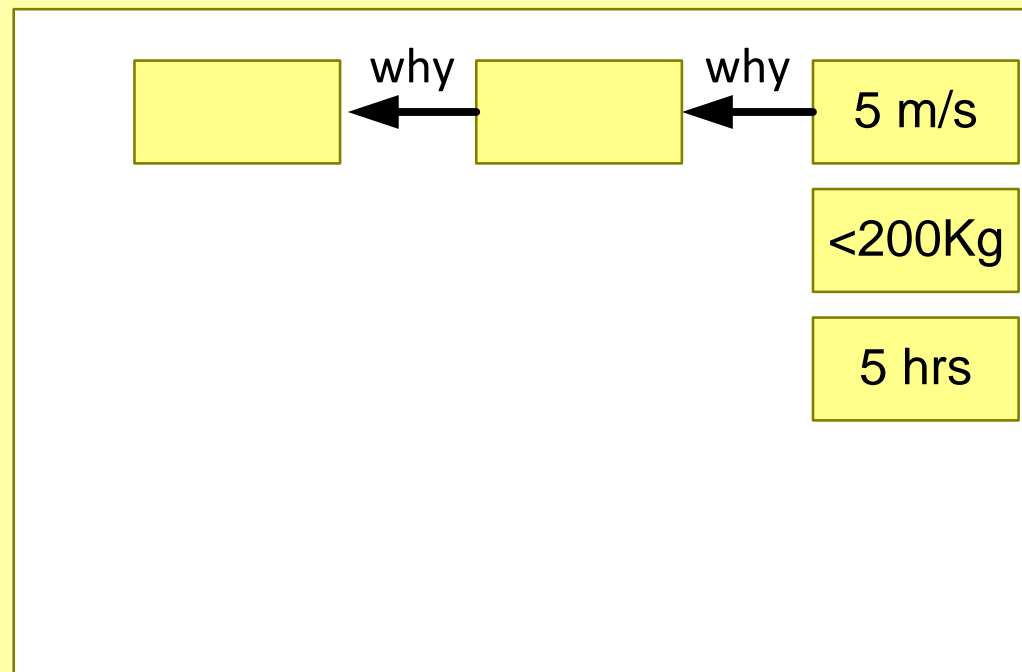
	concept 1 	concept 2 	concept 3 
critterion 1	vulnerable MTR 1	robust 30 hours 3	robust access 5
critterion n	reusing platform 4	reusing platform 4	resource shortage 2
			best, because ...

Exercise Customer Key Driver Graph

Make a **customer key driver graph**

Use yellow note stickers

Start at the right hand side



Analyze the **evolution** during the **lifecycle**.

- identify sources of change in customer context, life cycle context, and technology
- make a list of changes
- determine per change the expected rate of change and the required response time to the change
- optional: determine effort, impact, and risks per change

Exercise Line of Reasoning

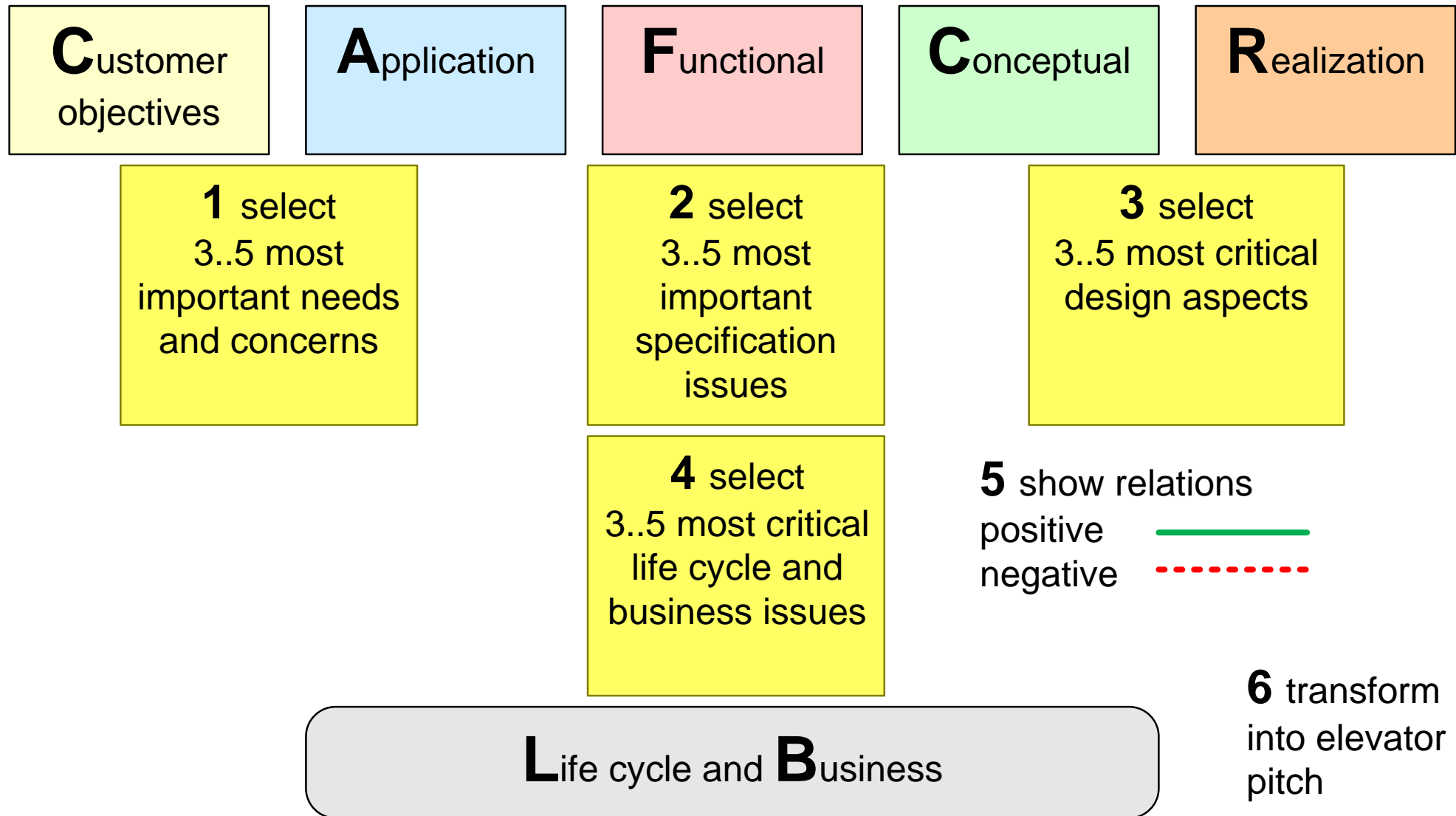
Make a **line of reasoning** for one of the dominant qualities.

- in the CA views; determine what customers do to achieve their goal
- in the F view determine the specification of your system supporting this quality
- in the CR views determine the relevant concepts and technologies
- Take the reverse viewpoints as well: what threatens this quality?

Make a **business plan** for the mid to long-term future.

- determine business model
- determine investments, sales volume, sales price, and costs
- estimate the cash flow and accumulated profit
- include at least 3 releases or generations of systems

Exercise Threads of Reasoning



Architecting for Business Value; Learning by Reflection

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]

e-mail: gaudisite@gmail.com

www.gaudisite.nl

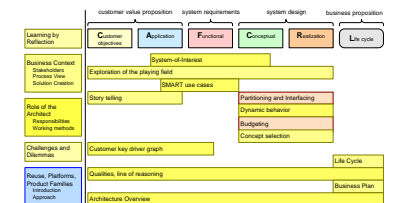
Abstract

This presentation explains the relevance of using reflection for learning.

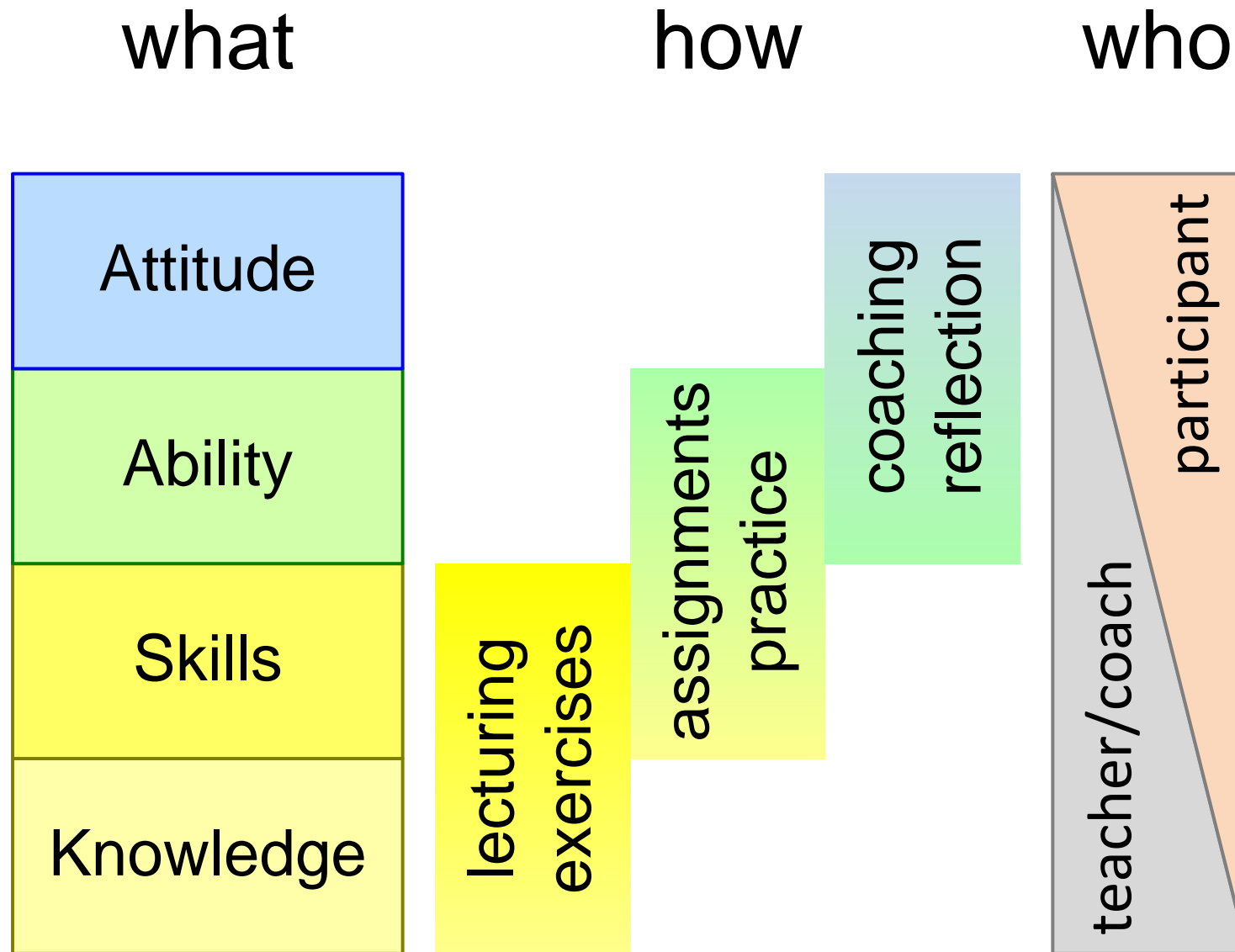
Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

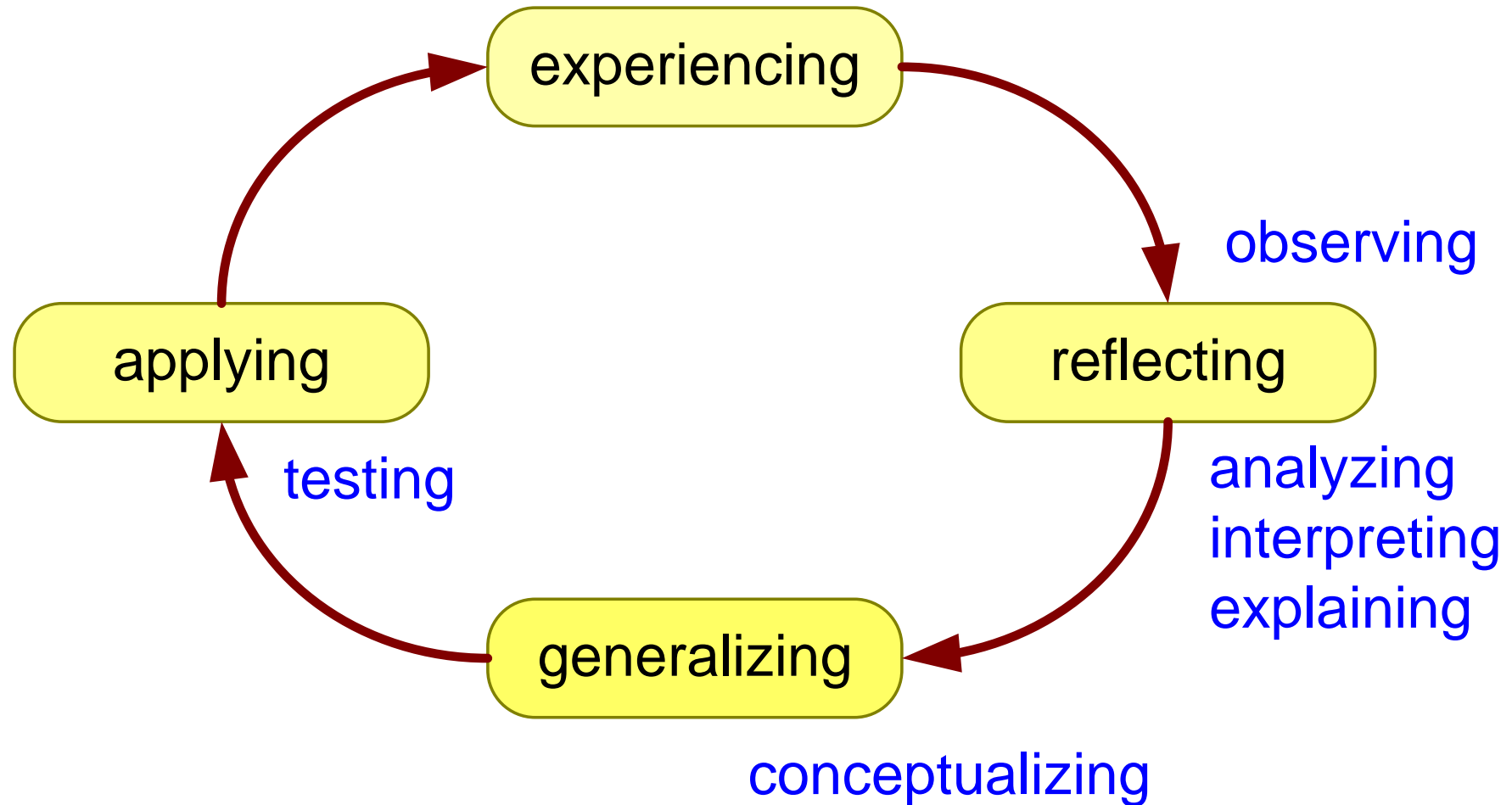
January 27, 2026
status: draft
version: 0



Competence Program Partitioning



Reflection Cycle



source: Kolb's learning cycle

<http://www.infed.org/biblio/b-explrn.htm>

When to Reflect

Reflection
Before
Action

*anticipation
preparation*

Reflection
In Action

concurrent

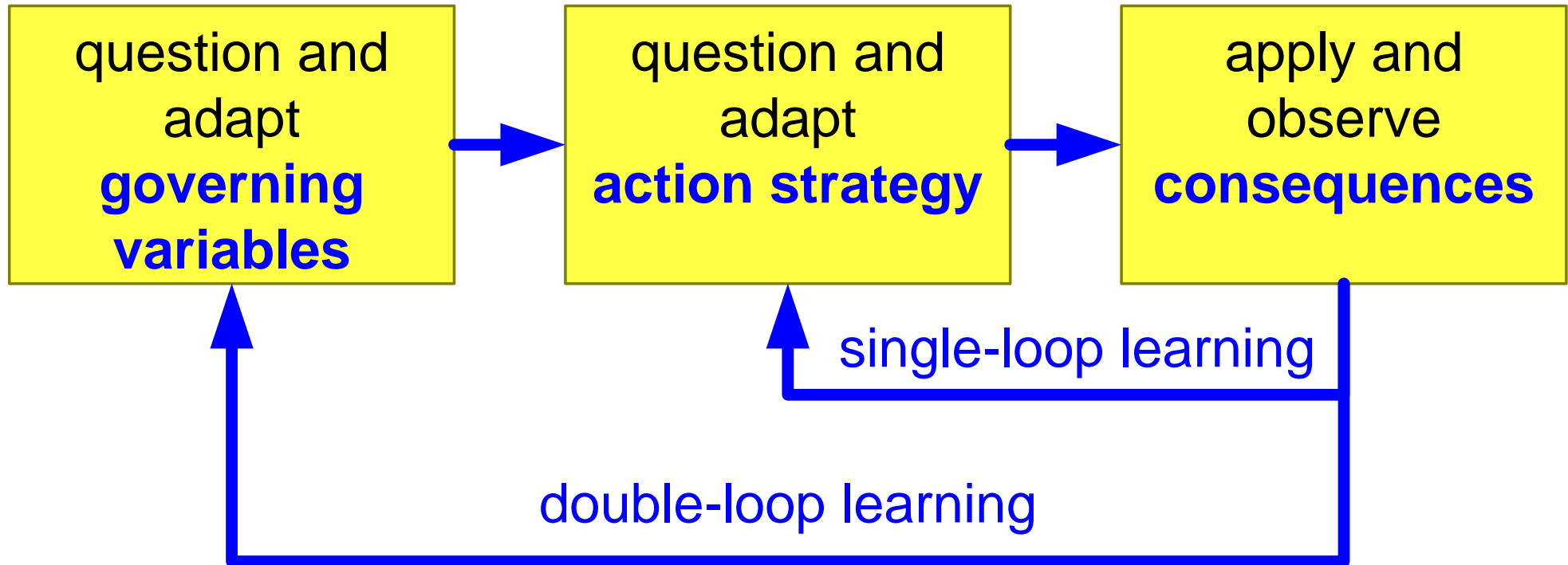
Reflection
On Action

retrospective

action

— time —>

Double Loop Learning



after C. Argyris

Architecting for Business Value; Business Context - Stakeholders

by *Gerrit Muller* [TNO-ESI, University of South-Eastern Norway]

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

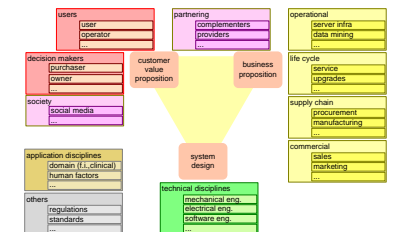
Abstract

This presentation introduces the various stakeholders and their relation to product innovation and architecting.

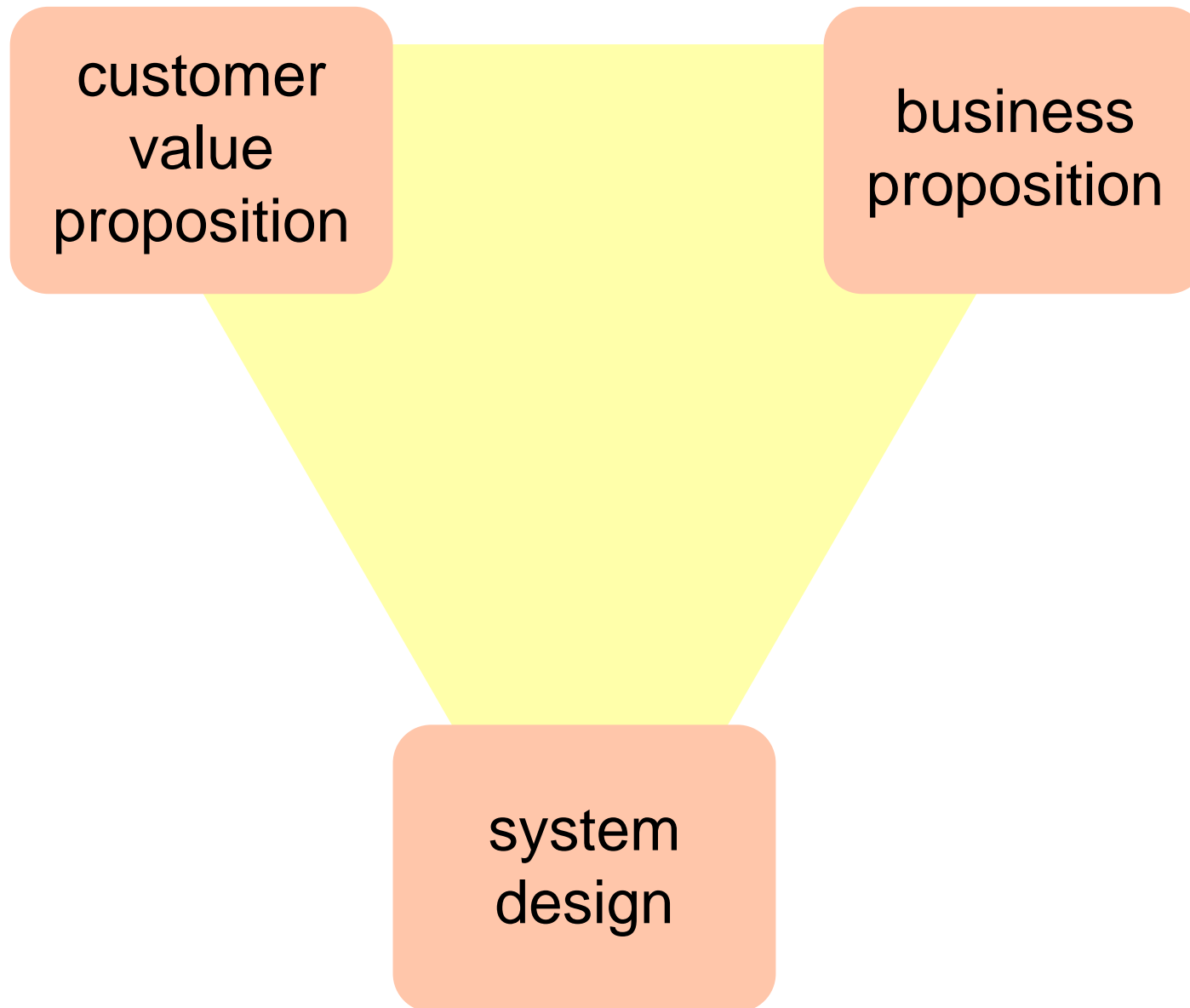
Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

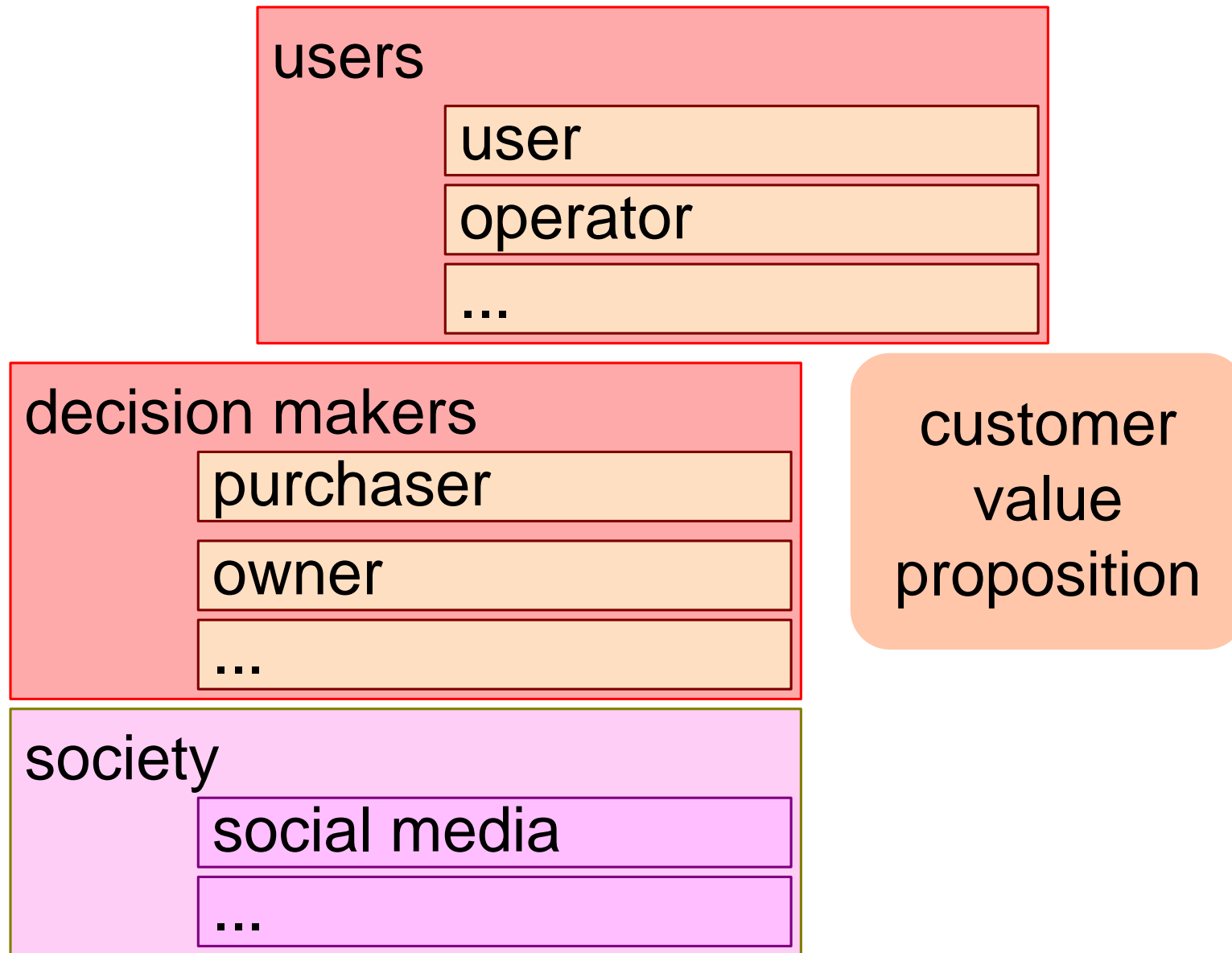
January 27, 2026
status: draft
version: 0



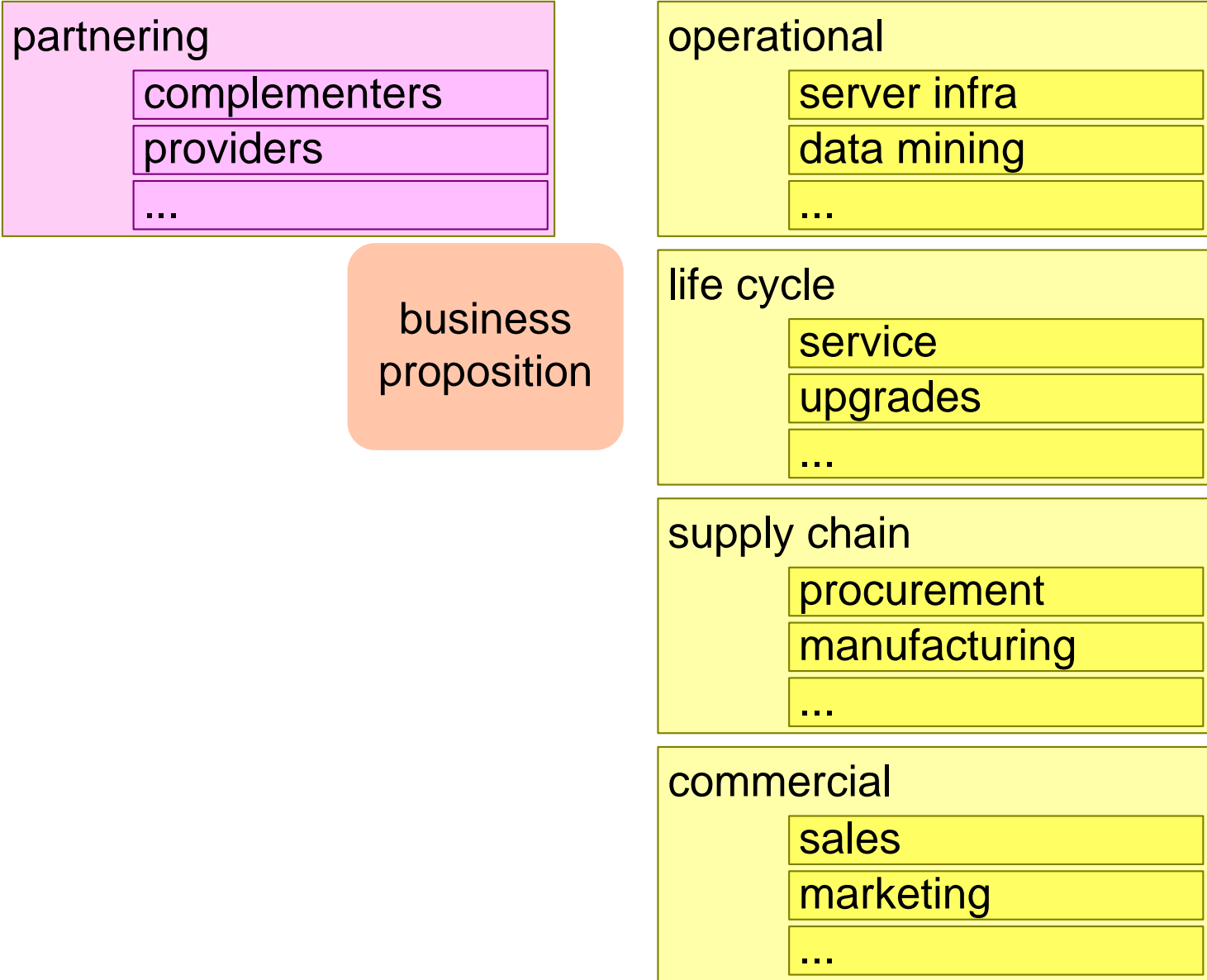
Top View



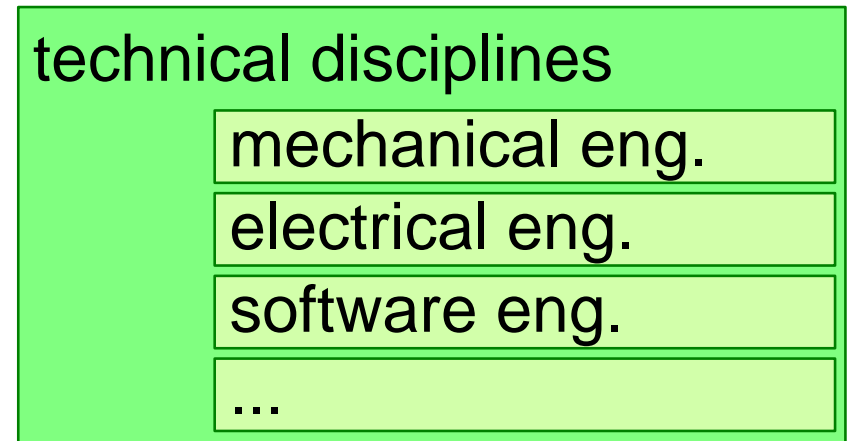
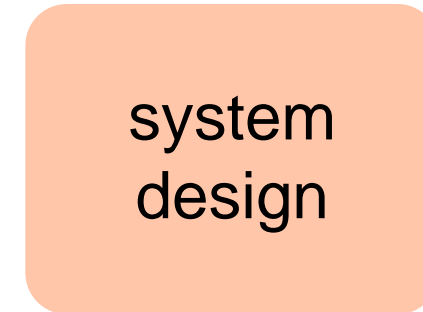
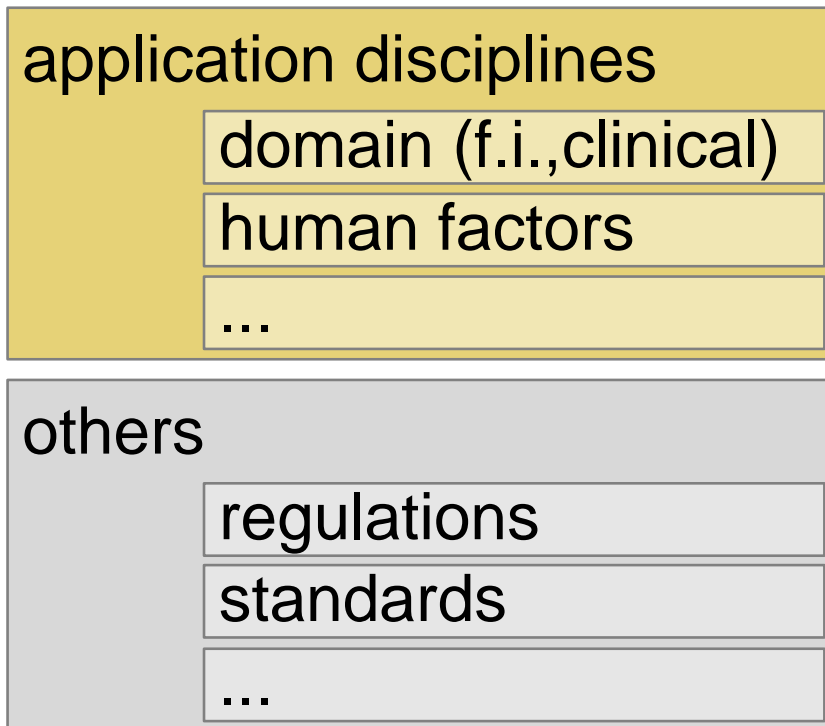
Customer Stakeholders



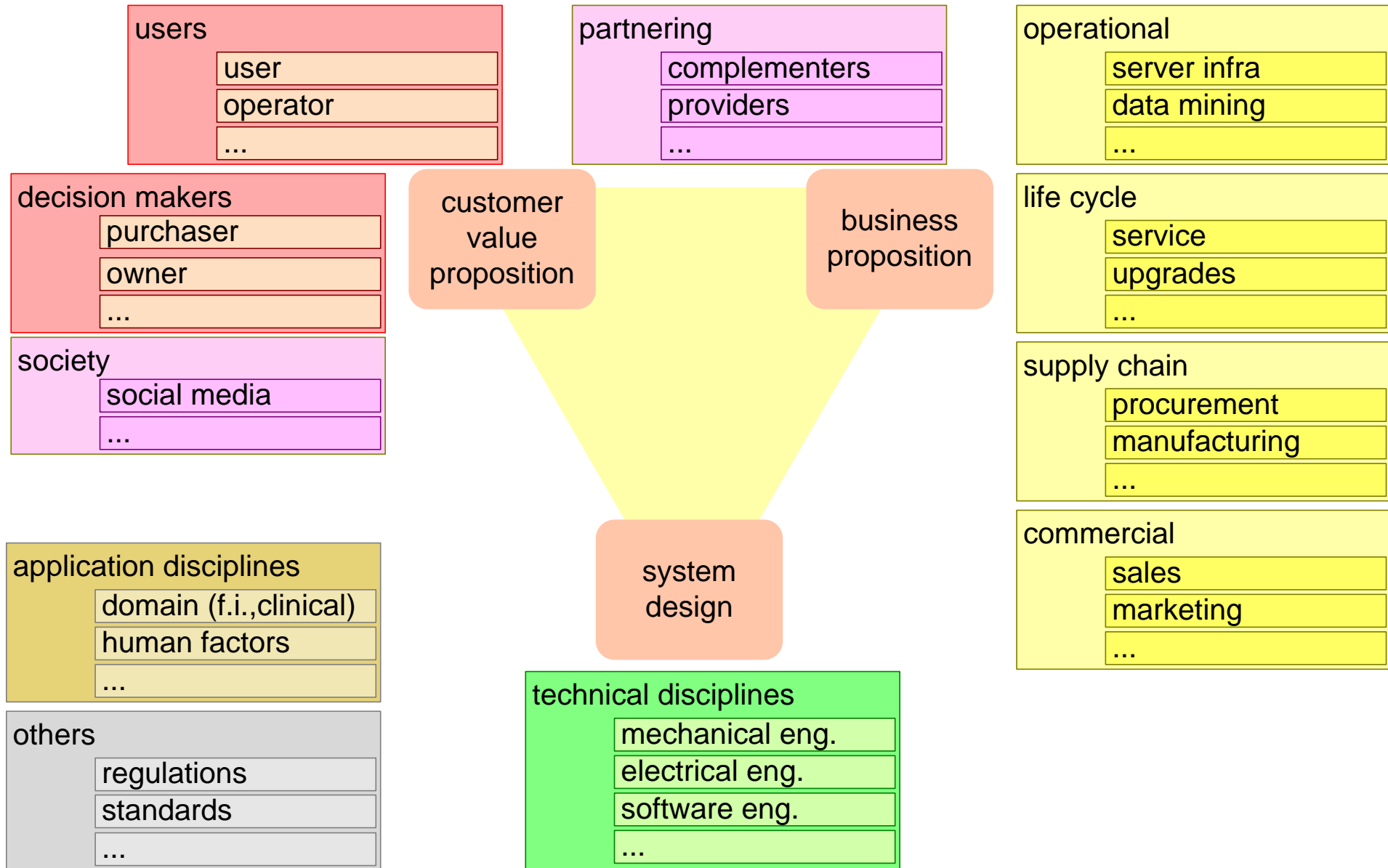
Business Stakeholders



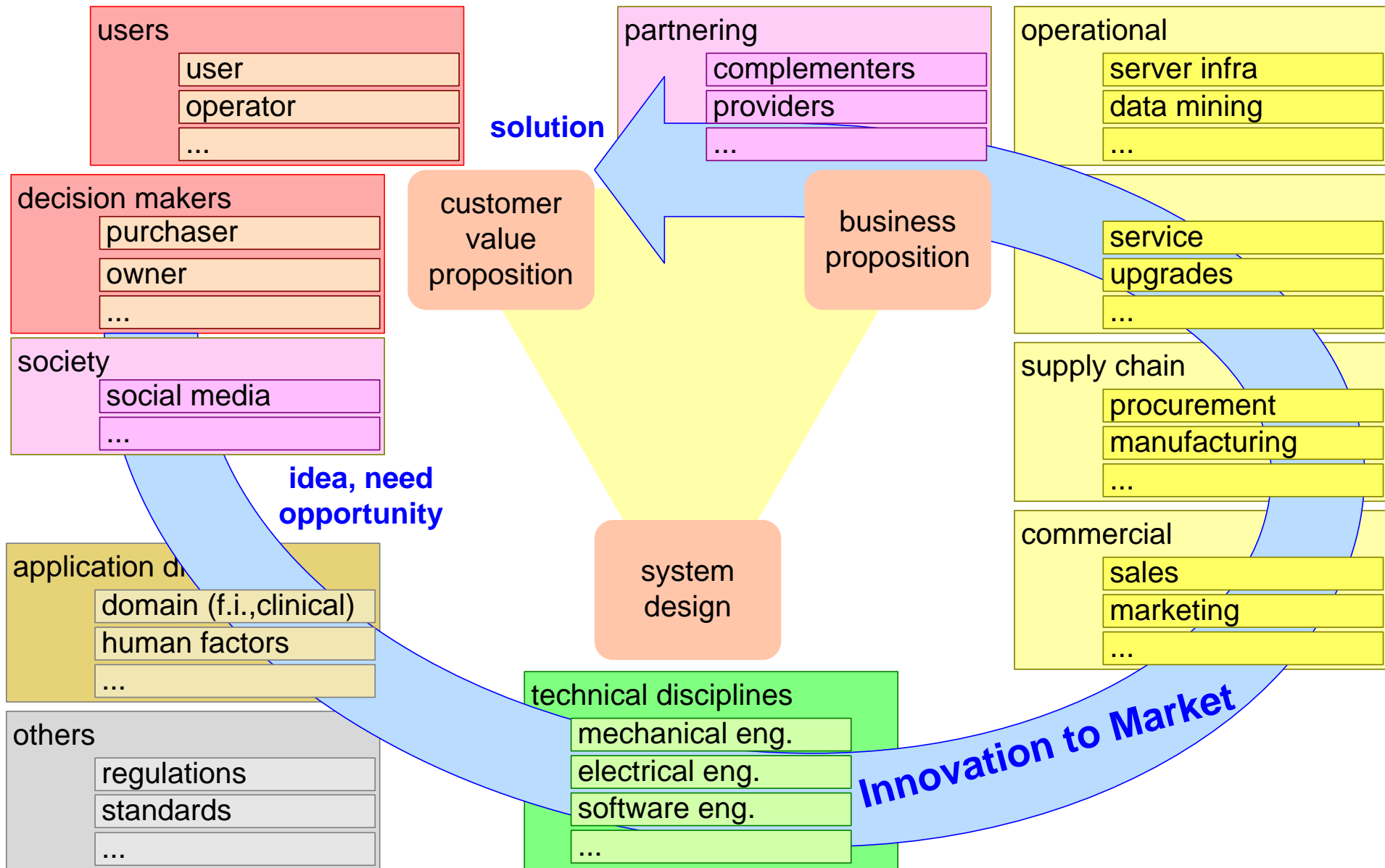
Discipline Stakeholders



All Stakeholders



Innovation to Market



Architecting for Business Value; System of Interest

by *Gerrit Muller* [TNO-ESI, University of South-Eastern Norway]

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

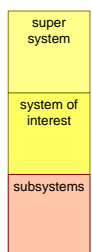
Abstract

This presentation introduces System of Interest in the context of the supersystem, and decomposed into subsystems.

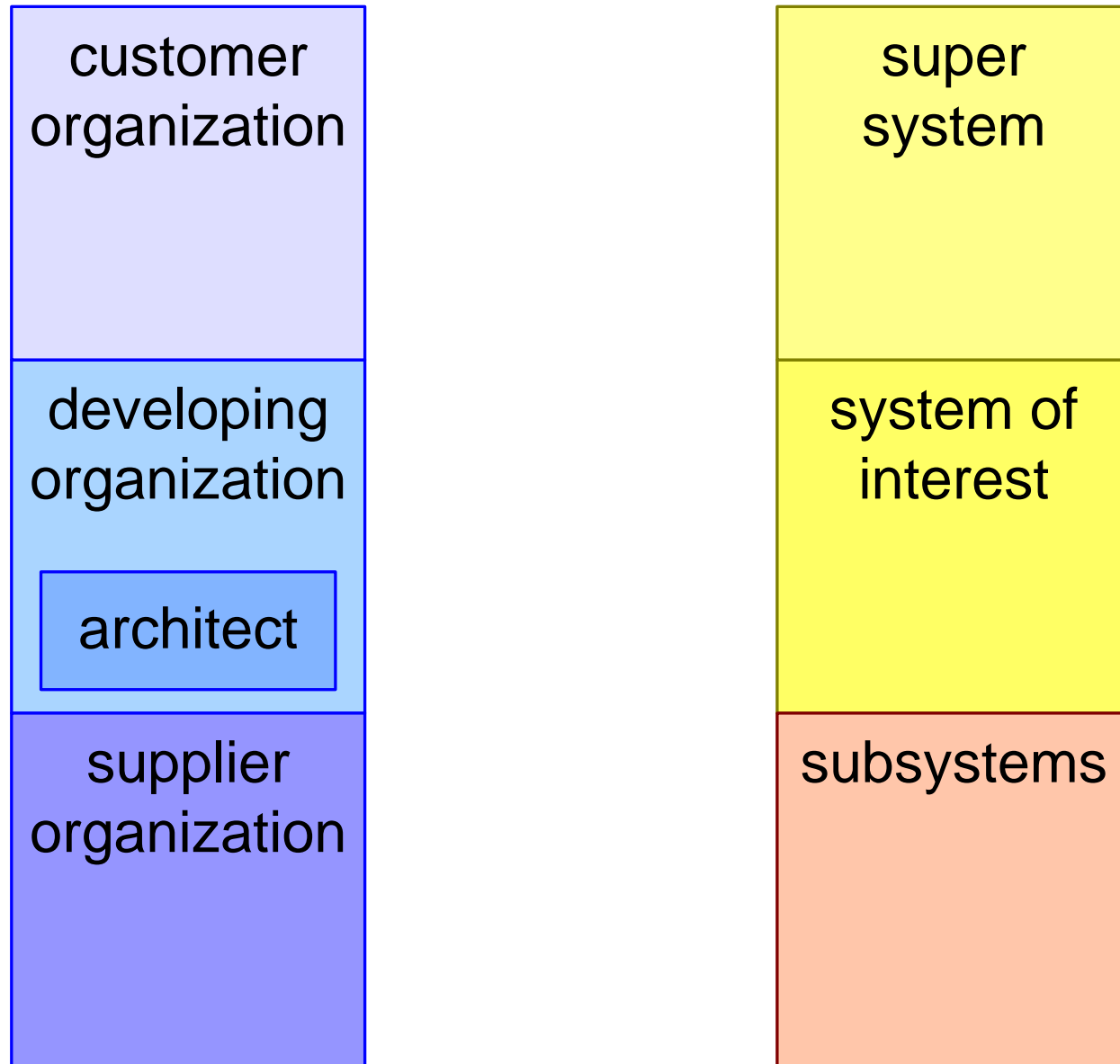
Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

January 27, 2026
status: preliminary
draft
version: 0



System of Interest

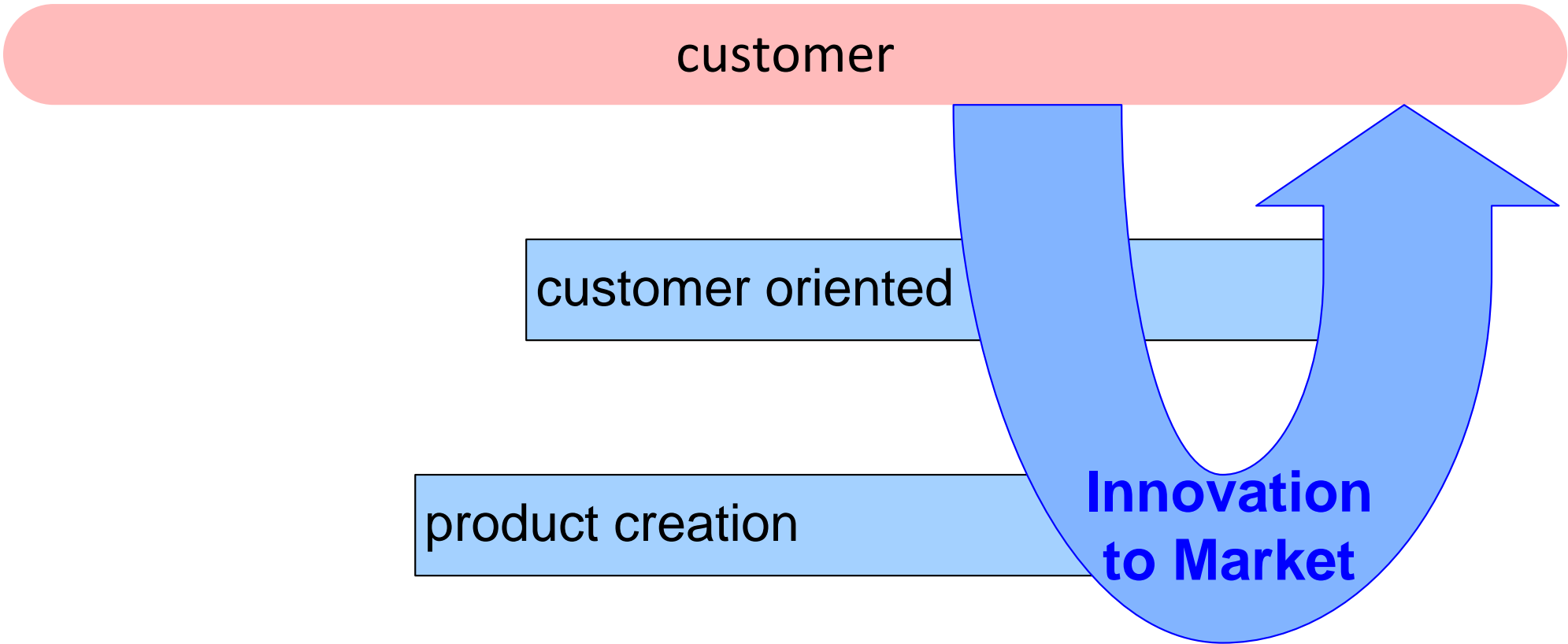


Exercise Sketch System of Interest

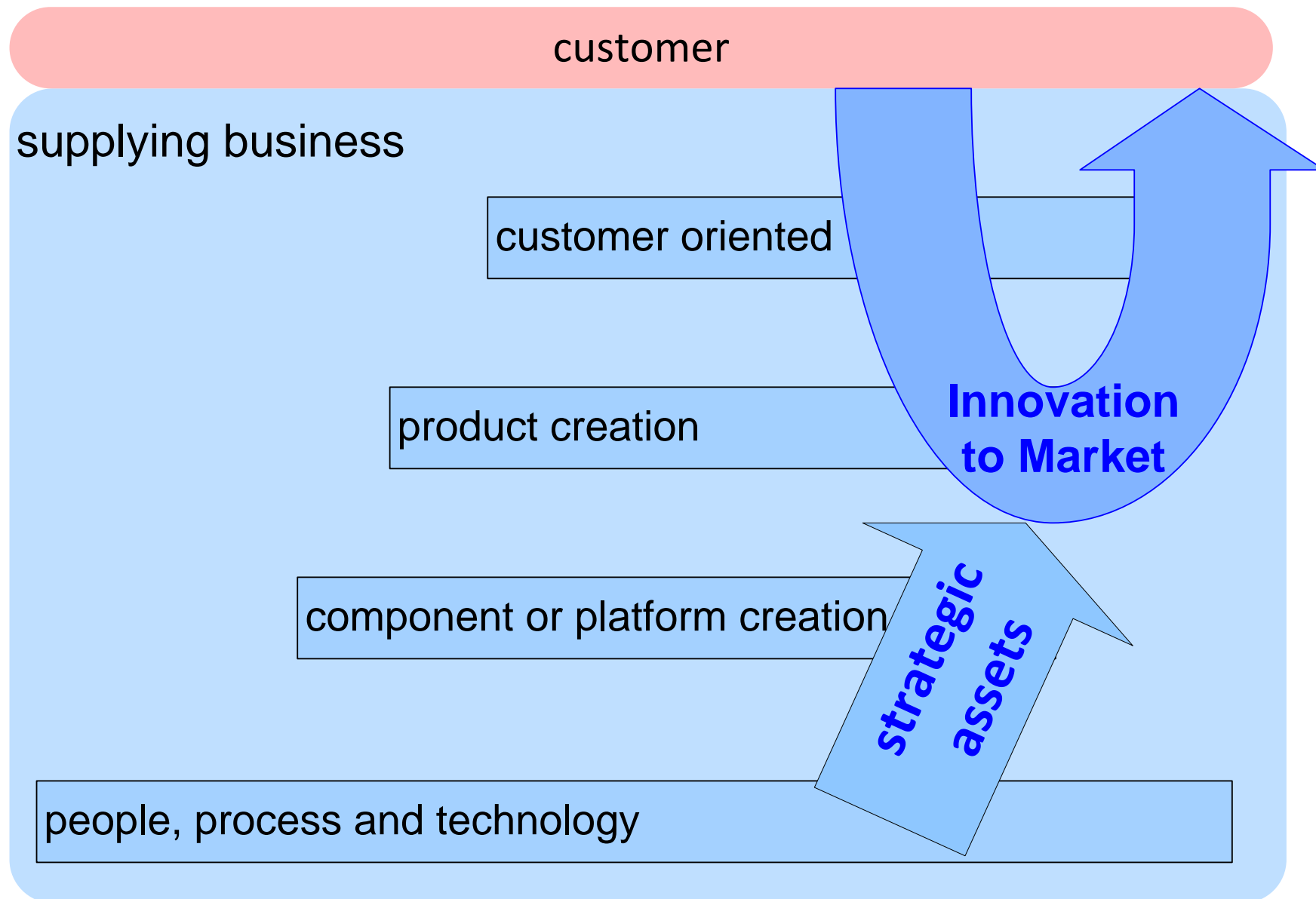
Sketch the **System-of-Interest** in its **context**

- Show some of the internals of the system-of-interest
- Indicate the boundary of the system-of-interest

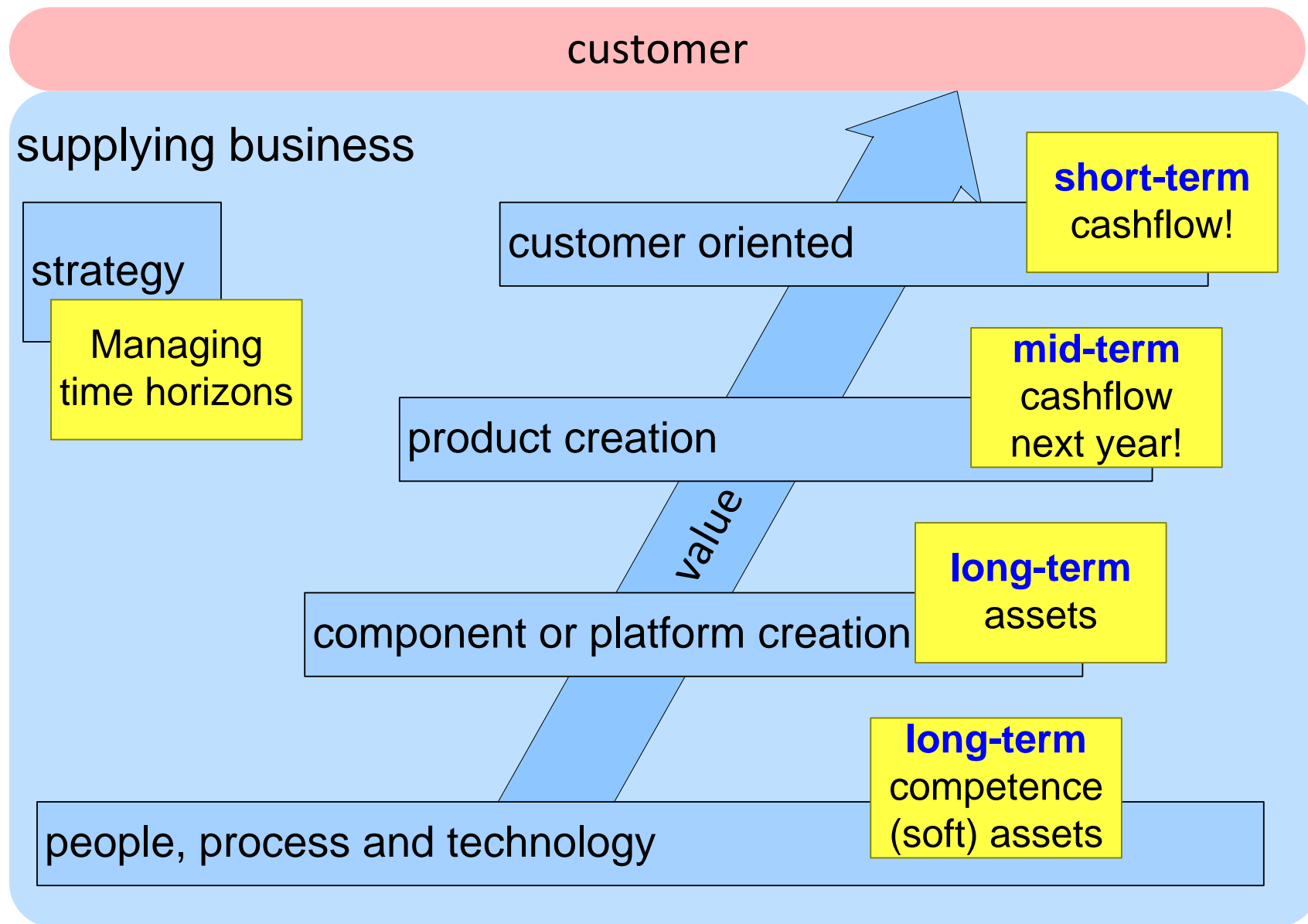
Process Model Innovation to Market



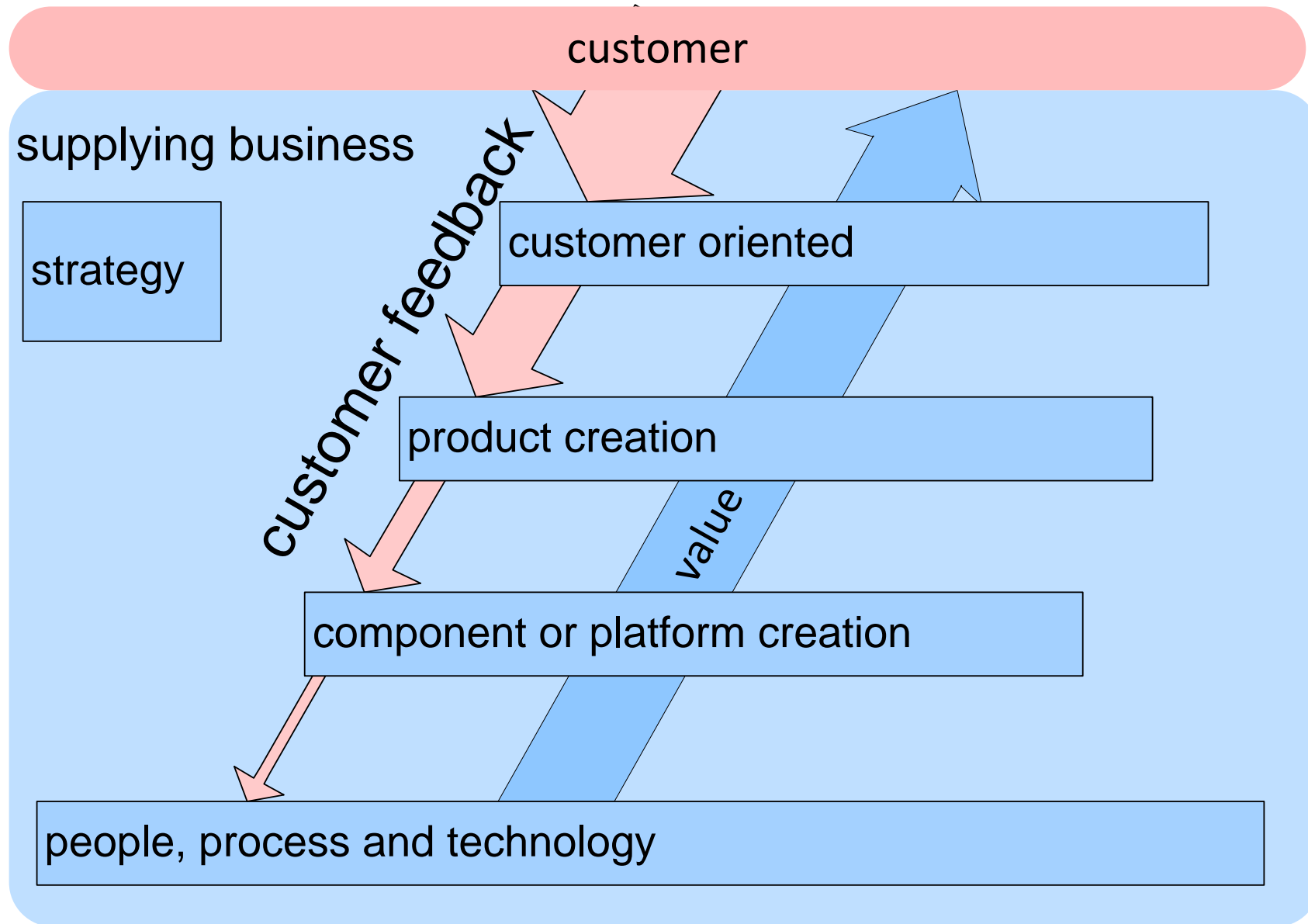
Strategic Assets Accelerate Time to Market



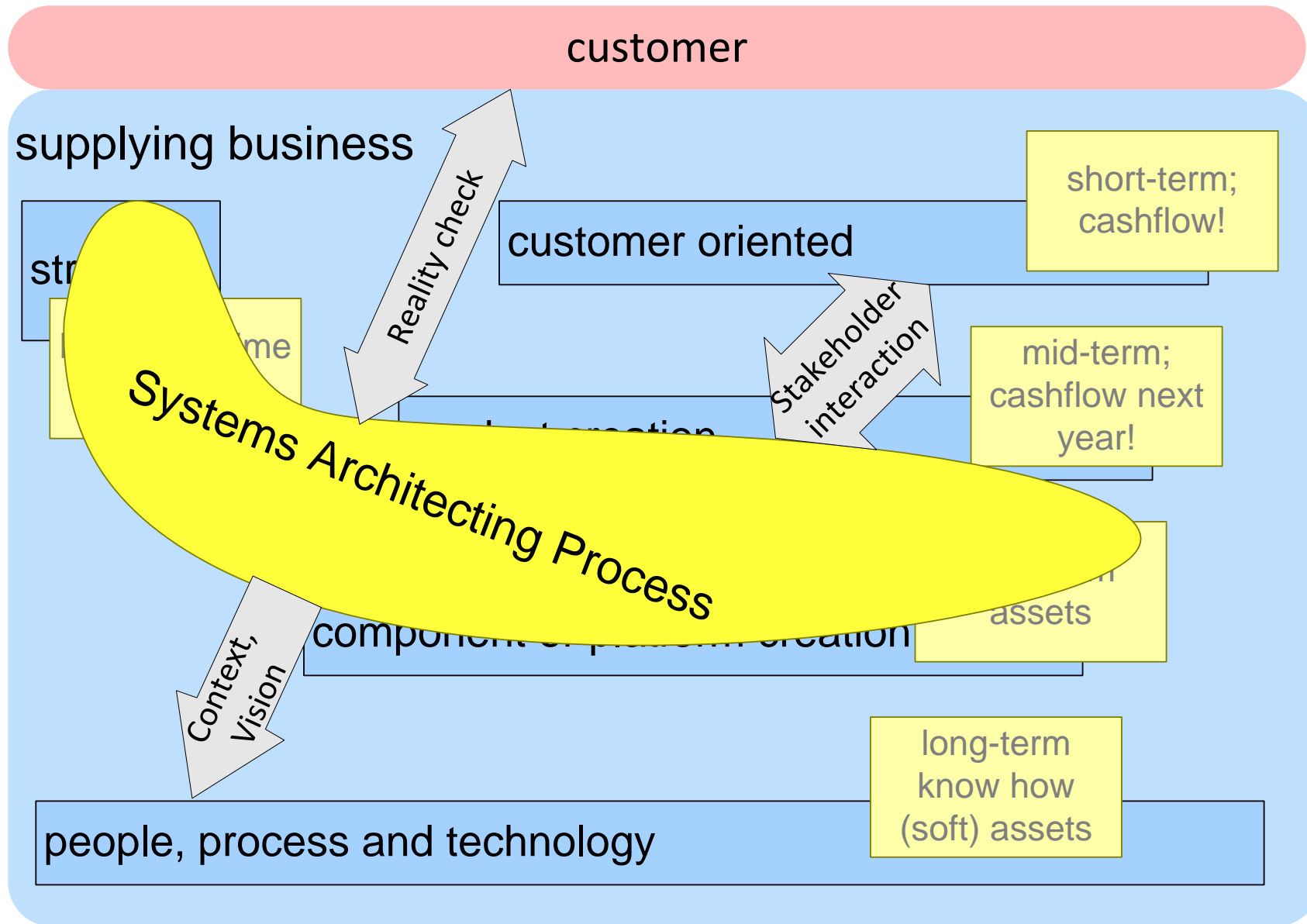
Time Dimension and Strategy



Information Flow



Integral Process Model



Architecting for Business Value; Business Context - Solution Creation

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

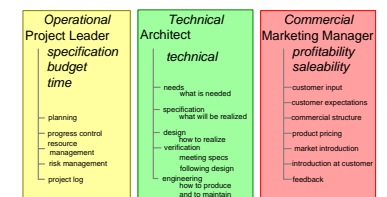
Abstract

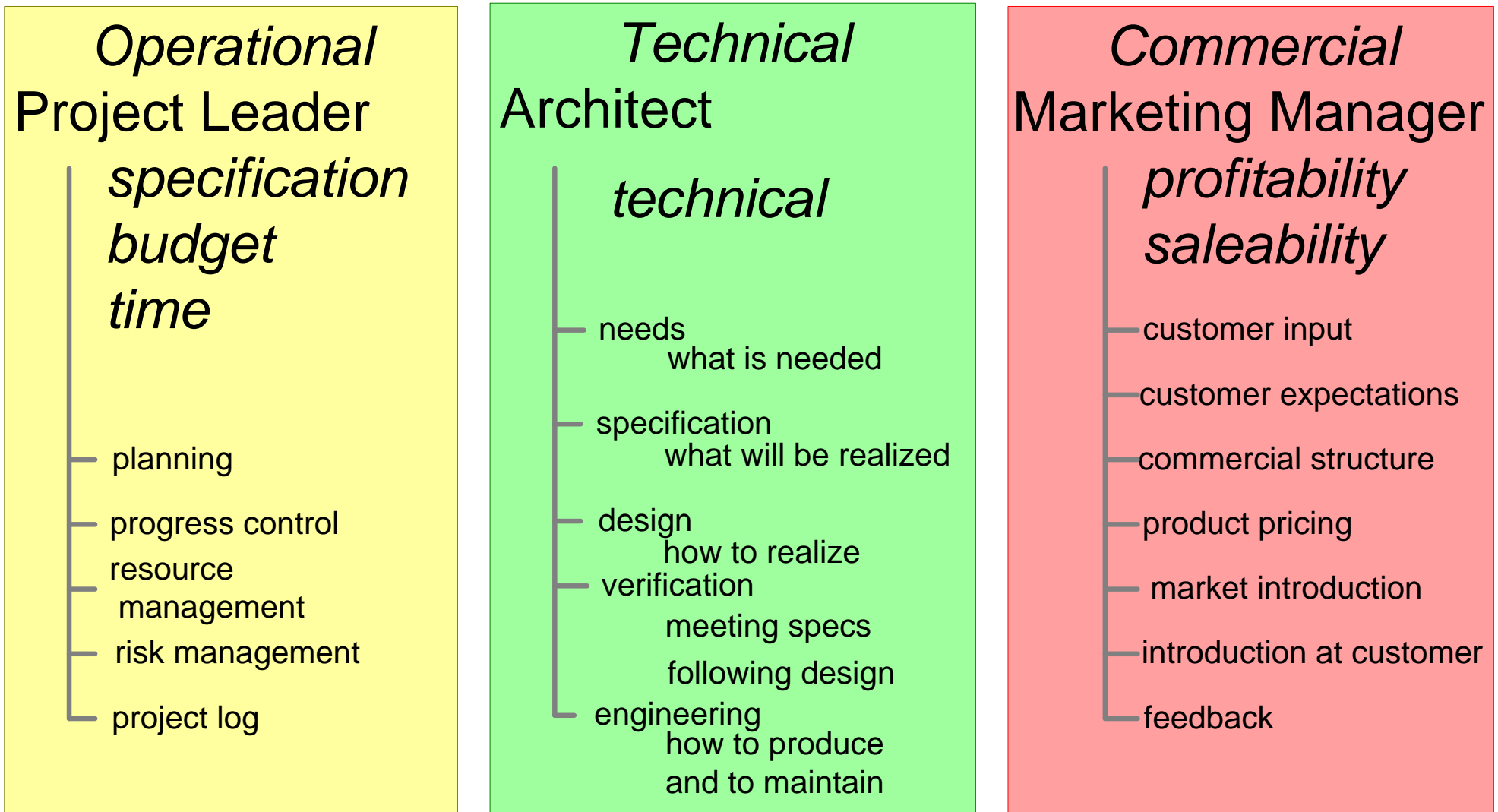
This presentation shows a model to create solutions via projects.

Distribution

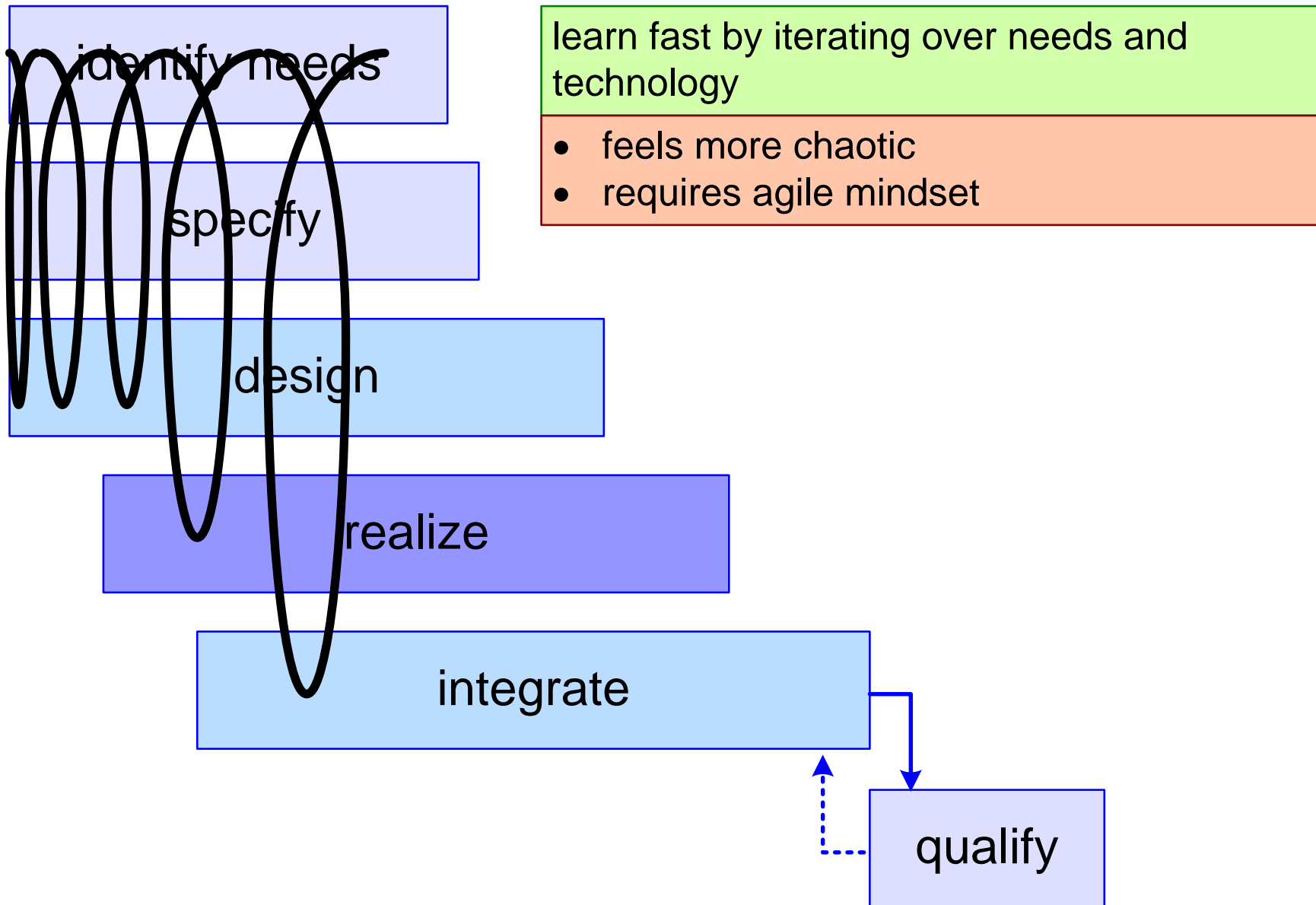
This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

January 27, 2026
status: draft
version: 0





Conventional and Agile Development Models



Architecting for Business Value; Role of the Architect

by *Gerrit Muller* [TNO-ESI, University of South-Eastern Norway]

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

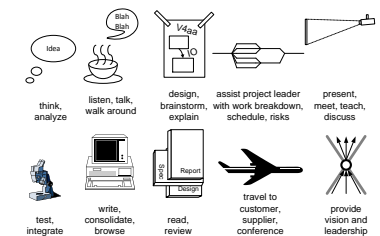
Abstract

This presentation discusses the role of the architect by looking at deliverables, responsibilities, and activities.

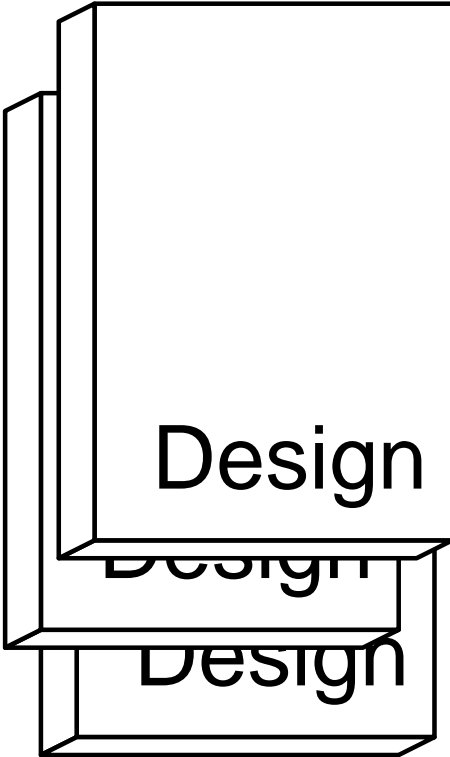
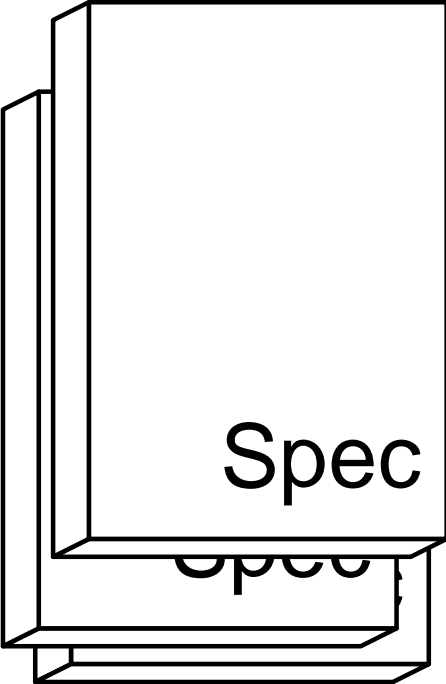
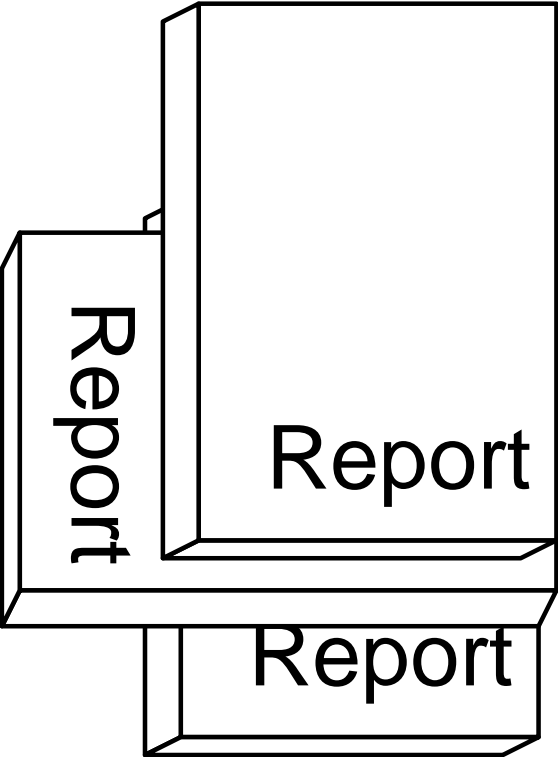
Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

January 27, 2026
status: draft
version: 0



Deliverables of the System Architect



List of Deliverables

Customer and Life-Cycle Needs (*what is needed*)

System Specification (*what will be realized*)

Design Specification (*how the system will be realized*)

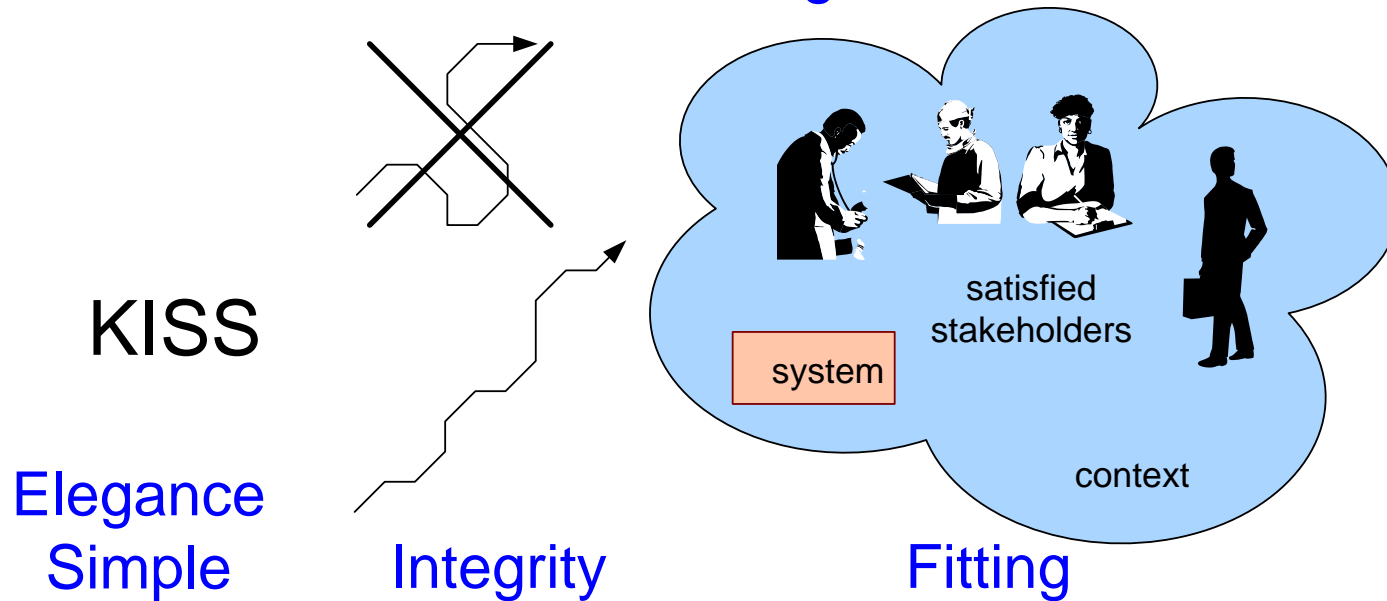
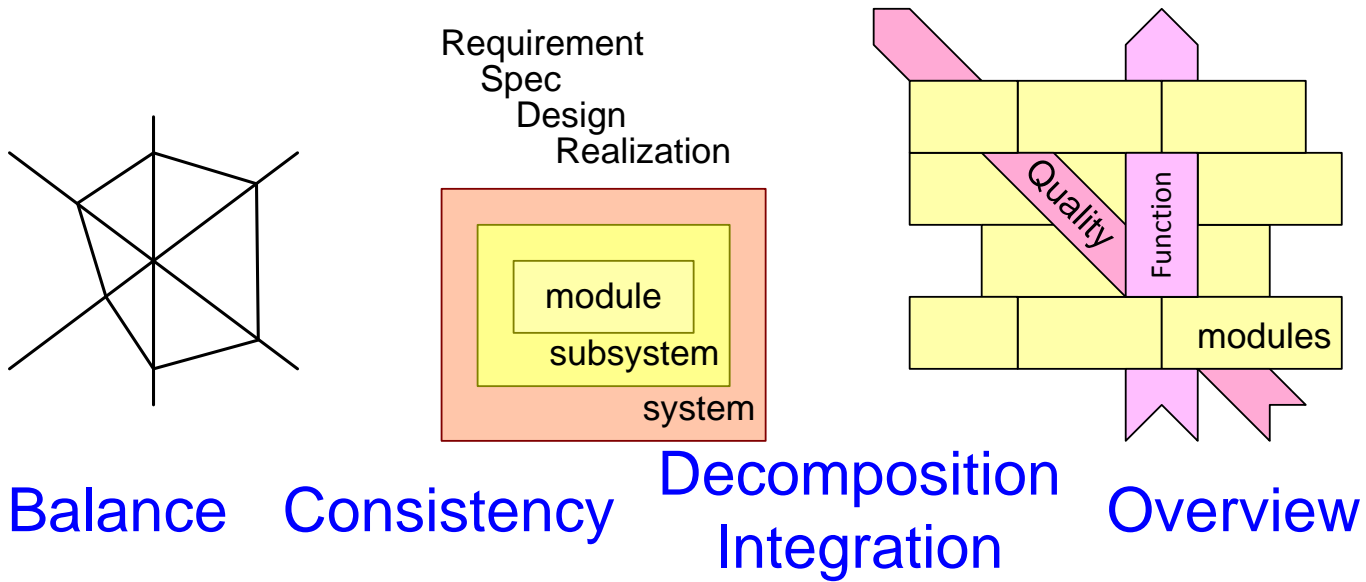
Verification Specification (*how the system will be verified*)

Verification Report (*the result of the verification*)

Feasibility Report (*the results of a feasibility study*)

Roadmap

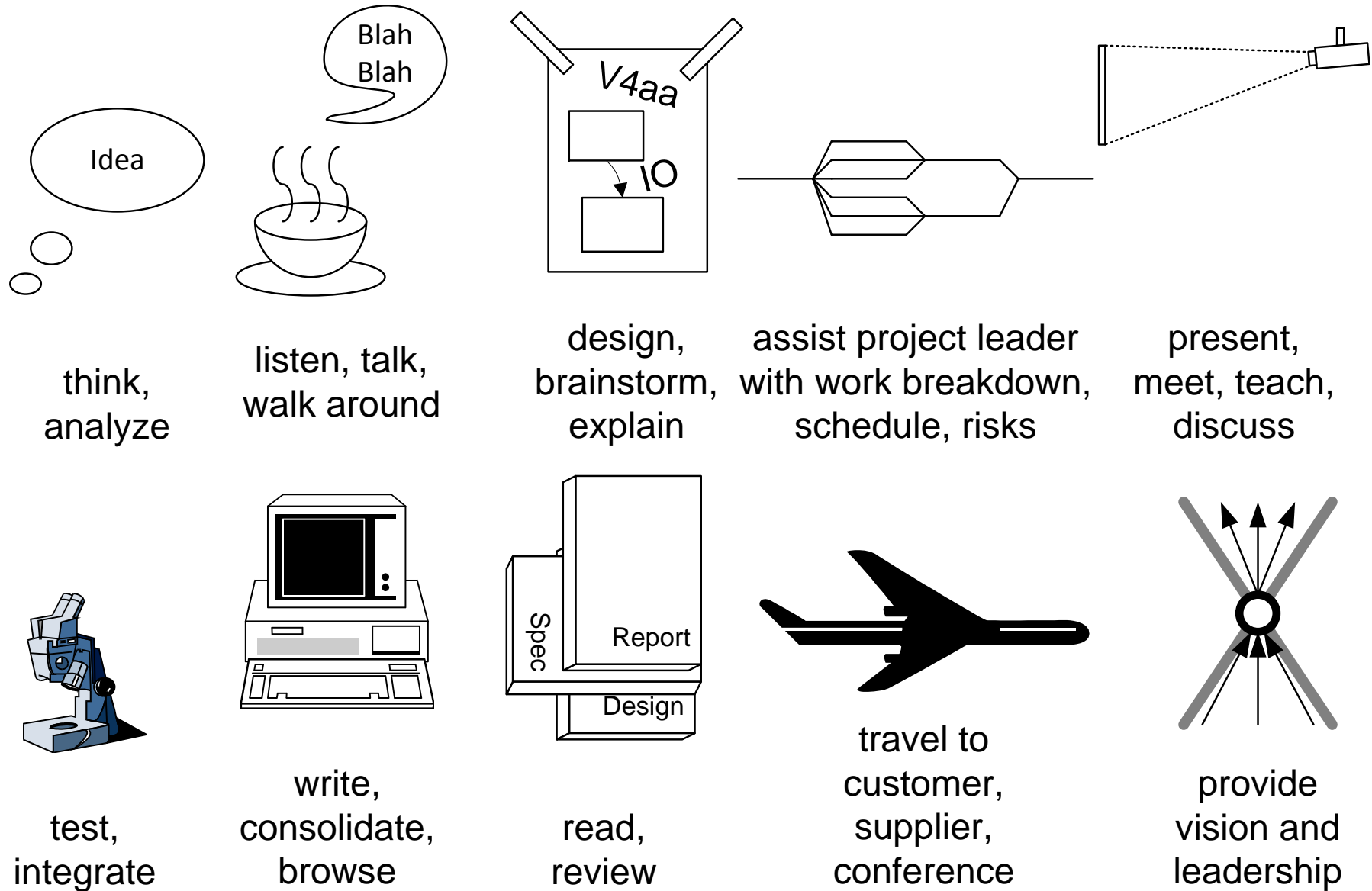
Responsibilities of the System Architect



Examples of Secondary Responsibilities

responsibility	primary owner
business plan, profit	business manager
schedule, resources	project leader
market, saleability	marketing manager
technology	technology manager
process, people	line manager
detailed designs	engineers

What does the System Architect do?



Architecting for Business Value; Viewpoint Hopping

by *Gerrit Muller* [TNO-ESI, University of South-Eastern Norway]

e-mail: gaudisite@gmail.com

www.gaudisite.nl

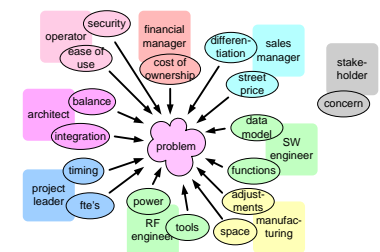
Abstract

This presentation explains how architects hop over various viewpoints to build up understanding of problem and solution space.

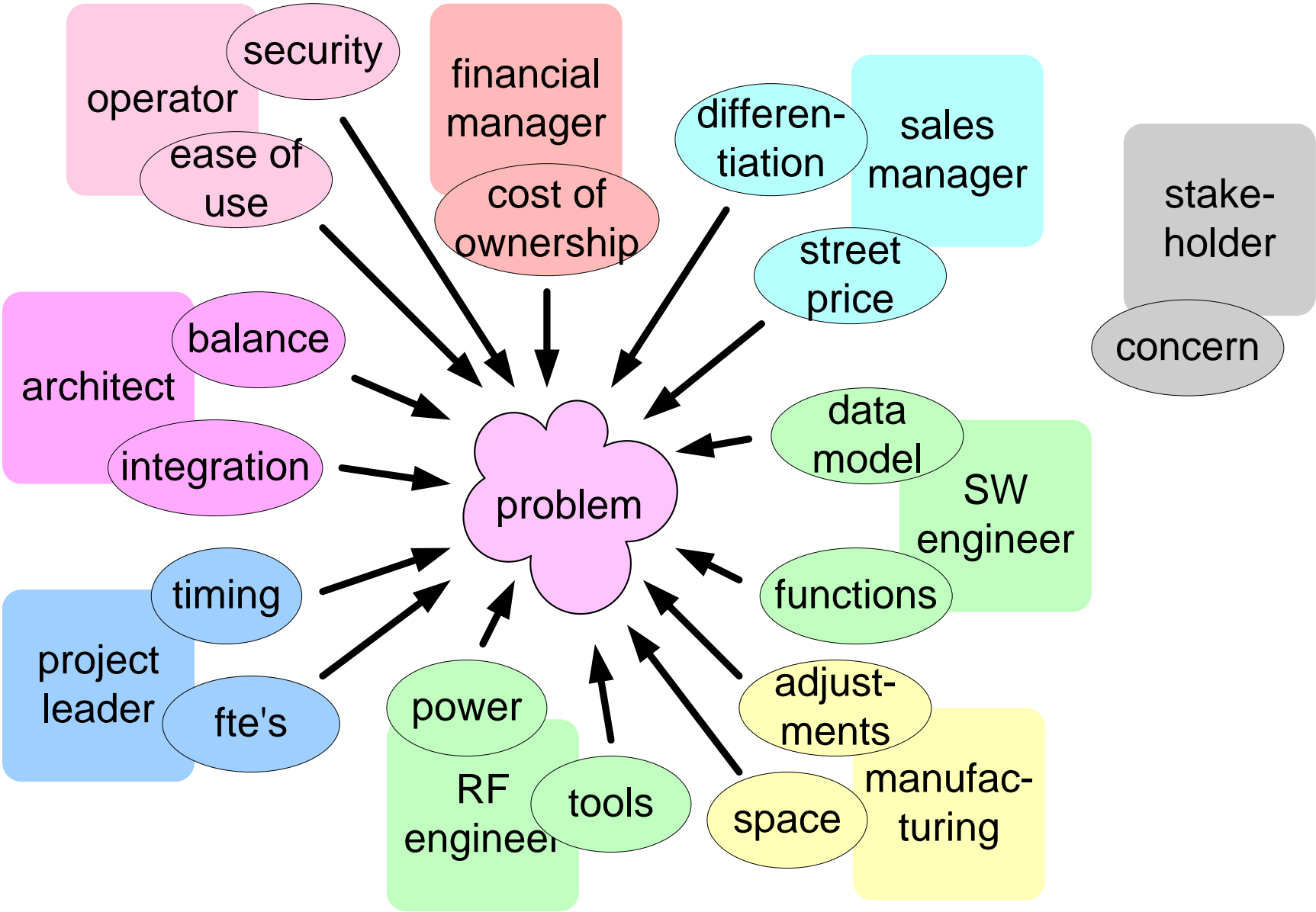
Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

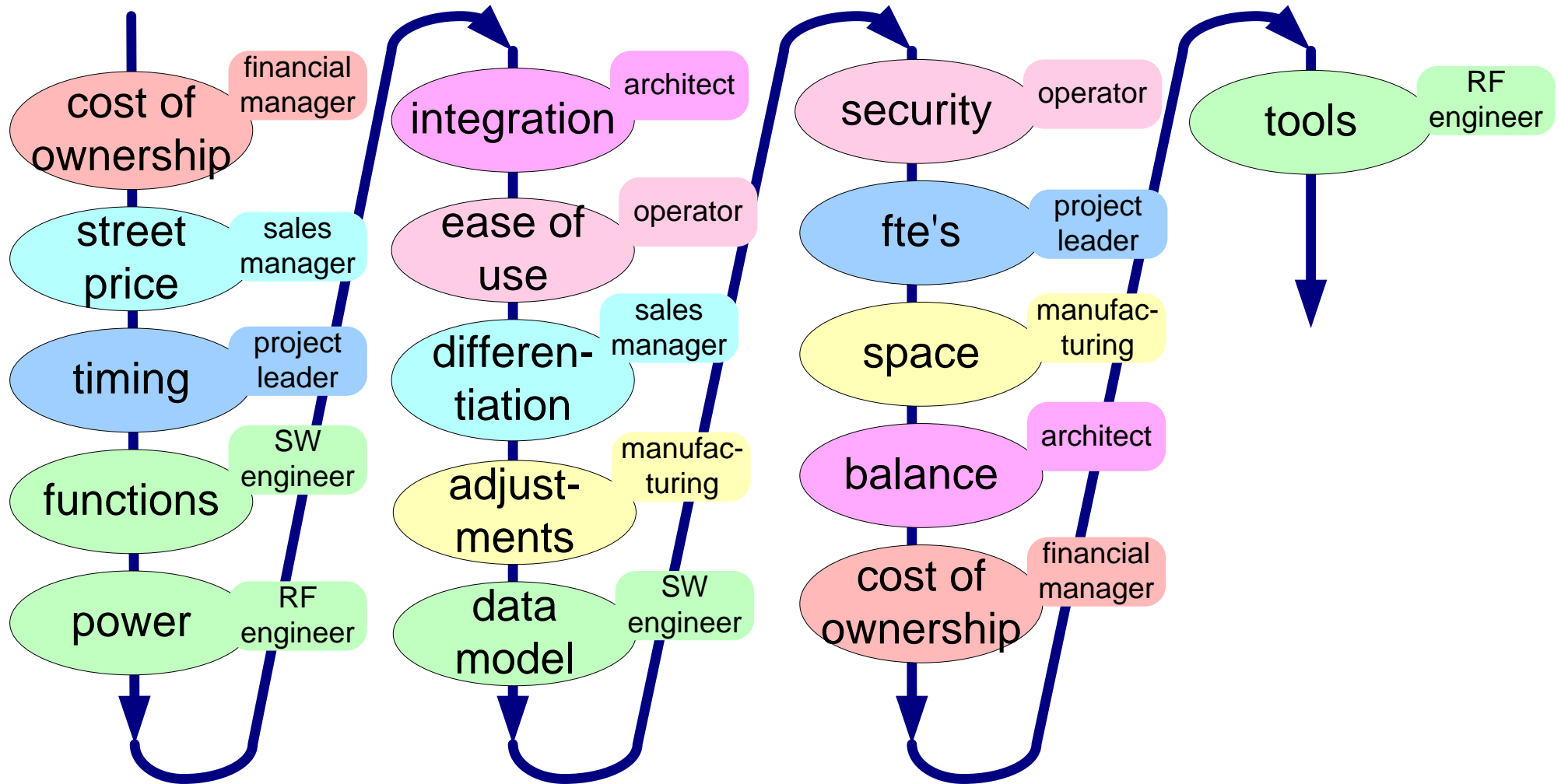
January 27, 2026
status: draft
version: 0



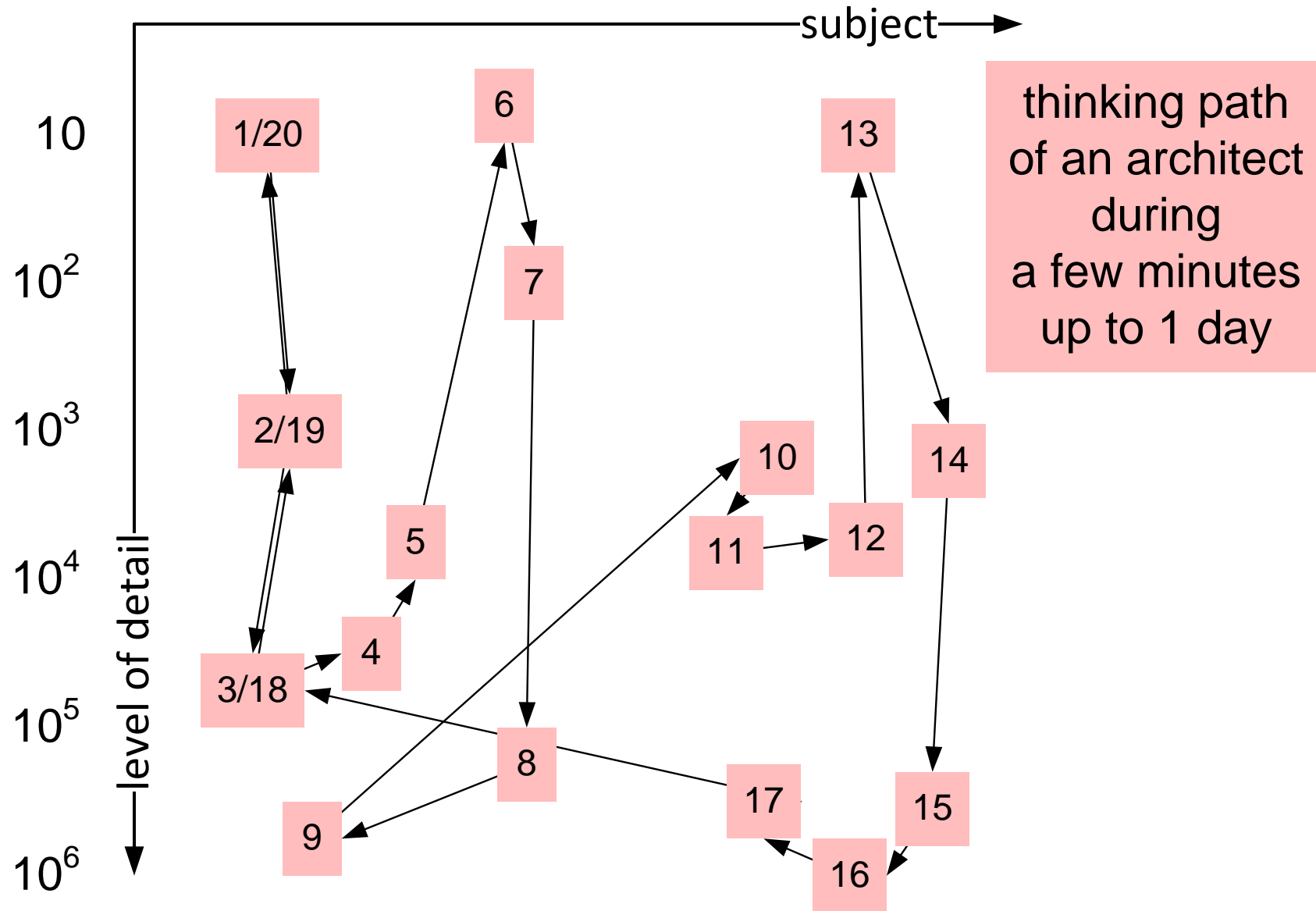
Many viewpoints



Viewpoint Hopping



The seemingly random exploration path



Architecting for Business Value; T-shaped Architects

by *Gerrit Muller* [TNO-ESI, University of South-Eastern Norway]

e-mail: gaudisite@gmail.com

www.gaudisite.nl

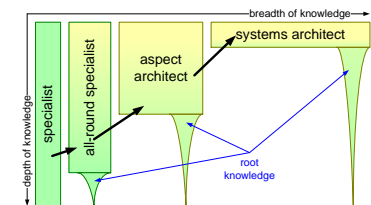
Abstract

This presentation positions architects as generalists, complementing sepcialists.

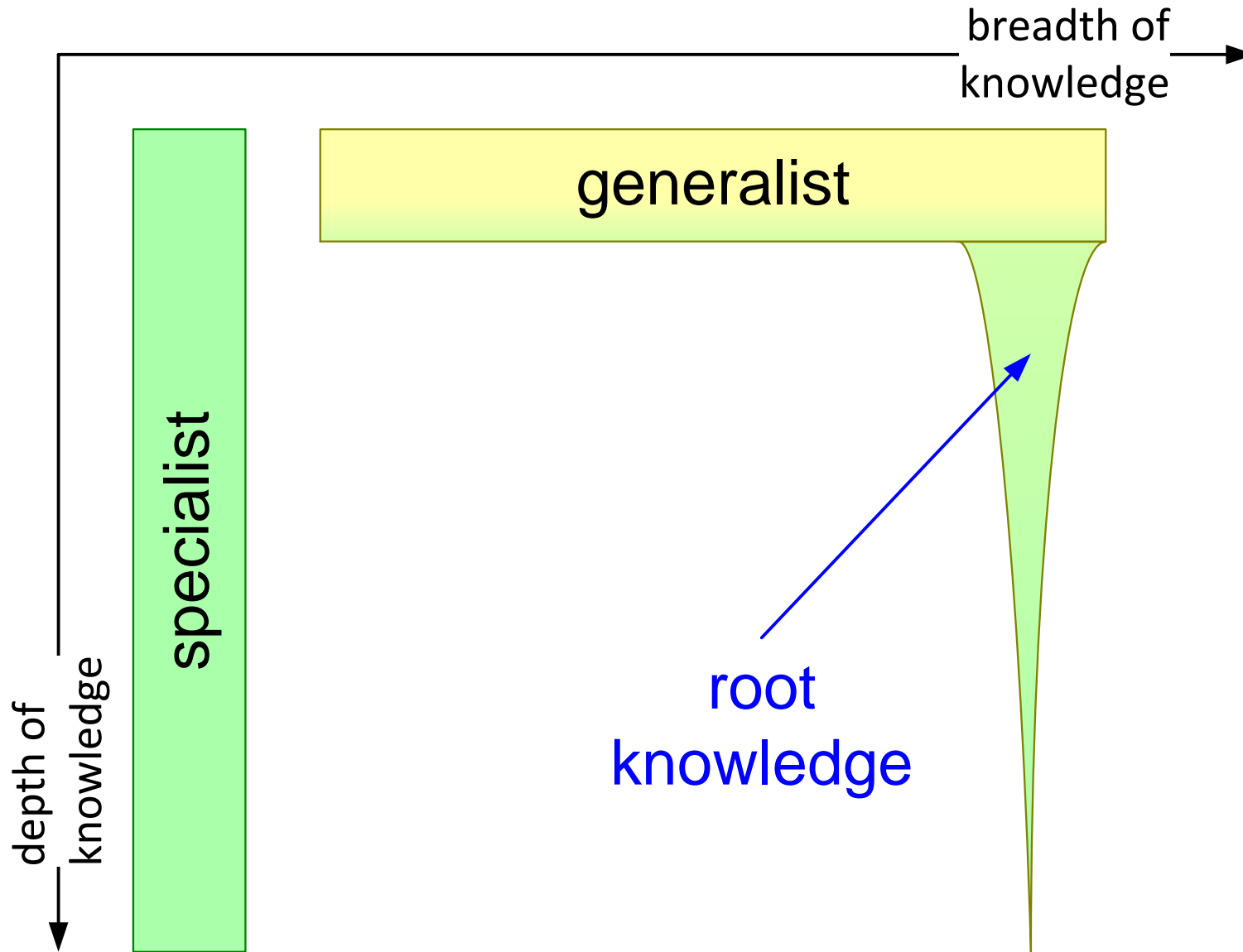
Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

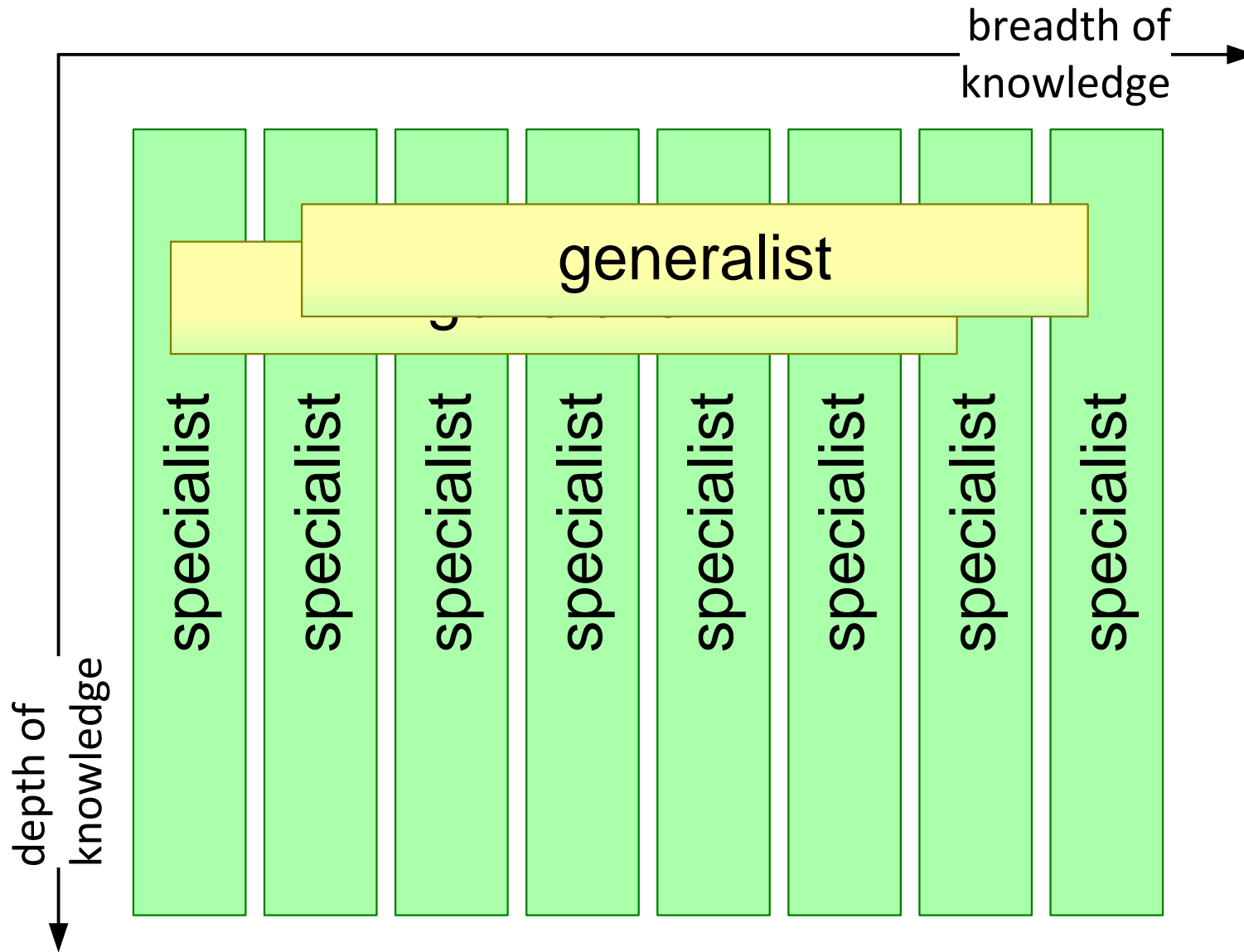
January 27, 2026
status: draft
version: 0



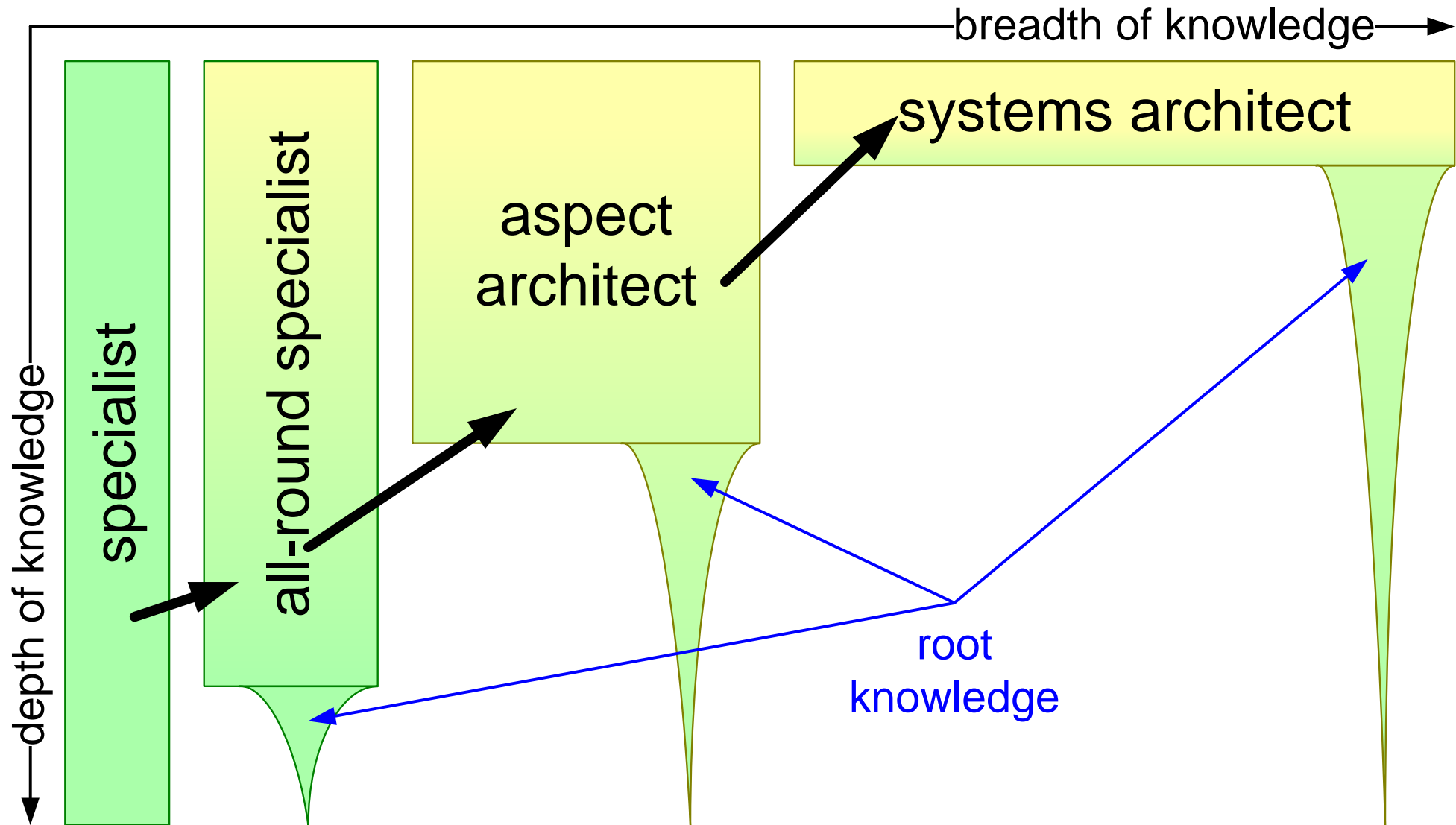
Generalist versus Specialist



Generalists and Specialists are Complementary



Spectrum from Specialist to System Architect



Architecting for Business Value; Challenges and Dilemmas

by *Gerrit Muller* [TNO-ESI, University of South-Eastern Norway]

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

Abstract

This presentation challenges, dilemmas, and pitfalls in architecting.

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

January 27, 2026

status: preliminary

draft

version: 0

logo

TBD

Challenges and Dilemmas - Business Context

- Multi-disciplinarity
 - Amount and distance stakeholders
 - Connecting breadth and depth:
 - Customer value proposition
 - business proposition
 - Design and Technology
 - Many unknowns, uncertainties, little time
- Role of the architect is vague (holistic)
 - Many stakeholders consider architect as a “trespasser”
 - Architects take decisions with lifecycle and business impact without detailed information.
 - How to deal with tensions between long term (strategic), mid term (tactical), and short term (operational).

Challenges and Dilemmas - Role and Task

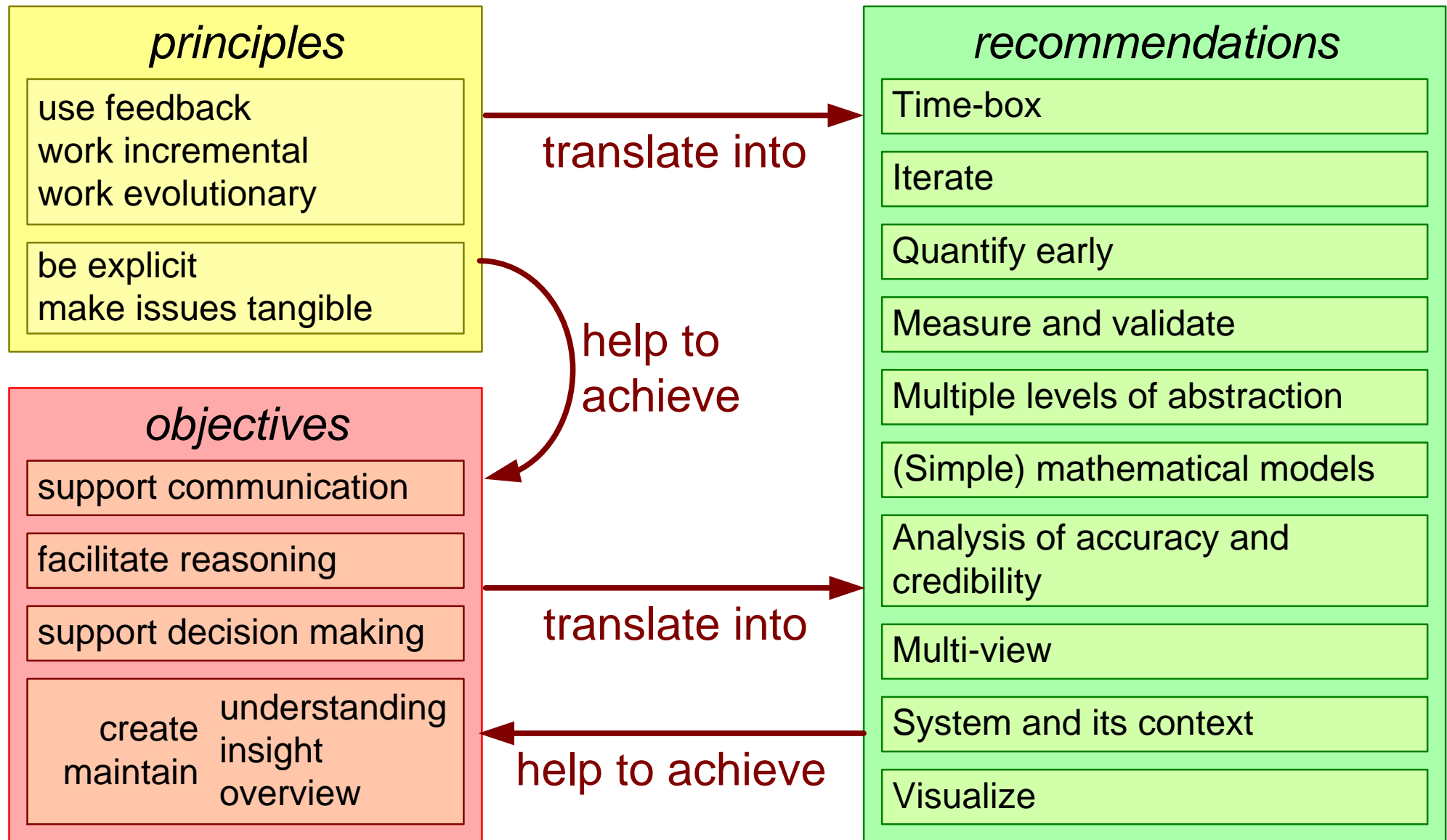
- Limited deliveries; information/documentation predominantly
- Vague responsibilities
- Most responsibilities are shared and owned by someone else
- Many activities are necessary for the result, however, more or less invisible
- Architect works via influence, without formal power
- Architect and project leader are “opposing mates”

- How to grow in breadth?
- How to maintain depth?
- How to connect with other depth disciplines?
- Customer value vs. engineering delight
- Required skills & process available skills & process
- Architectural cleanness product release date
- Technology improvement risk reduction

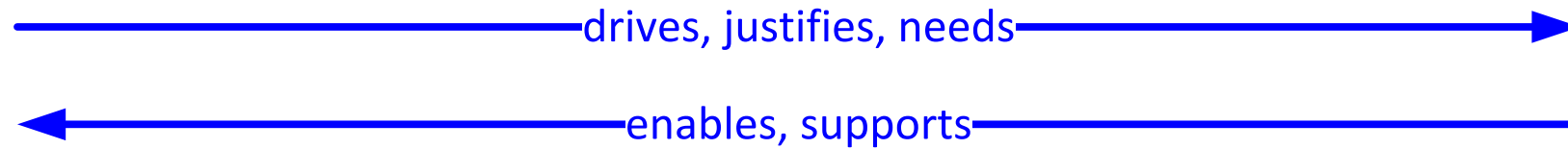
Risks and Pitfalls - Architecting

- What happens without architect or architecting team?
- Ivory tower architects
- When working agile, architecting is not needed
- Architecting costs time
- Investment in architecting is a waste, since we do not know enough
- Architect does not have power, needs perseverance

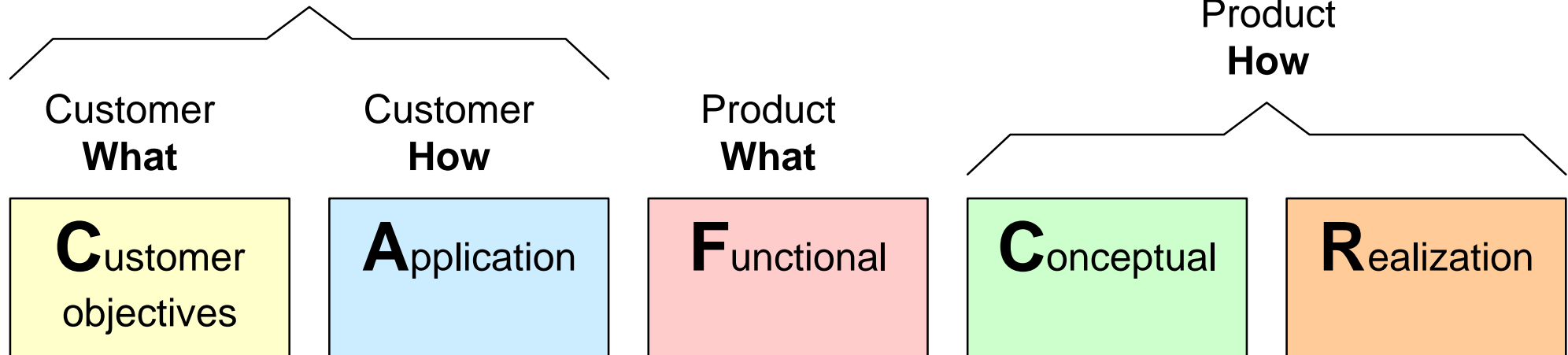
Recommendations as Central Thread



The “CAFCR” model

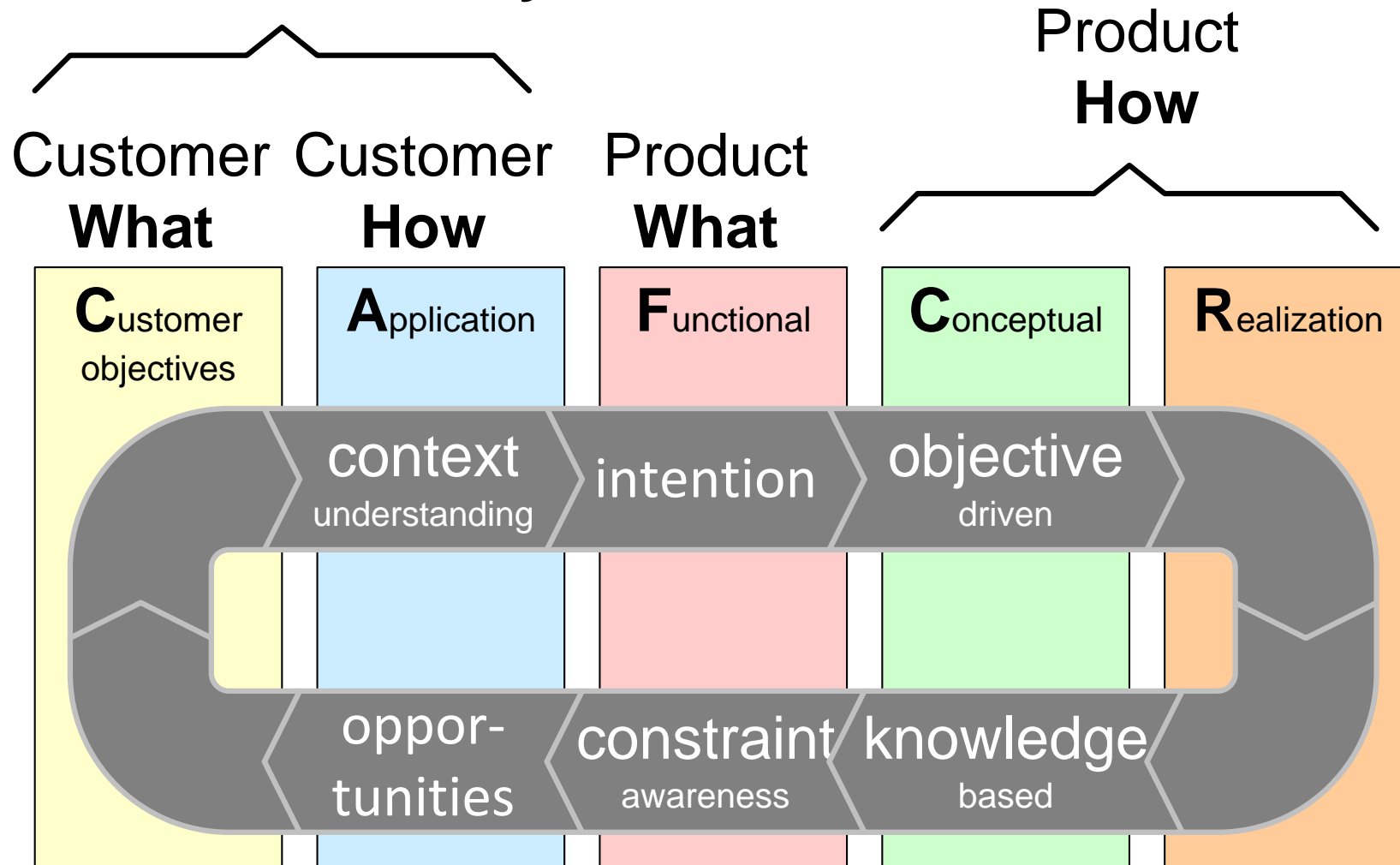


What does Customer need
in Product and **Why?**

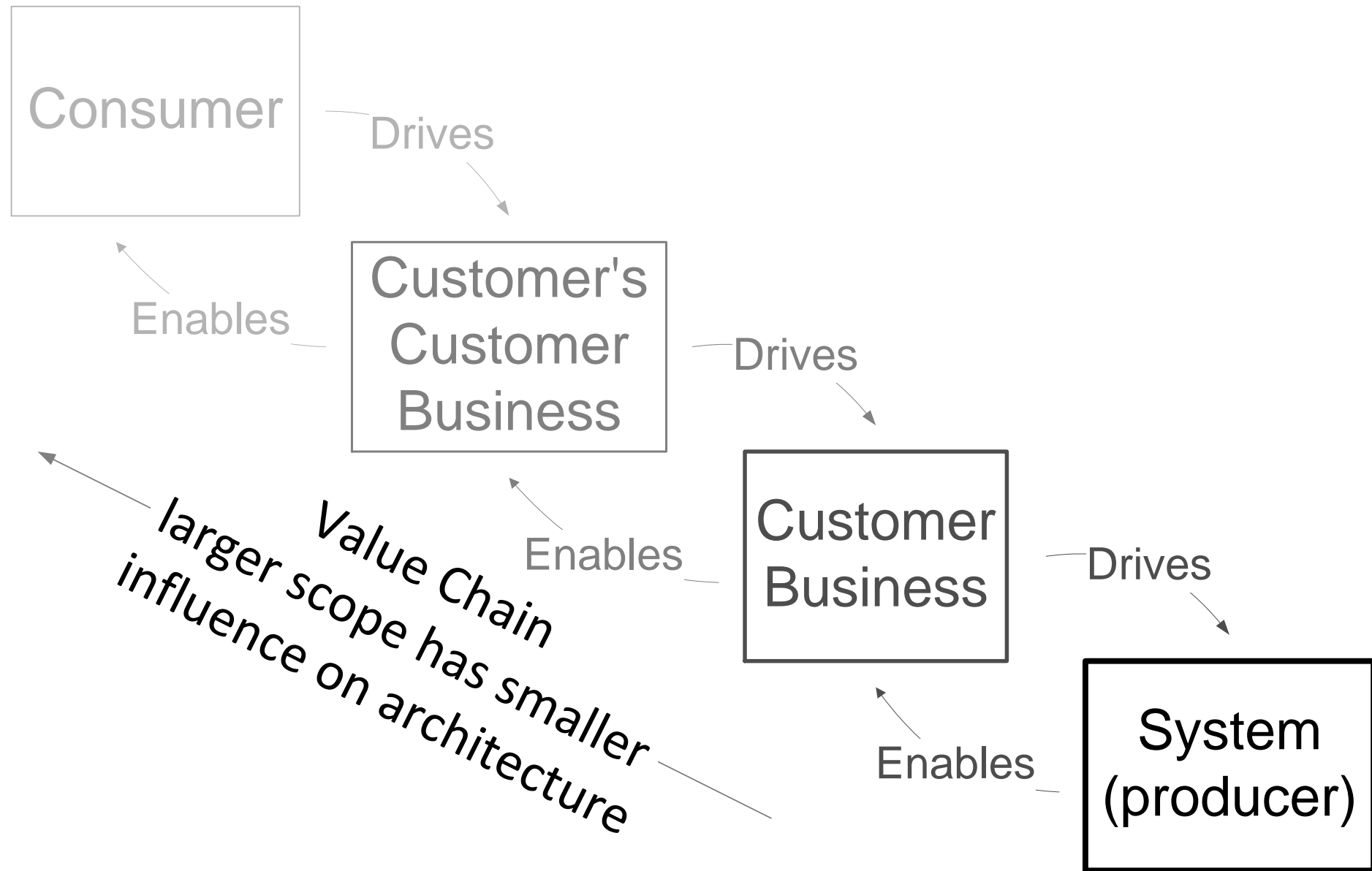


Integrating CAFCR

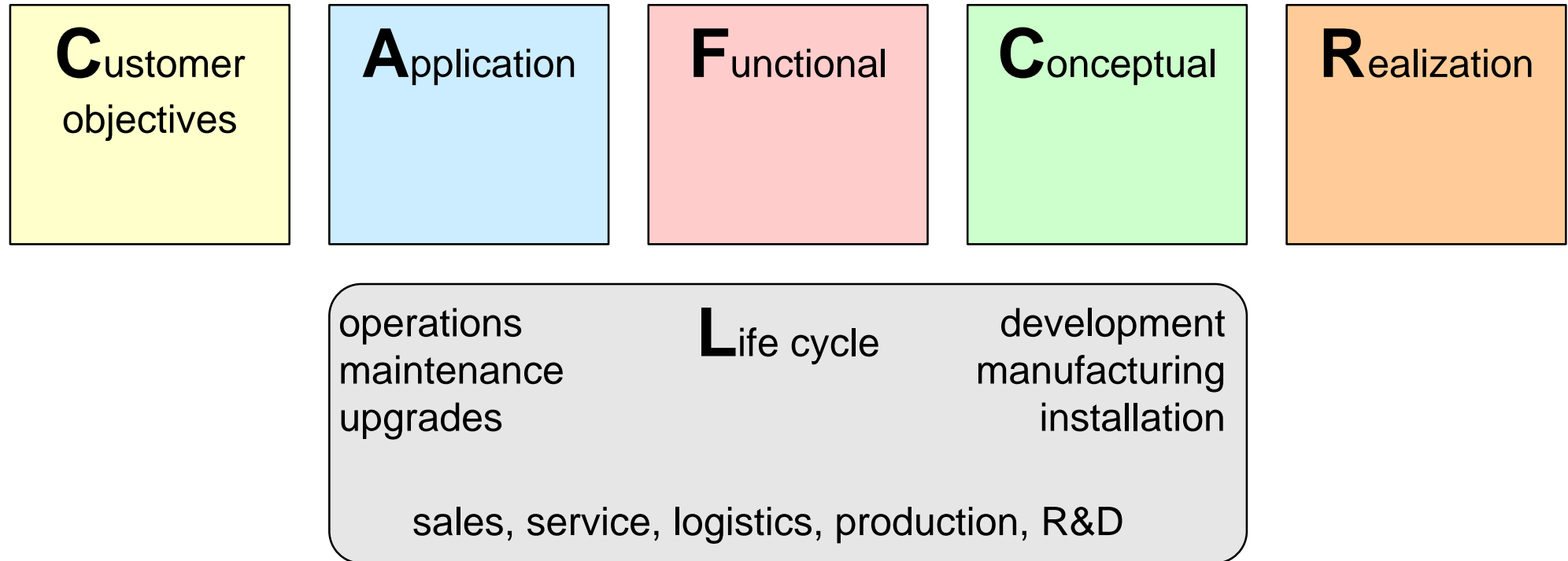
What does Customer need
in Product and **Why?**



CAFCR can be applied recursively



CAFCR+ model; Life Cycle View



Final Delivery: Presentation to Top Management

Value Proposition

*Why do customers want to buy?
Why do users like to use the system?*

Business Proposition

*How do we earn money?
How do we run a healthy business?*

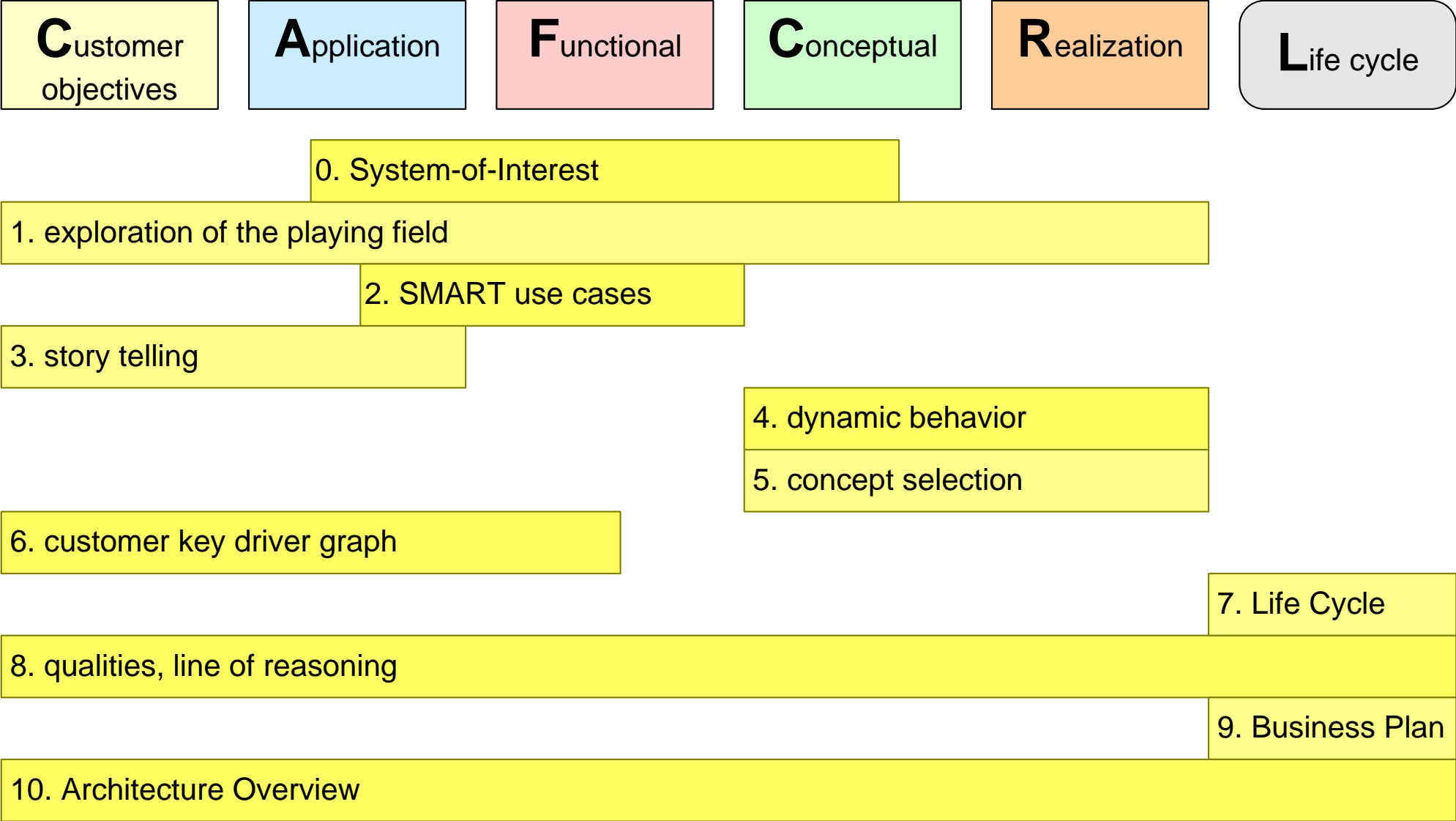
System Specification

*What do customers get?
What is the system-of-interest that we deliver?*

Design

*How will we realize this specification?
How do we ensure performance, safety,
robustness, etc.?*

Exercises Mapped on CAFCR+



Initial CAFCR scan; top-down

by *Gerrit Muller* HSN-NISE, TNO-ESI

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

Abstract

This presentation guides a team through a quick CAFCR scan. Such quick scan with typically 15 minutes per view helps to build an initial overview of the problem and solution space. This version uses a top-down scan, which is typically useful in greenfield approaches.

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

January 27, 2026
status: preliminary
draft
version: 0.1

Present the results top-down
Use two to three flip charts of the six that have been created.
Explain in five minutes the needs of the customer, the system, and the major design choices.

Exercise Top-down Scan CAFCR

make a top-down analysis of your product:

1. customer objectives
2. application
3. functional
4. conceptual
5. realization
6. qualities

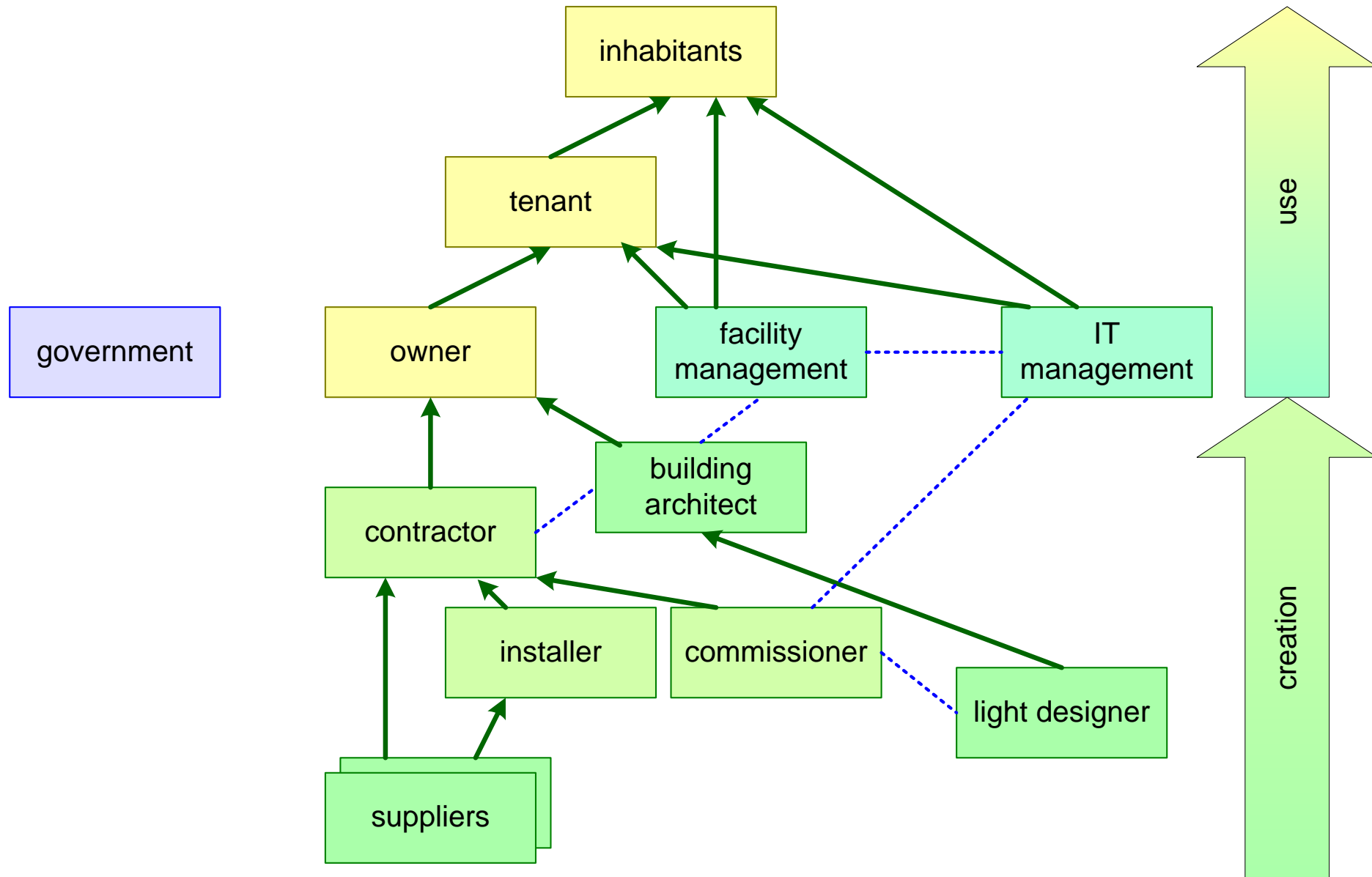
use time boxes of 15 minutes per view

show the most dominant decomposition of that view, as diagram or as a list; some more guidance will be given per step.

Do and Don't

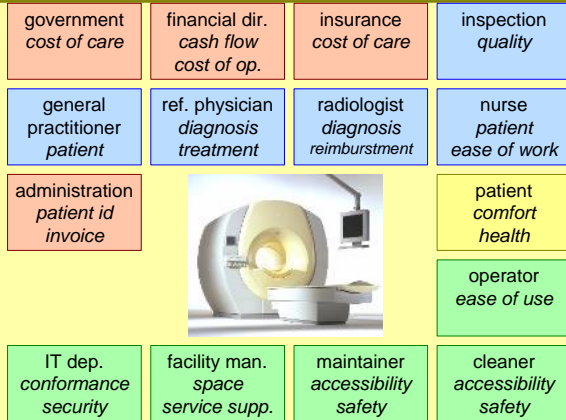
Do	Do not	Because
<ul style="list-style-type: none">• start sketching/drawing as soon as possible• use shared large sheets of paper (e.g. flip-over)• number the flip-overs and add a title• annotate (add notes) during discussions• use yellow note stickers and flip-over markers• be open for ideas and surprises	<ul style="list-style-type: none">• write long texts• immediately capture electronic• have nice but volatile discussions• write with pen or pencil• Do not stick to the first solution	<ul style="list-style-type: none">• sketches stimulate sharing and discussion• sharing and discussion help to explore faster• remembering the order gets challenging• information and insight is quickly lost• stickers are easily (re)moved• you hopefully discover a lot; increased insight will change problem and solution

Step 1: Customer Objectives View; Value Network

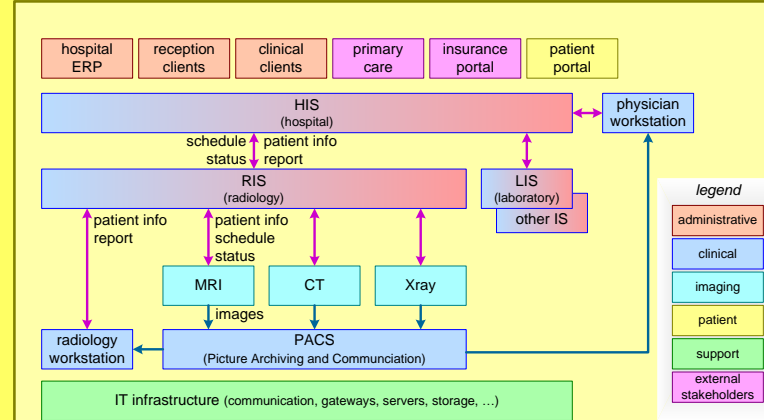


Step 2: Application View

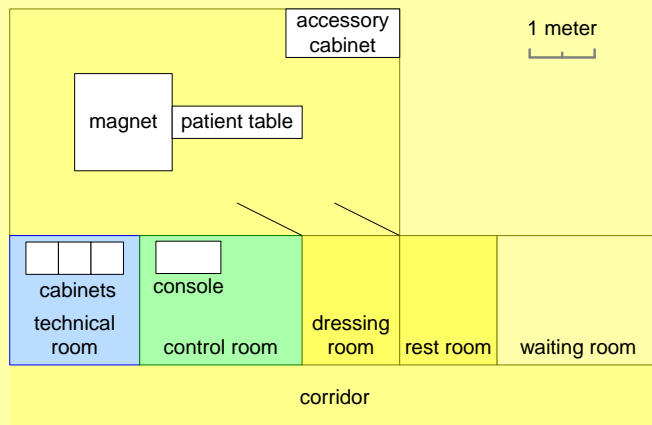
Chose 1 or 2 items from below



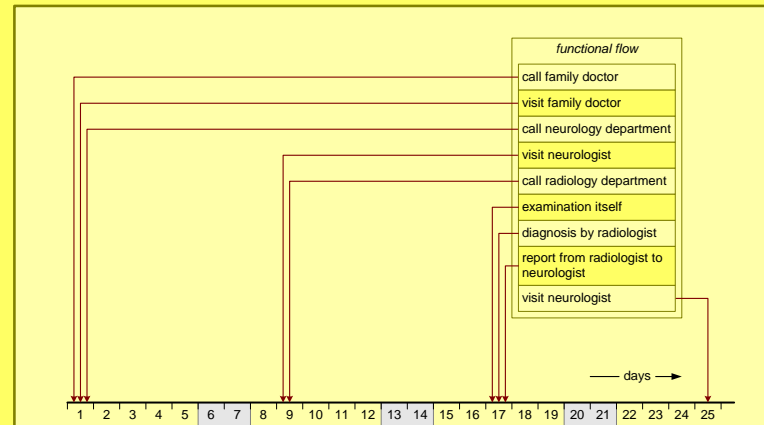
stakeholders and concerns (who)



system context

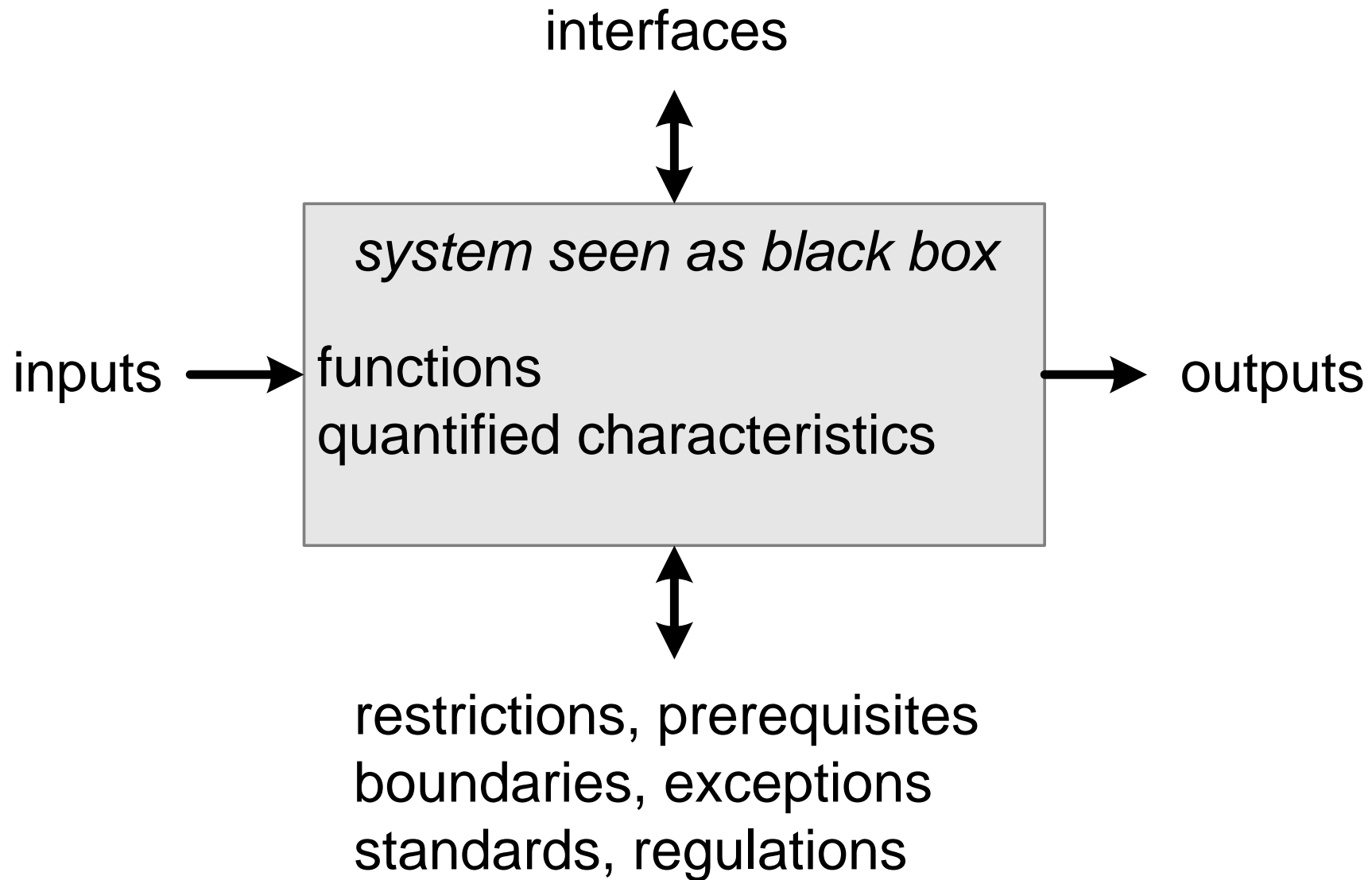


2D map (where)



work flow & time line (what, when)

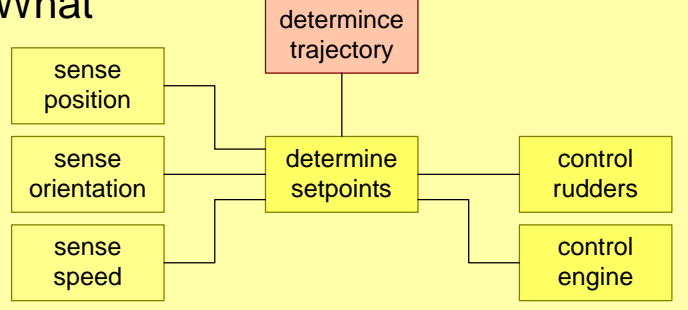
Step 3: Functional View; Top level Spec



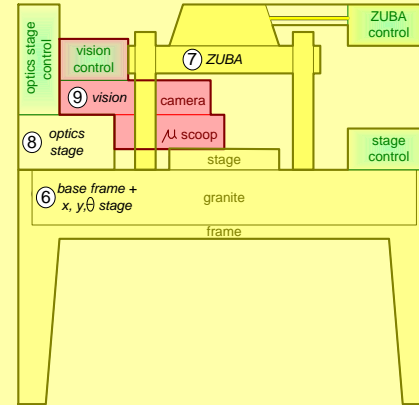
Step 4: Conceptual View

Chose 1 or 2 items from below

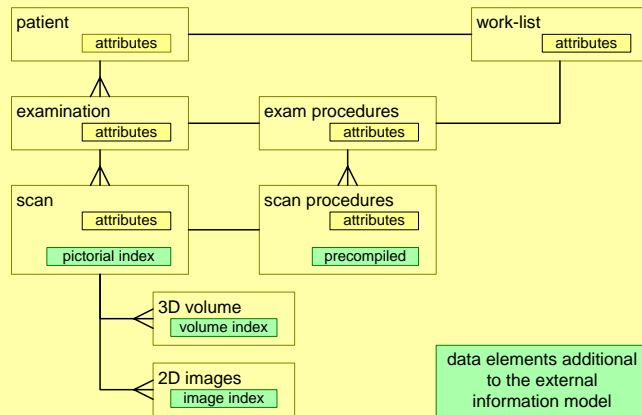
What



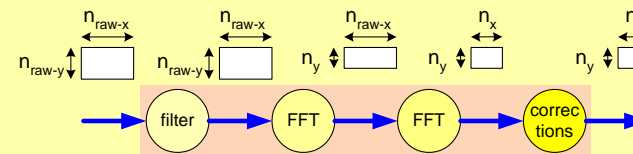
functional model



subsystem decomposition



information model



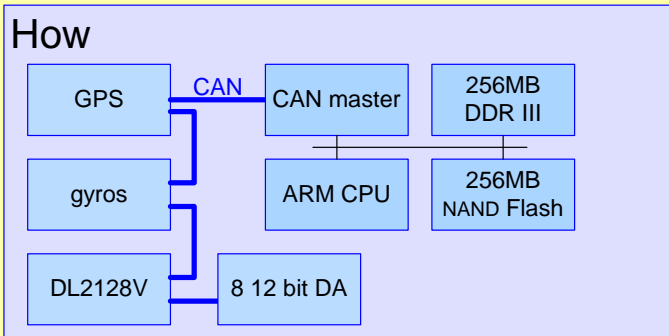
$$\begin{aligned}
 t_{recon} = & t_{filter}(n_{raw-x}, n_{raw-y}) + \\
 & n_{raw-x} * (t_{fft}(n_{raw-y}) + t_{col-overhead}) + \\
 & n_y * (t_{fft}(n_{raw-x}) + t_{row-overhead}) + \\
 & t_{corrections}(n_x, n_y) + \\
 & t_{control-overhead}
 \end{aligned}$$

$$t_{fft}(n) = c_{fft} * n * \log(n)$$

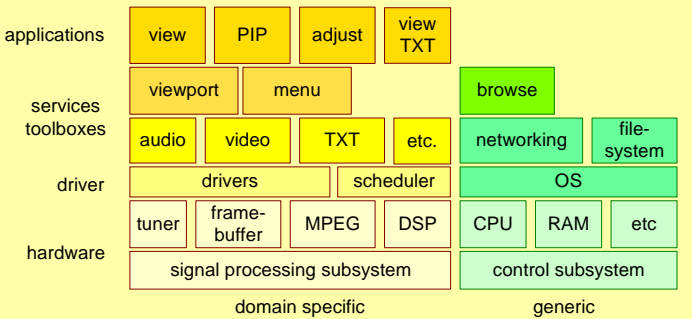
performance model

Step 5: Realization View

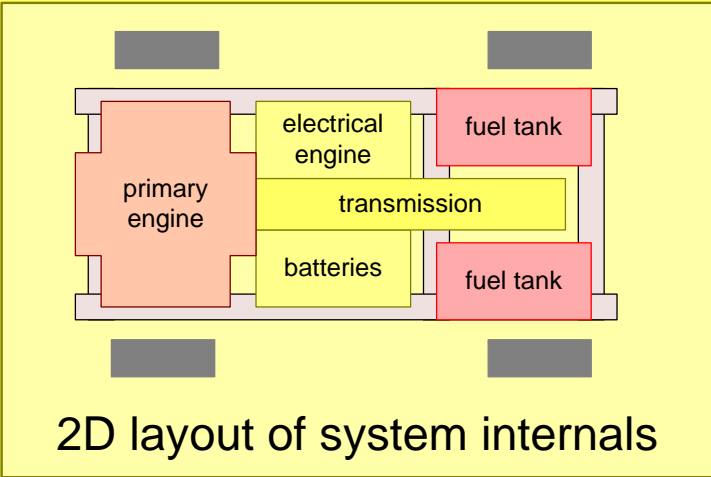
Choose 1 or 2 items from below



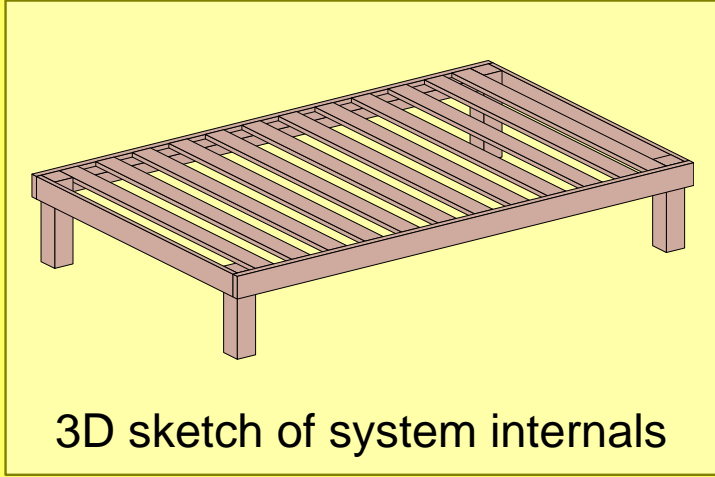
HW block diagram



SW layer diagram



2D layout of system internals



3D sketch of system internals

Annotate/mark most critical technologies or characteristics

Step 6: Qualities

Determine the 5 most relevant qualities from the checklist

- Make the chosen qualities as specific as possible
- Explain for each quality why it is relevant

Step 6: Qualities Checklist

usable

usability
attractiveness
responsiveness
image quality
wearability
storability
transportability

dependable

safety
security
reliability
robustness
integrity
availability

effective

throughput or
productivity

interoperable

connectivity
3rd party extendible

liable

liability
testability
traceability
standards compliance

efficient

resource utilization
cost of ownership

consistent

reproducibility
predictability

serviceable

serviceability
configurability
installability

future proof

evolvability
portability
upgradeability
extendibility
maintainability

logistics friendly

manufacturability
logistics flexibility
lead time

ecological

ecological footprint
contamination
noise
disposability

down to earth attributes

cost price
power consumption
consumption rate
(water, air,
chemicals,
et cetera)
size, weight
accuracy

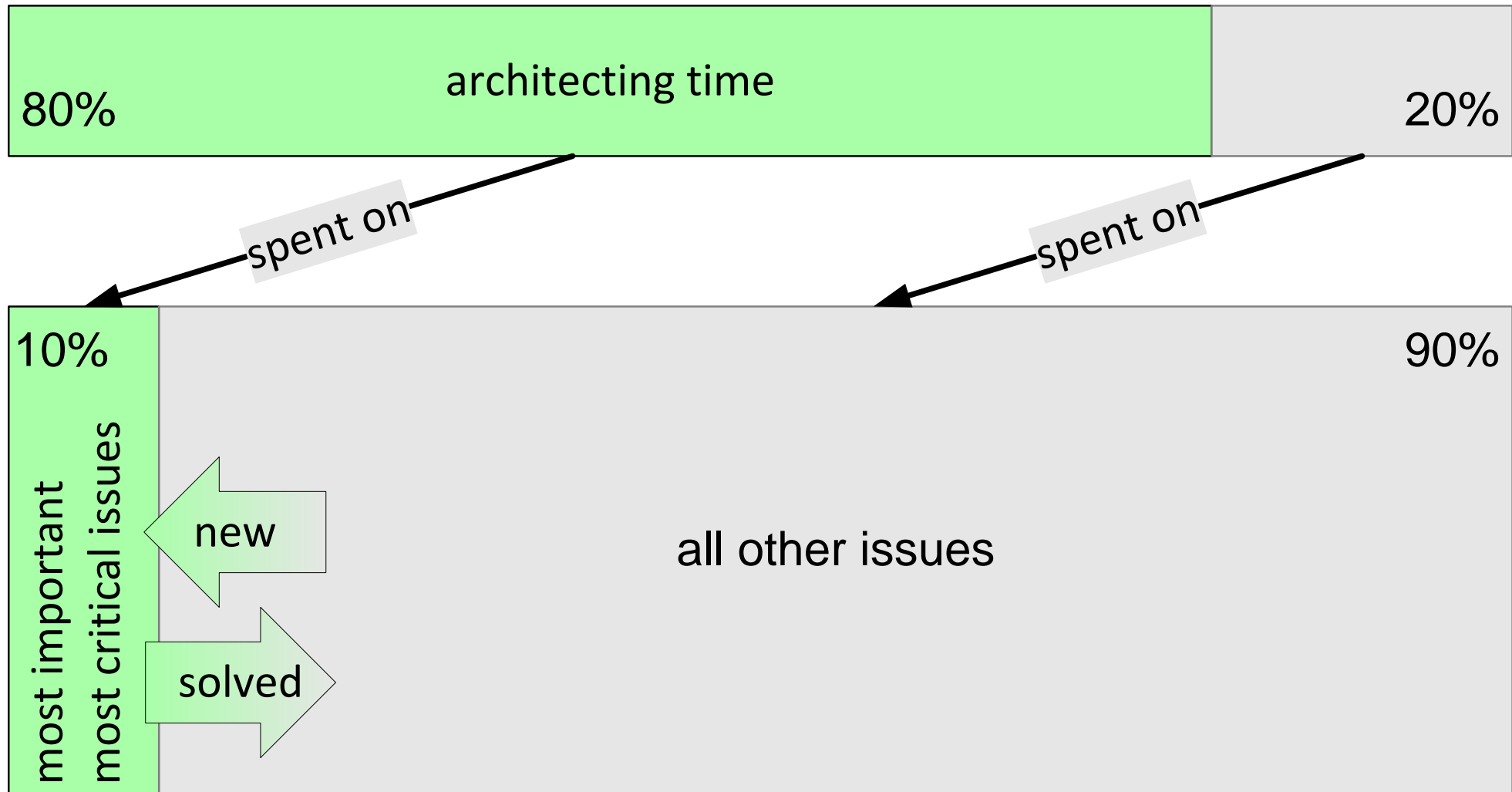
Presentation

Present the results top-down

Use two to three flip charts of the six that have been created.

Explain in five minutes the needs of the customer, the system, and the major design choices.

Architect: Focus on most Important Issues



SMART Requirements

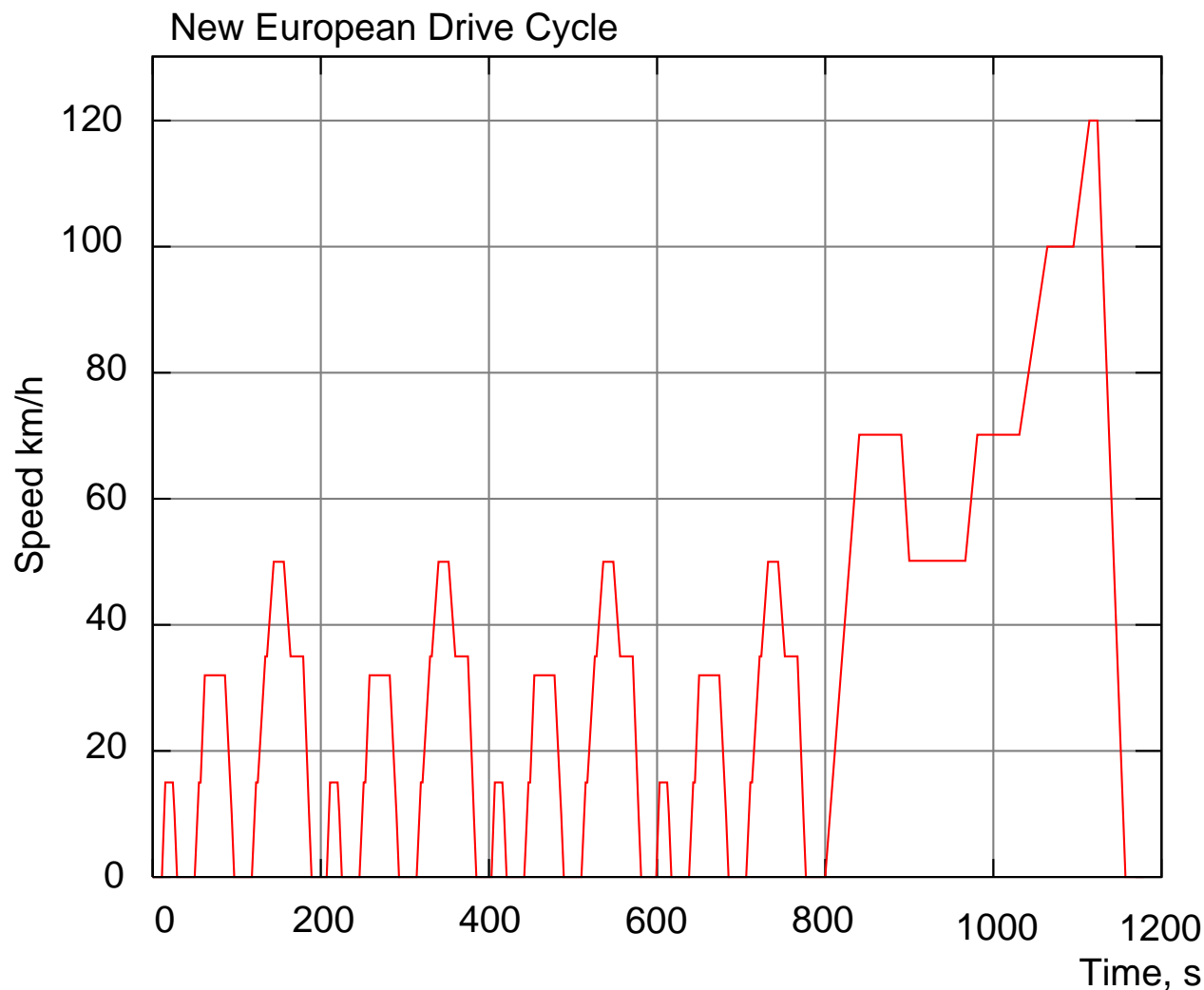
- **S**pecific quantified
- **M**easurable verifiable
- **A**chievable (Attainable, Action oriented, Acceptable, Agreed-upon, Accountable)
- **R**ealistic (Relevant, Result-Oriented)
- **T**ime-bounded (Timely , Tangible, Traceable)

Quantified Use Case to Define Key Performance

Electric Vehicle Driving Range

Range = f(
v(t),
Circumstances,
Driving style,
Car load,
Charging state,
Battery age)

A quantified Use Case defines under what circumstances the EV will achieve the specified range.

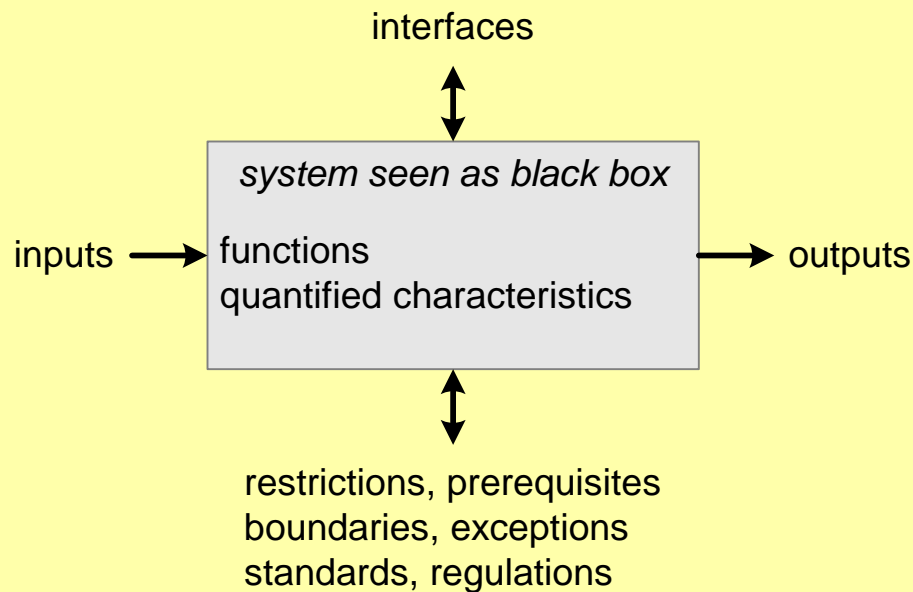


http://en.wikipedia.org/wiki/New_European_Driving_Cycle#/media/File:New_European_Driving_Cycle.svg
Published under GFDL, thanks to Orzetto

Break Out Functional View

Make specification overview with ~10 **SMART** Key Performance Parameters (or functions or interfaces)

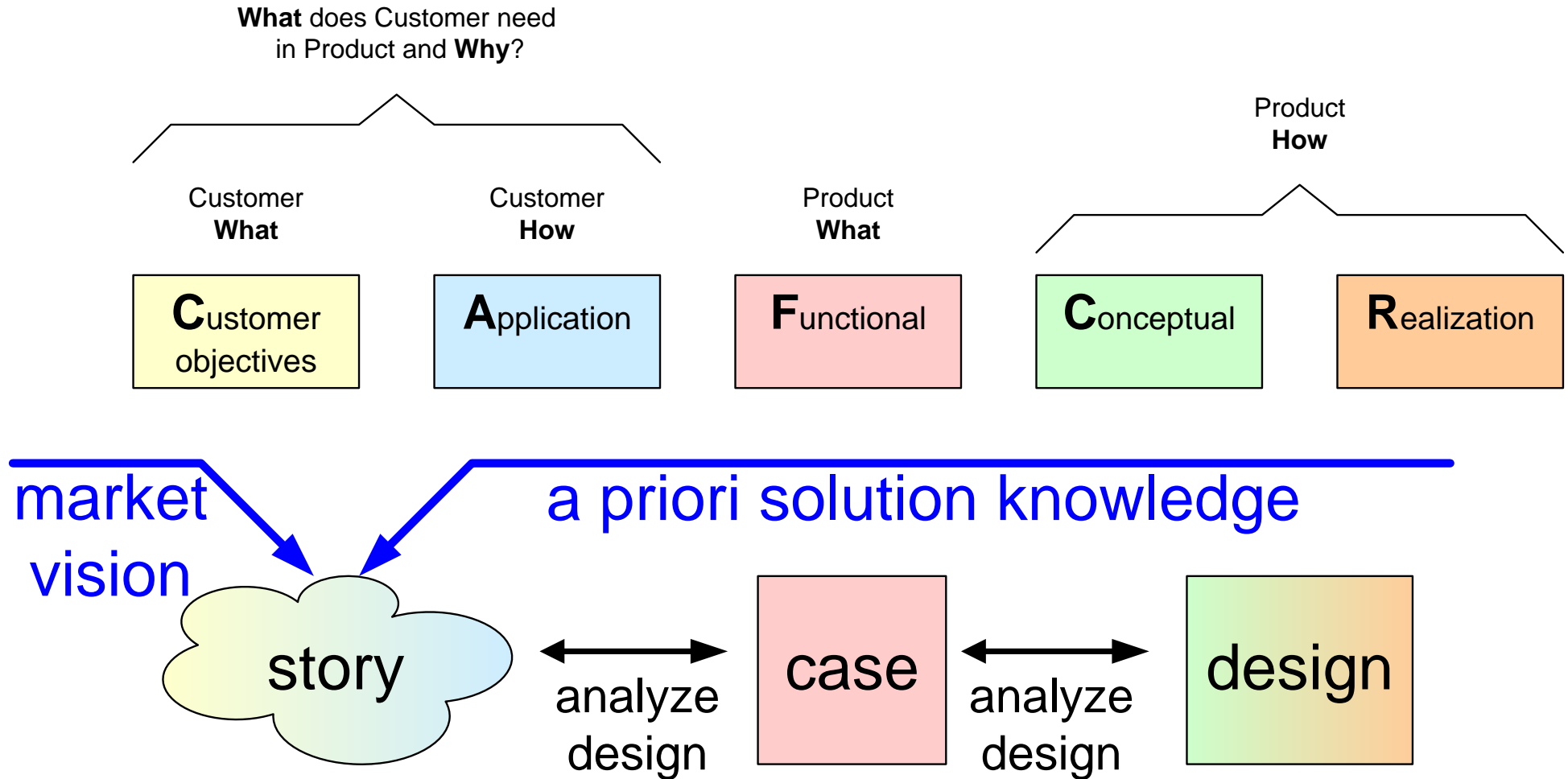
determine at least one use case



use case
typical use with relevant
context data (quantified!)

- **S**pecific *quantified*
- **M**easurable *verifiable*
- **A**chievable (Attainable, Action oriented, Acceptable, Agreed-upon, Accountable)
- **R**ealistic (Relevant, Result-Oriented)
- **T**ime-bounded (Timely, Tangible, Traceable)

From story to design



Criteria for a good story

Customer objectives
Application

- accessible, understandable

"Do you see it in front of you?"

Customer objectives
Application

- valuable, appealing

attractive, important

"Are customers queuing up for this?"

Conceptual
Realization

- critical, challenging

"What is difficult in the realization?"

"What do you learn w.r.t. the design?"

Application

- frequent, no exceptional niche

"Does it add significantly to the bottom line?"

Application
Functional

- specific

names, ages, amounts, durations, titles, ...

Exercise Story Telling

Create a story

as text + sketch or as cartoon

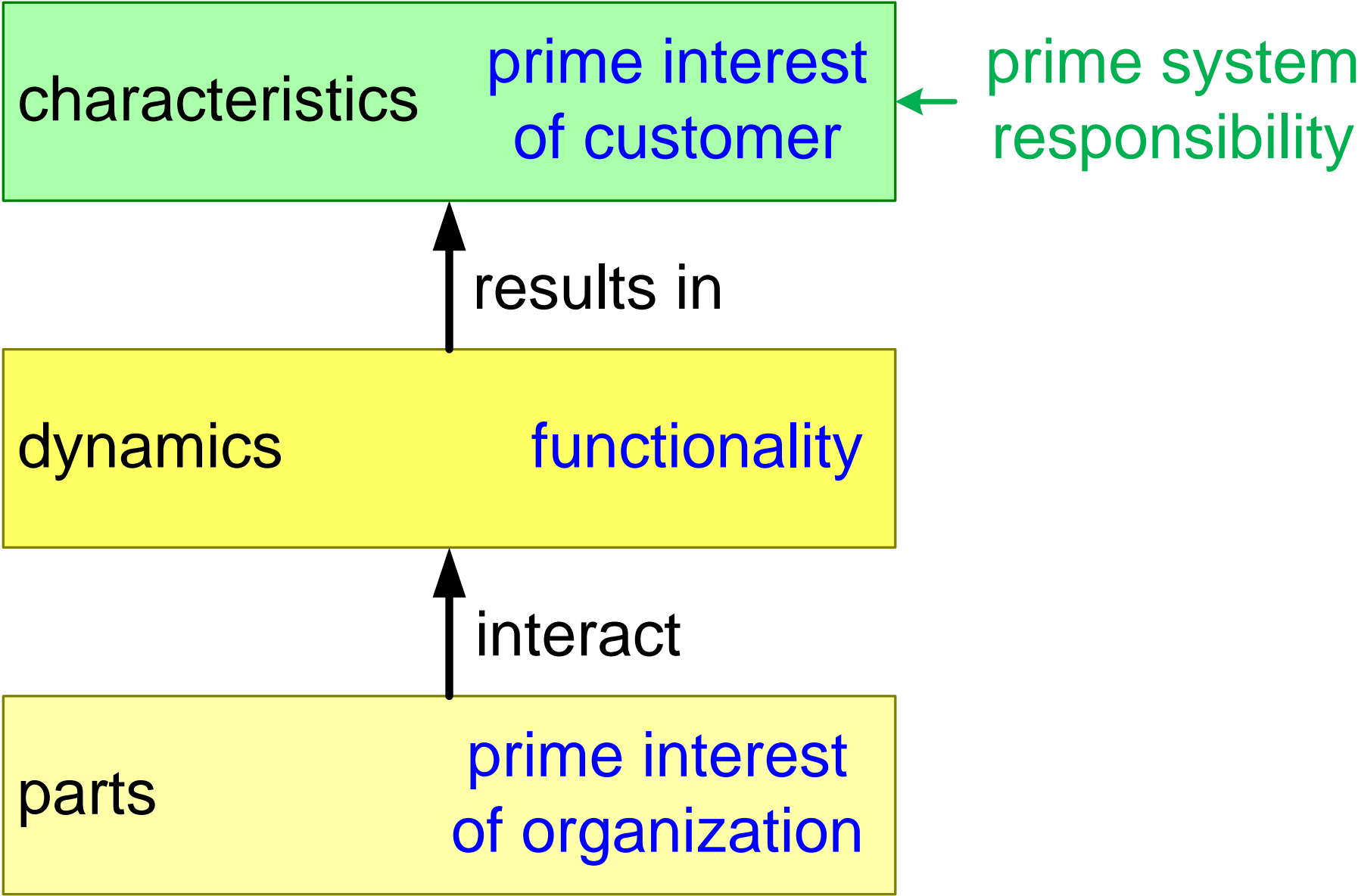
Use the criteria

be highly specific!

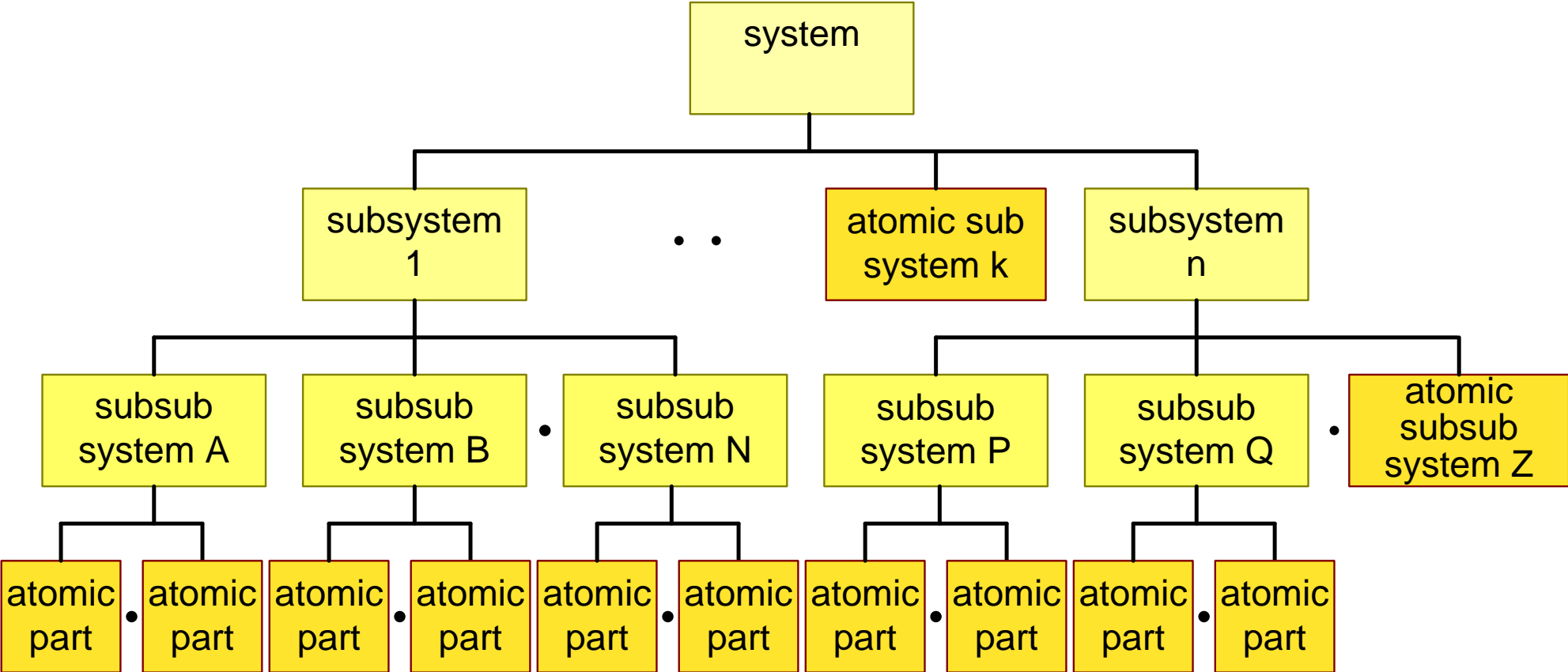
envision the future value proposition

Enjoy!

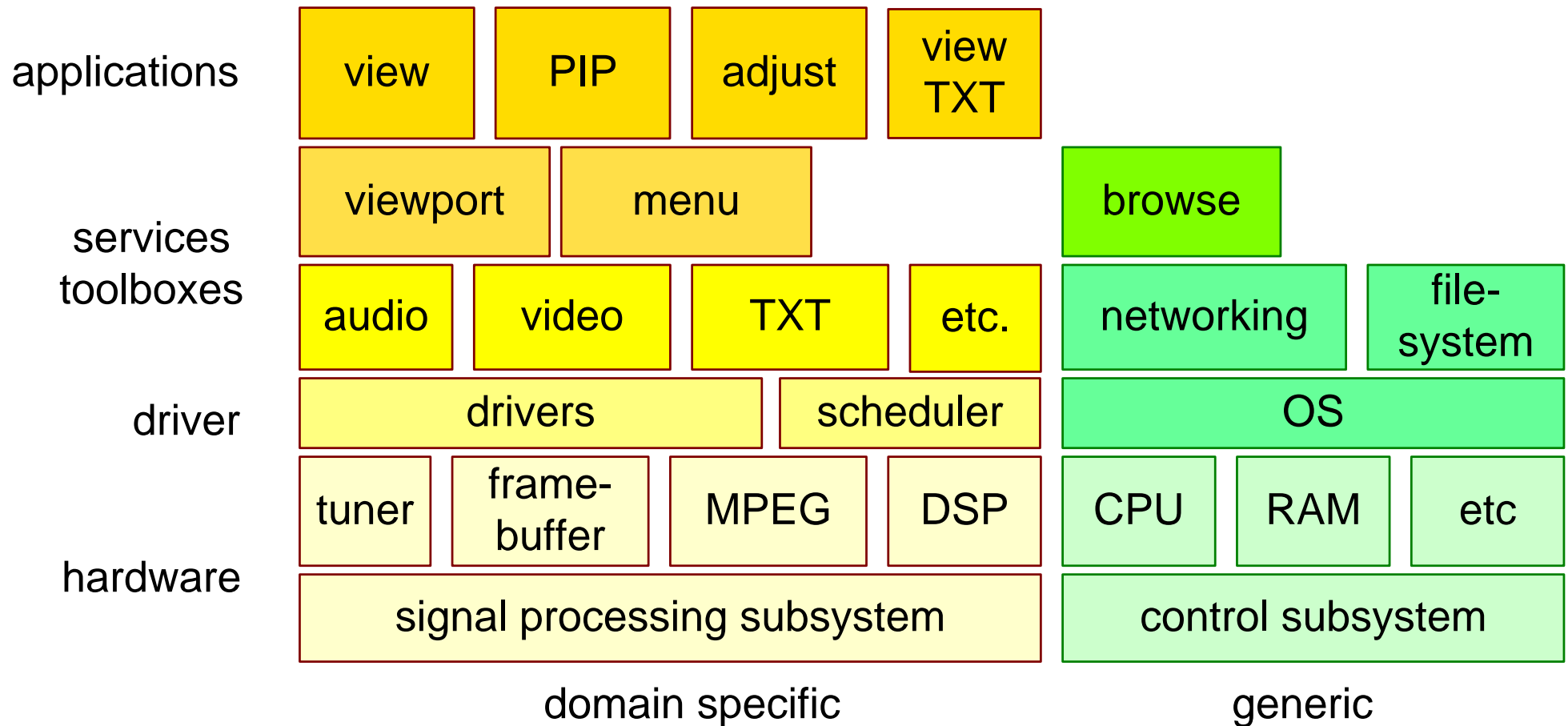
Parts, Dynamics, Characteristics



Partitioning is Applied Recursively



Software Partitioning



the part is cohesive

functionality and technology belongs together

the coupling with other parts is minimal

minimize interfaces

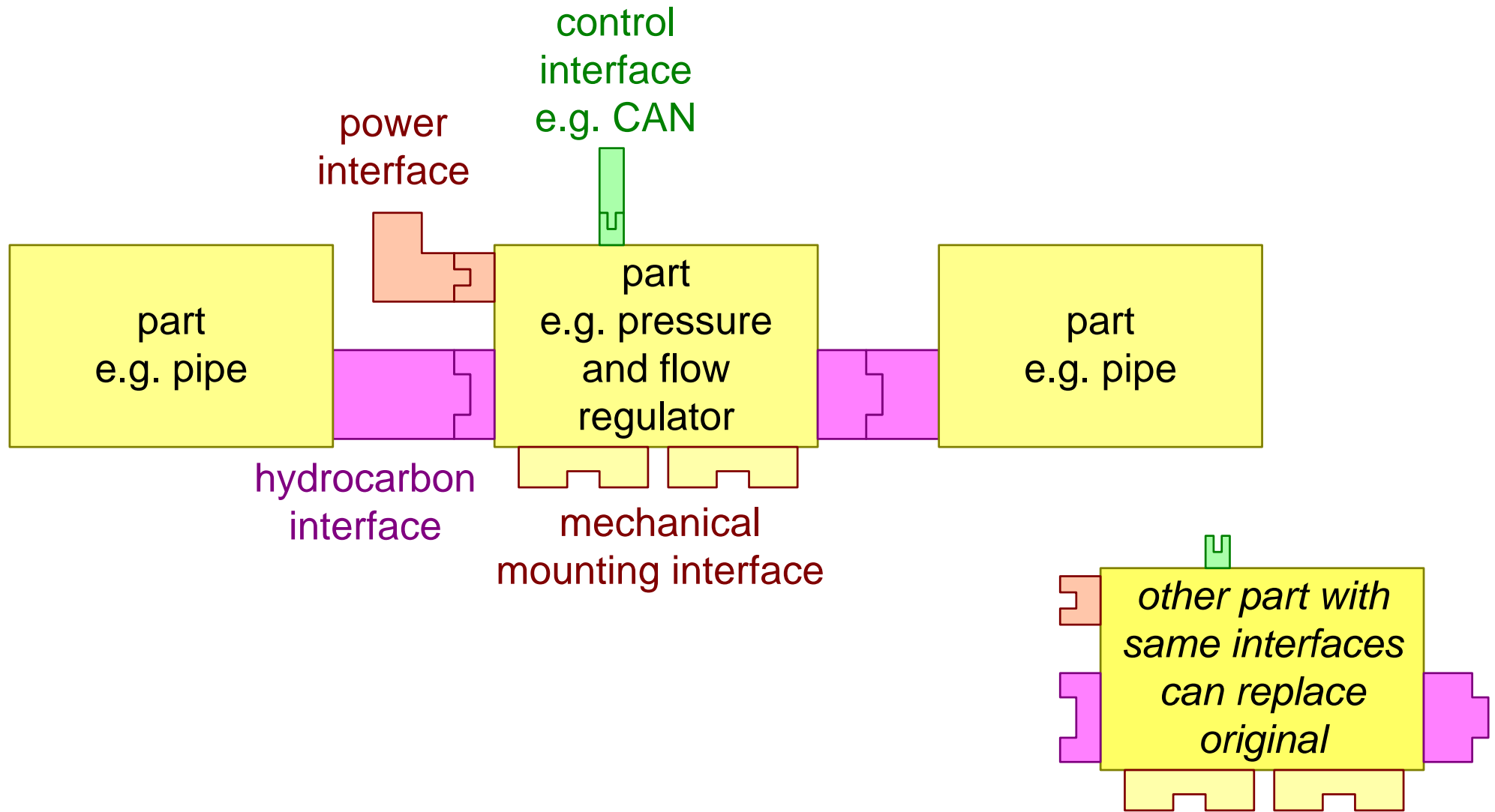
the part is selfsustained for production and qualification

can be in conflict with cost or space requirements

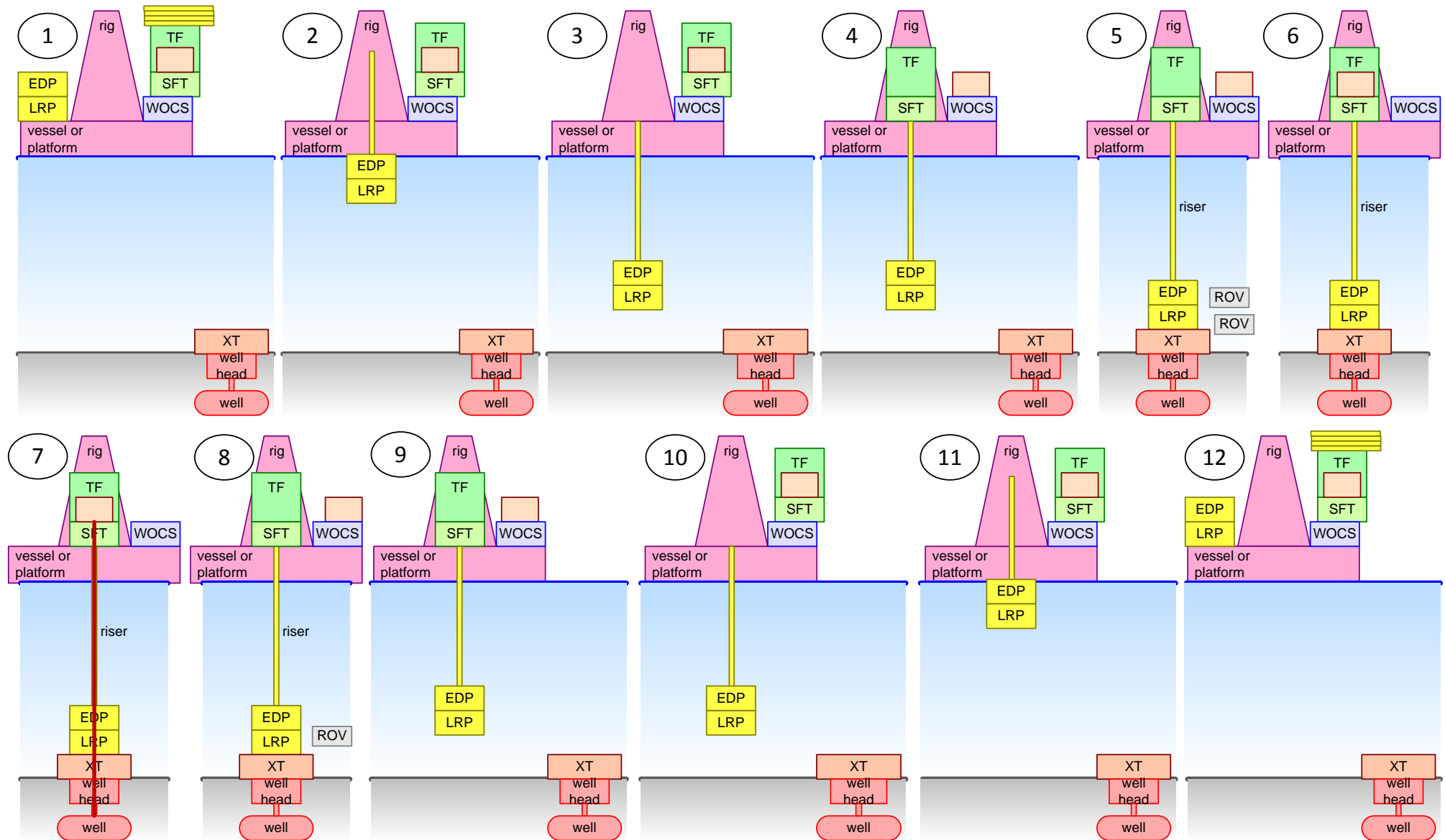
clear ownership of part

e.g. one department or supplier

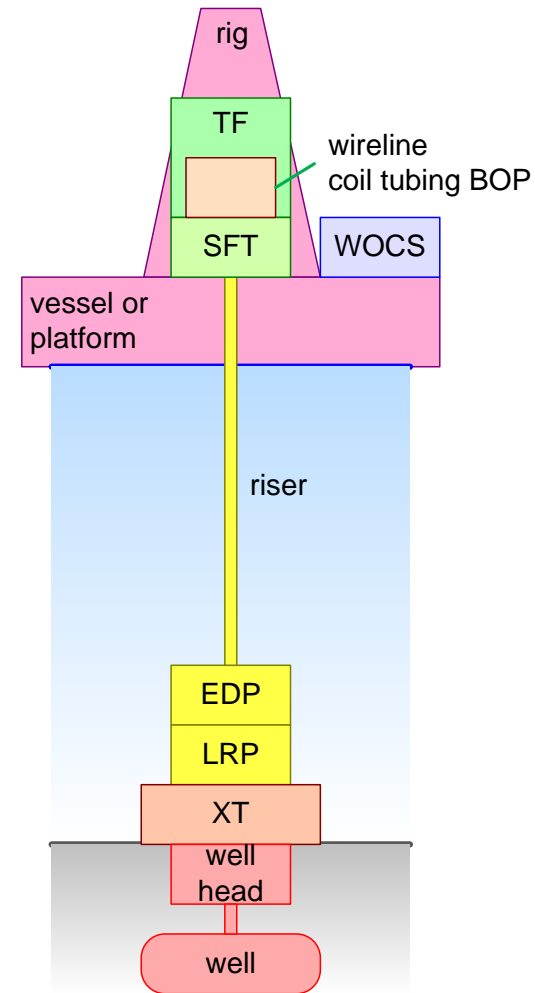
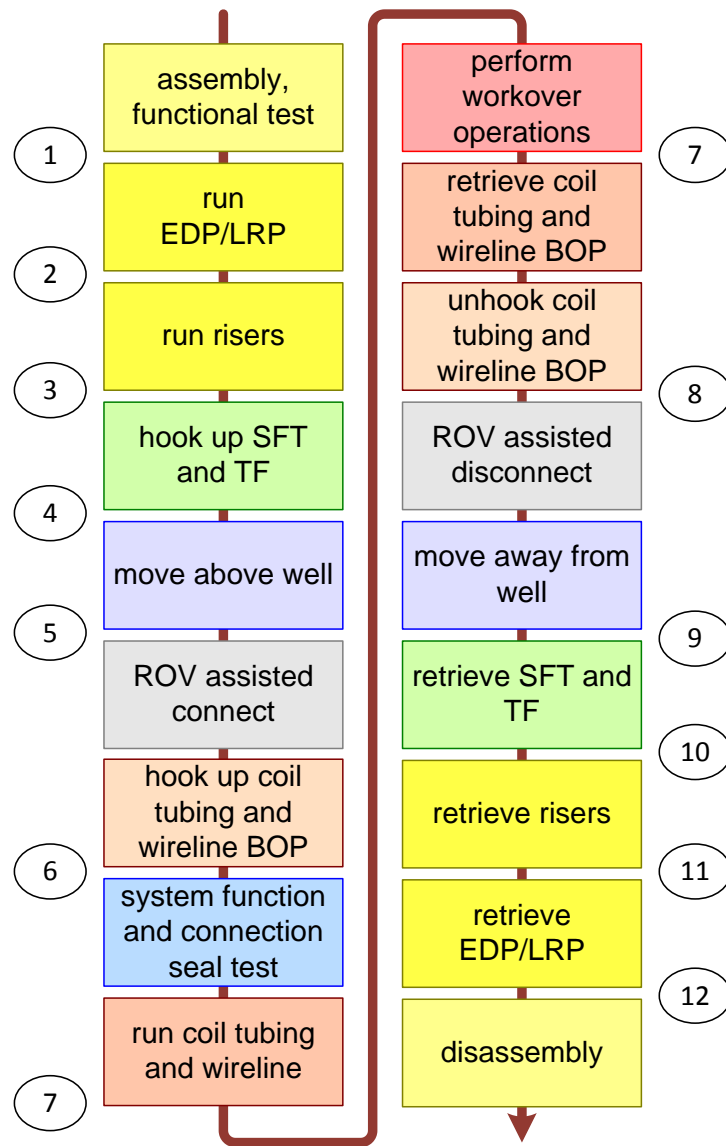
Decoupling via Interfaces



Typical Workover Operation as Cartoon

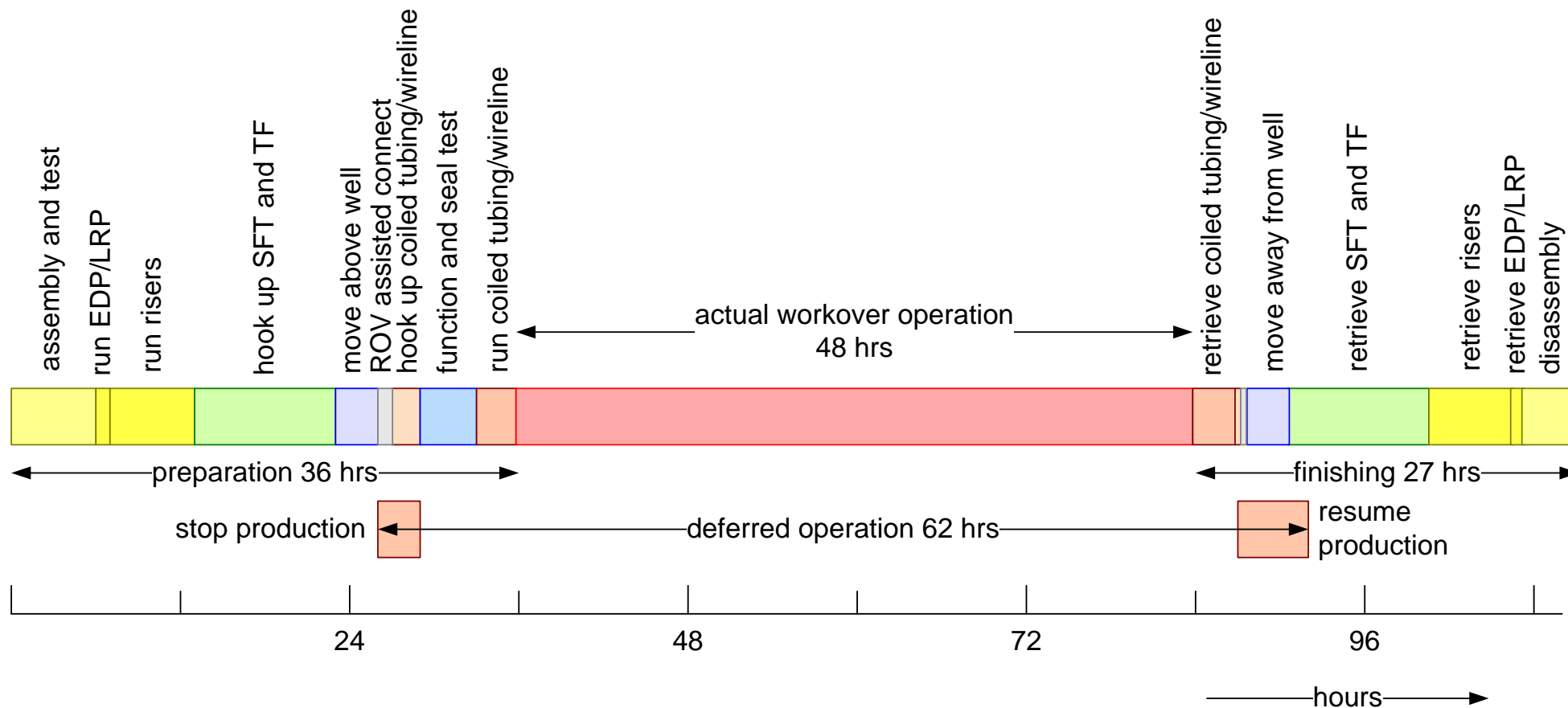


Typical Workover Operation



Typical Workover Operation on Timeline

assumptions:
 running and retrieving risers: 50m/hr
 running and retrieving coiled tubing/wireline: 100m/hr
 depth: 300m



Functional Decomposition

How does the system work and operate?

Functions describe *what* rather than *how*.

Functions are *verbs*.

Input-Process-Output paradigm.

Multiple kinds of flows:

- physical (e.g. hydrocarbons)

- information (e.g. measurements)

- control

At lower level one part \sim one function

- pump pumps, compressor compresses, controller controls

At higher level functions are complex interplay of physical parts

- e.g. regulating constant flow, pressure and temperature

Exercise Dynamic Behavior

Capture the **dynamic behavior** of the **internals** of your system in **multiple** diagrams.

Diagrams that capture dynamic behavior are among others:

- Functional flow (of control or information, material or goods, or energy)
- Activity or sequence diagrams (e.g. with “swimming lanes”)
- State diagrams

What is a Budget?

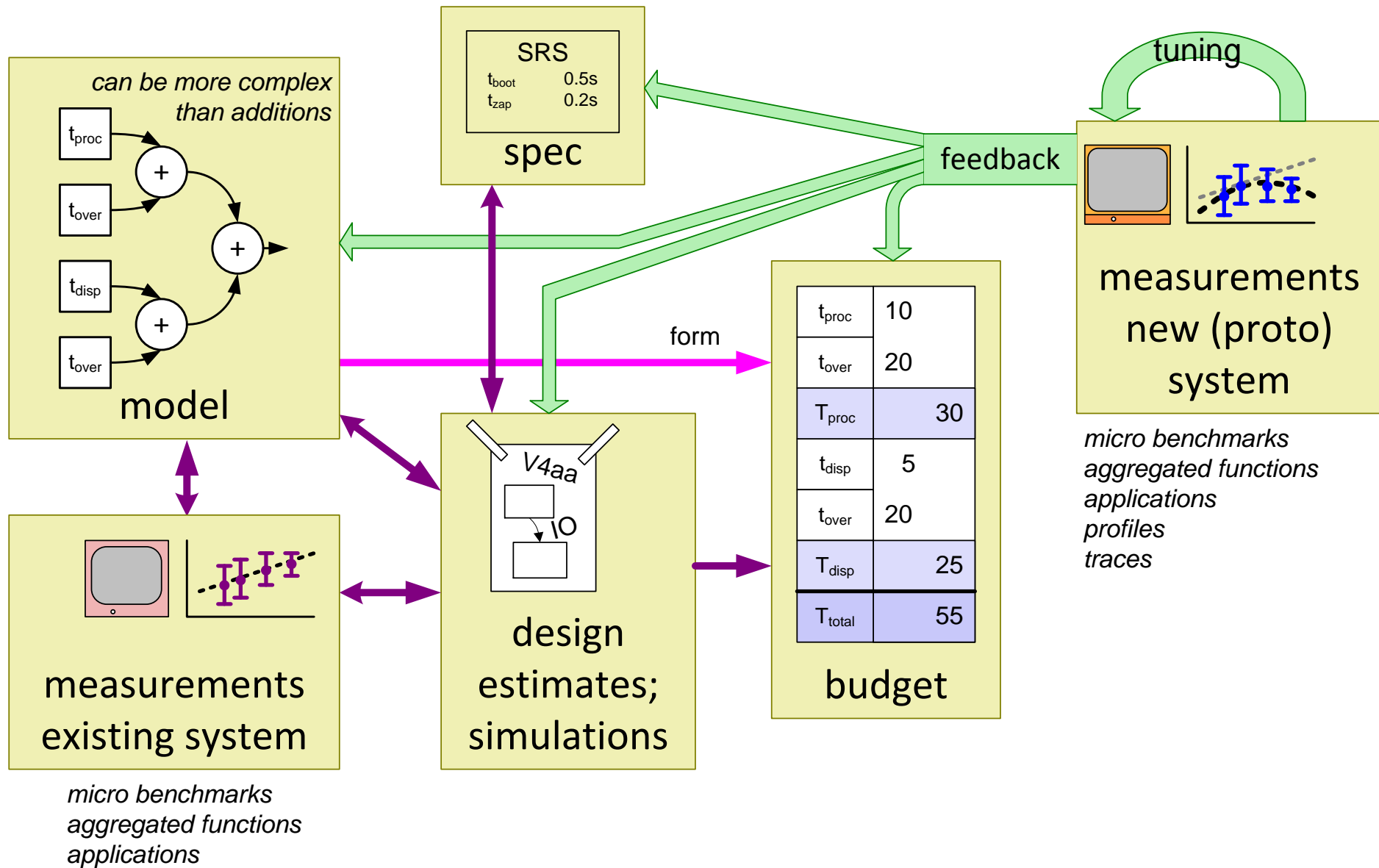
A **budget** is
a **quantified instantiation** of a **conceptual model**

A **budget** can
prescribe or **describe** the **contributions**
by **parts** of the **solution**
to the **system quality** under consideration

Why Budgets?

- to make the design explicit
- to provide a baseline to take decisions
- to specify the requirements for the detailed designs
- to have guidance during integration
- to provide a baseline for verification
- to manage the design margins explicitly

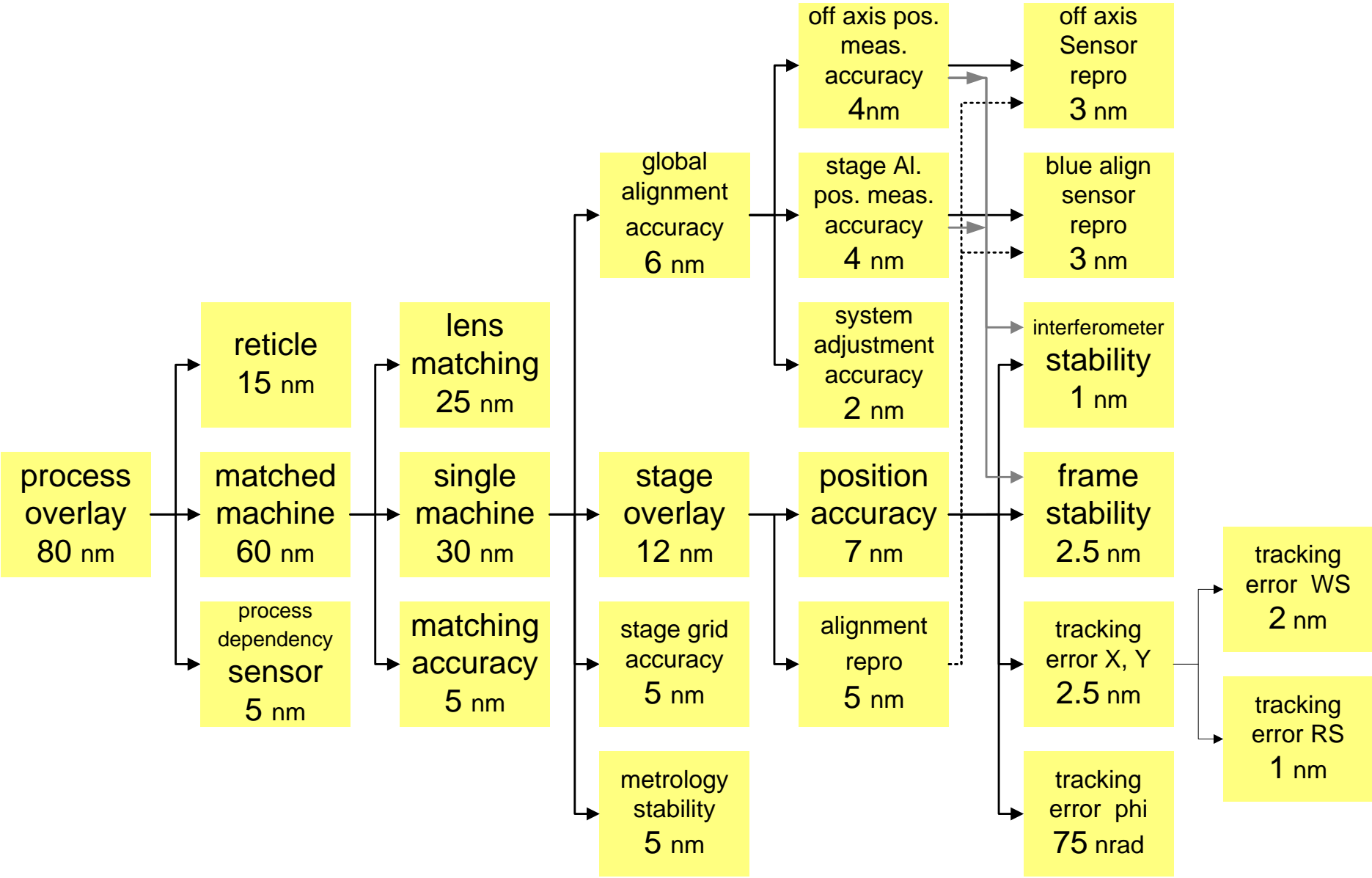
Visualization of Budget Based Design Flow



Stepwise Budget Based Design Flow

step	example
1A measure old systems	micro-benchmarks, aggregated functions, applications
1B model the performance starting with old systems	flow model and analytical model
1C determine requirements for new system	response time or throughput
2 make a design for the new system	explore design space, estimate and simulate
3 make a budget for the new system:	models provide the structure measurements and estimates provide initial numbers specification provides bottom line
4 measure prototypes and new system	micro-benchmarks, aggregated functions, applications profiles, traces
5 Iterate steps 1B to 4	

Budgets Applied on Waferstepper Overlay



Budgets Applied on Medical Workstation Memory Use

<i>memory budget in Mbytes</i>	code	obj data	bulk data	total
shared code	11.0			11.0
User Interface process	0.3	3.0	12.0	15.3
database server	0.3	3.2	3.0	6.5
print server	0.3	1.2	9.0	10.5
optical storage server	0.3	2.0	1.0	3.3
communication server	0.3	2.0	4.0	6.3
UNIX commands	0.3	0.2	0	0.5
compute server	0.3	0.5	6.0	6.8
system monitor	0.3	0.5	0	0.8
application SW total	13.4	12.6	35.0	61.0
UNIX Solaris 2.x				10.0
file cache				3.0
total				74.0

Summary of Budgeting

A budget is a quantified instantiation of a model

A budget can prescribe or describe the contributions by parts of the solution to the system quality under consideration

A budget uses a decomposition in tens of elements

The numbers are based on historic data, user needs, first principles and measurements

Budgets are based on models and estimations

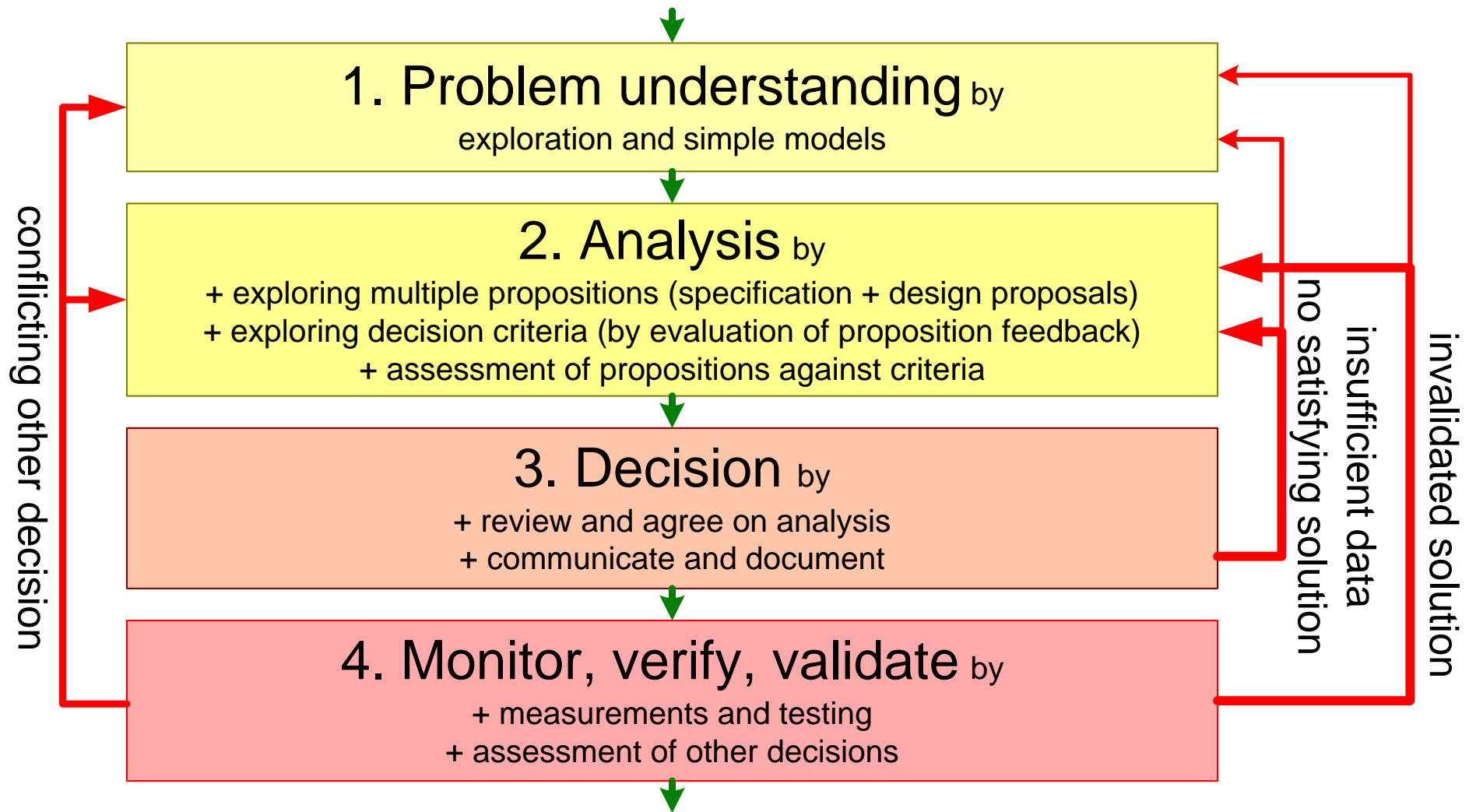
Budget visualization is critical for communication

Budgeting requires an incremental process

Many types of budgets can be made; start simple!

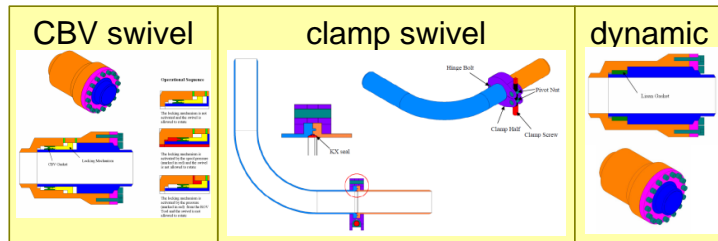
Problem Solving Approach

vague problem statement



Examples of Pugh Matrix Application

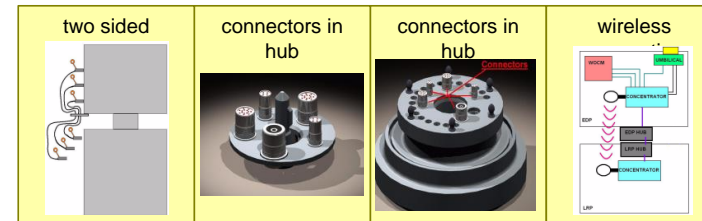
Swivel concept selection



evaluation criteria	weight	CBV		clamp		dynamic		
Maturity	10	5	50	2	20	2	50	
Development level								
Cost	20	4	80	2	40	5	100	
Hardware cost								
Development cost		5	100	2	40	2	40	
Design robustness	25							
Design life								
swivel cycles		5	125	3	75	3	75	
pressure cycles		5	125	4	100	5	125	
Pressure range								
internal		4	100	4	100	4	100	
external		2	50	5	125	2	50	
Temperature range		4	100	4	100	4	100	
Installation	20							
Initial installatio/retrieval			2	40	3	60	4	80
Connection/disconnection		2	40	4	80	5	100	
Operation	25							
Swivel resistance			1	25	4	100	5	125
Spool Length Short			1	25	4	100	5	125
Spool Length Long			3	75	5	125	5	125
Hub loads			2	50	4	100	5	125
Σ points			985		1165		1290	

from master paper Halvard Bjørnsen, 2009

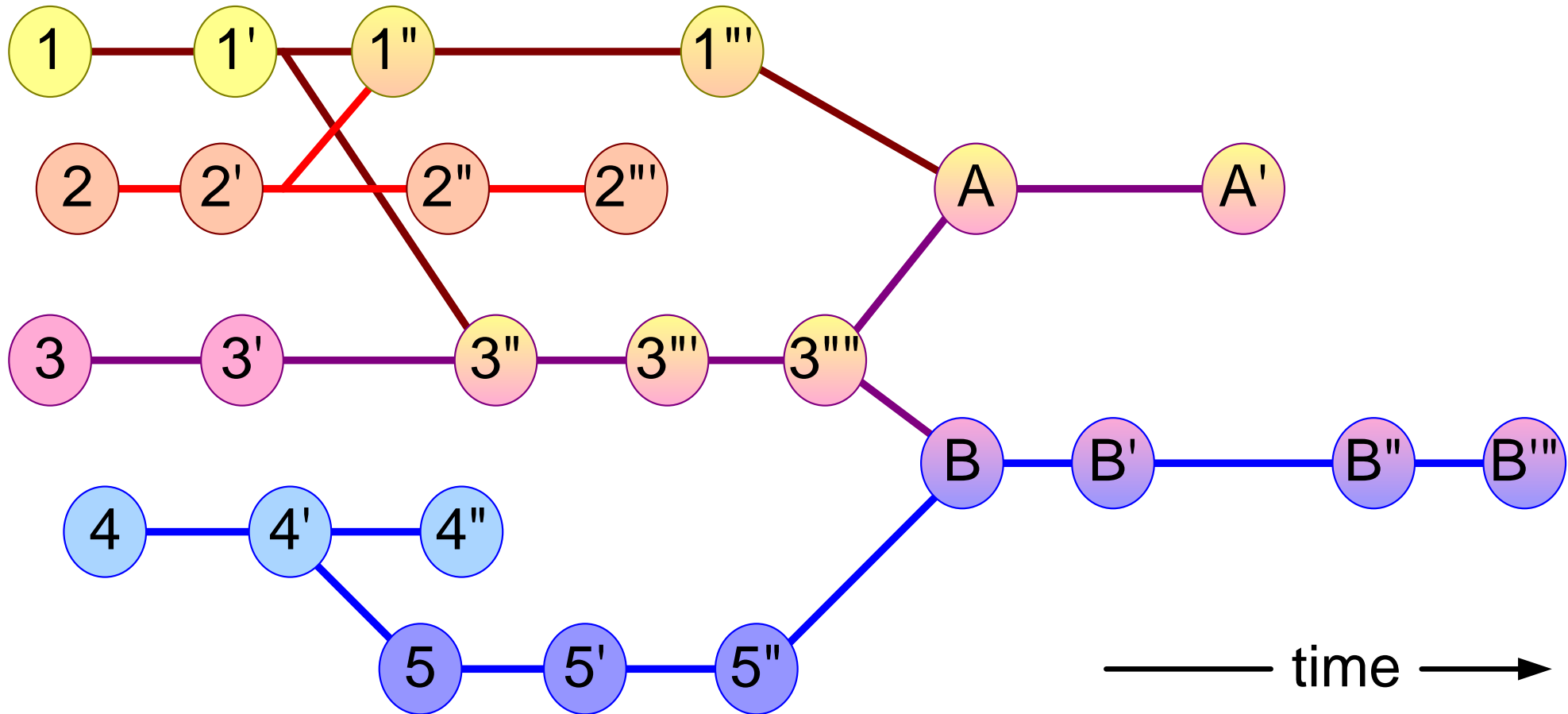
EDP-LRP connection



Evaluation Criteria	Score	Concepts			
		1	2	3	4
Time to connect					
Need for ROV		-	+	+	+
Design		-	+	+	+
Robustness					
Connector design		-	S	S	+
Number of parts		-	-	+	+
Handle roll-off		+	-	S	+
Influence other		+	S	-	S
Redundancy					
Design		+	-	-	S
Interchangeability		+	-	-	-
Cost					
HW cost		-	-	-	-
Manufacturing cost		S	S	-	S
Engineering cost		+	-	S	-
Service cost		-	+	+	+
Maturity		-	-	S	+
Σ -		7	7	5	3
Σ S		1	3	4	3
Σ +		5	3	4	7
Pos.		3	4	2	1

from master paper Dag Jostein Klever, 2009




Evolution of Design Options



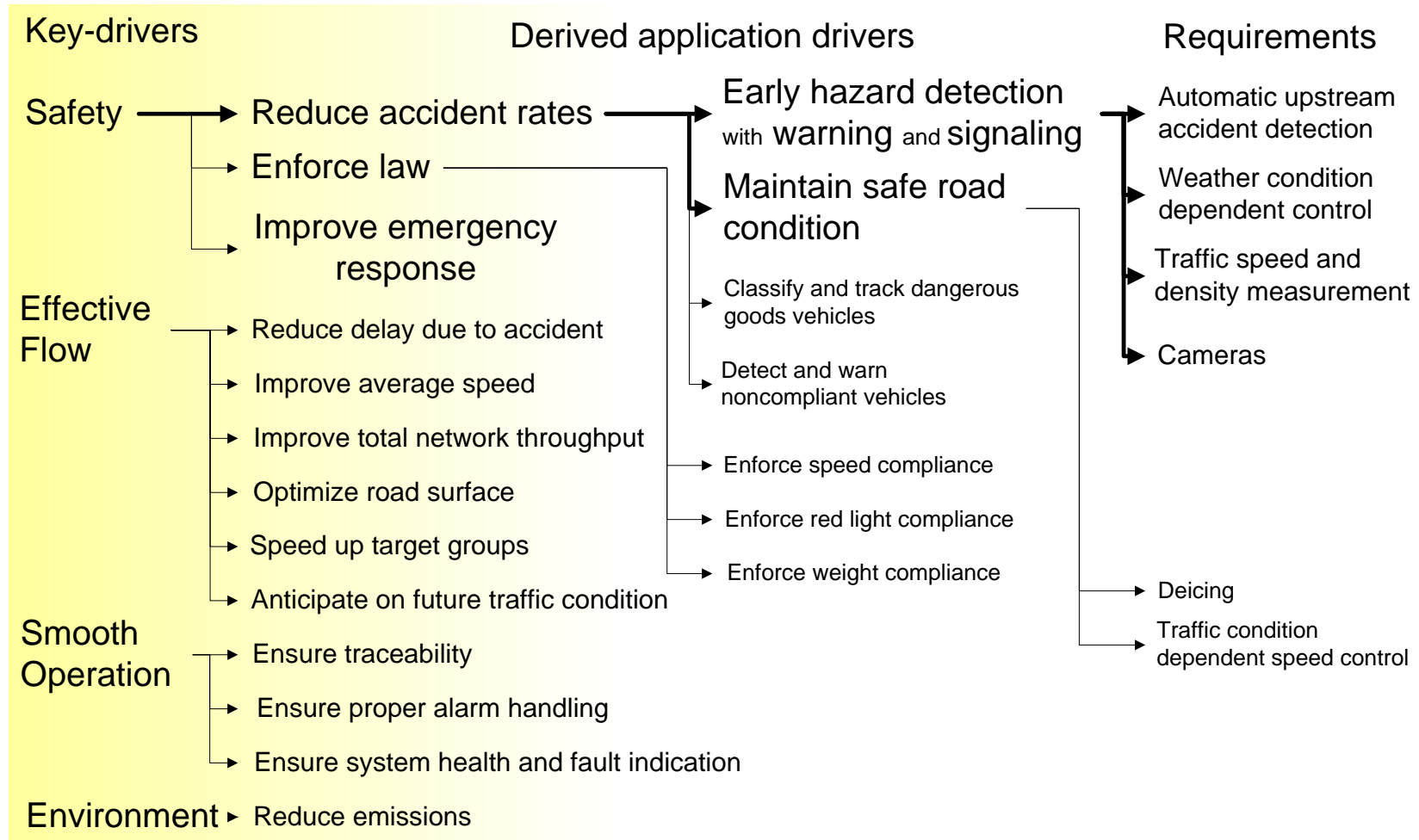
Exercise Budgeting

Make a **decision matrix** for one of the **concept selections**.

- define at least 3 concepts
- define 7 to 10 criteria for selection
- score the concepts against the criteria, for example using a scale from 1 to 5: 1 = very poor, 5 = very good
- recommend a concept with a rationale

	concept 1 	concept 2 	concept 3 
critterion 1	vulnerable MTTR 1	robust 30 hours 3	robust access 5
critterion n	reusing platform 4	reusing platform 4	resource shortage 2
			best, because ...

Example Motorway Management Analysis



Note: the graph is only partially elaborated for application drivers and requirements

Method to create Key Driver Graph

- | | |
|--|--|
| • Define the scope specific. | in terms of stakeholder or market segments |
| • Acquire and analyze facts | extract facts from the product specification
and ask why questions about the specification of existing products. |
| • Build a graph of relations between drivers and requirements
by means of brainstorming and discussions | where requirements
may have multiple drivers |
| • Obtain feedback | discuss with customers, observe their reactions |
| • Iterate many times | increased understanding often triggers the move of issues
from driver to requirement or vice versa and rephrasing |

Recommendation for the Definition of Key Drivers

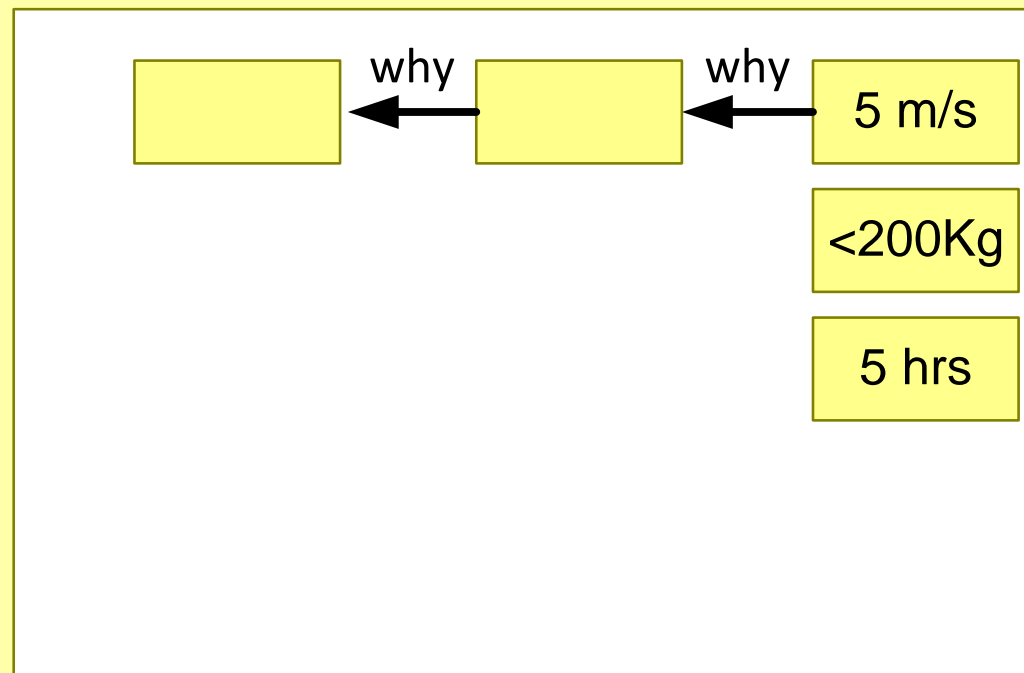
- Limit the number of key-drivers minimal 3, maximal 6
- Don't leave out the obvious key-drivers for instance the well-known main function of the product
- Use short names, recognized by the customer.
- Use market-/customer- specific names, no generic names for instance replace “ease of use” by “minimal number of actions for experienced users”, or “efficiency” by “integral cost per patient”
- Do not worry about the exact boundary between Customer Objective and Application create clear goal means relations

Exercise Customer Key Driver Graph

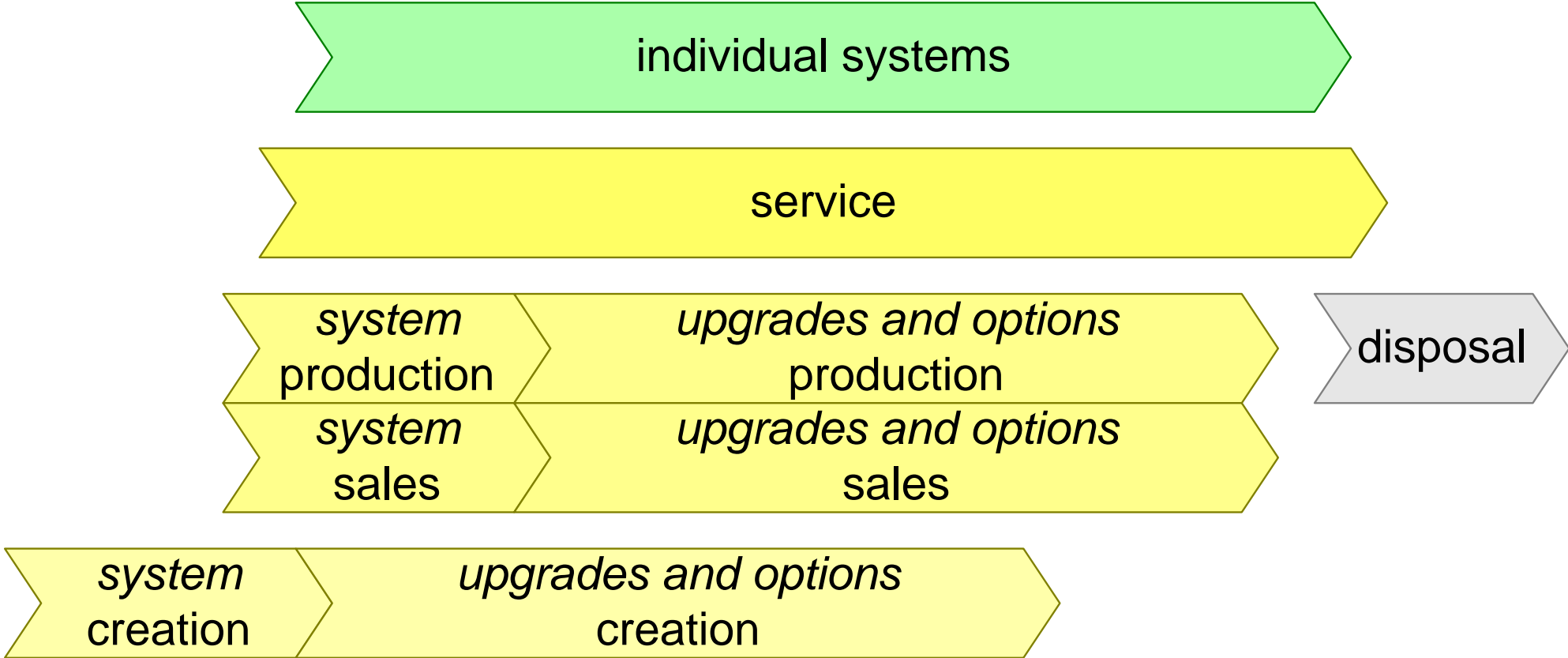
Make a **customer key driver graph**

Use yellow note stickers

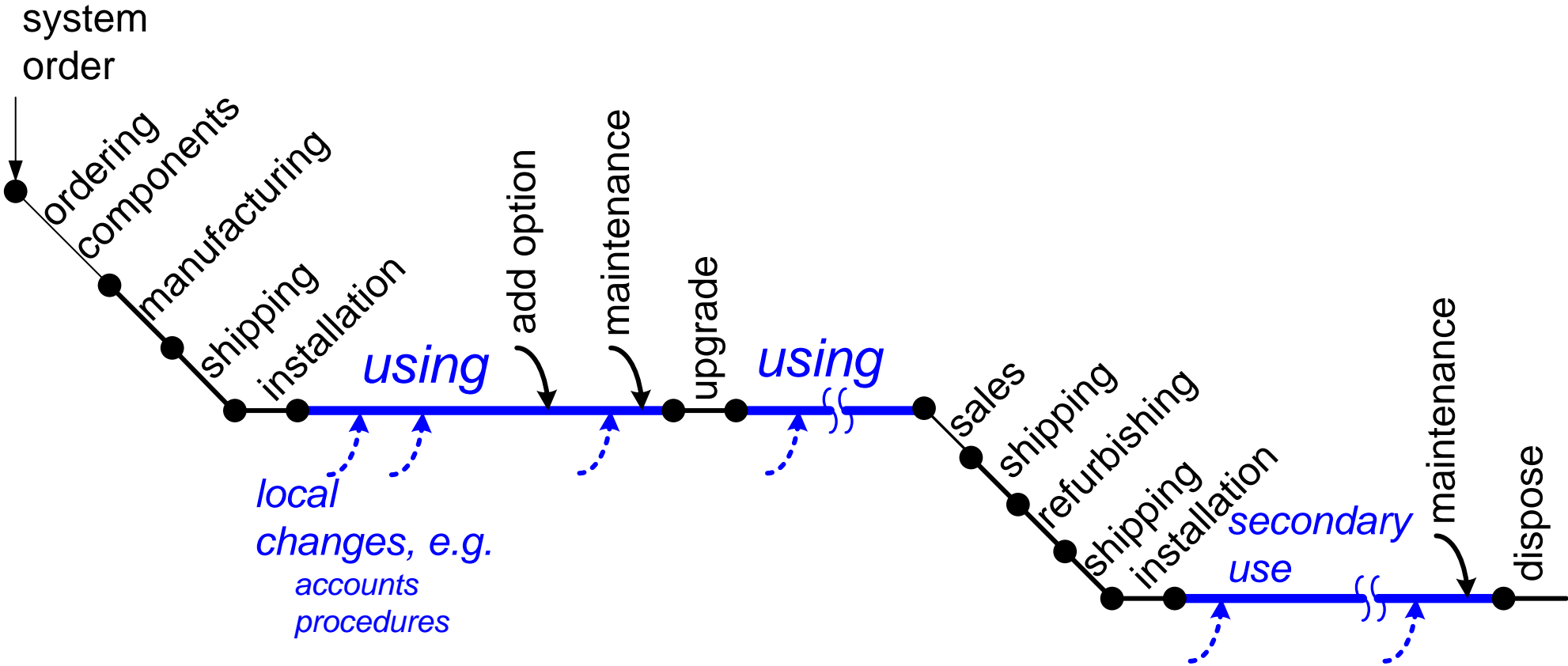
Start at the right hand side



Product Related Life Cycles



System Life Cycle



Approach to Life Cycle Modeling

Identify potential life cycle changes and sources	
Characterize time aspect of changes	how often how fast
Determine required effort	amount type
Determine impact of change on system and context	performance reliability
Analyse risks	business

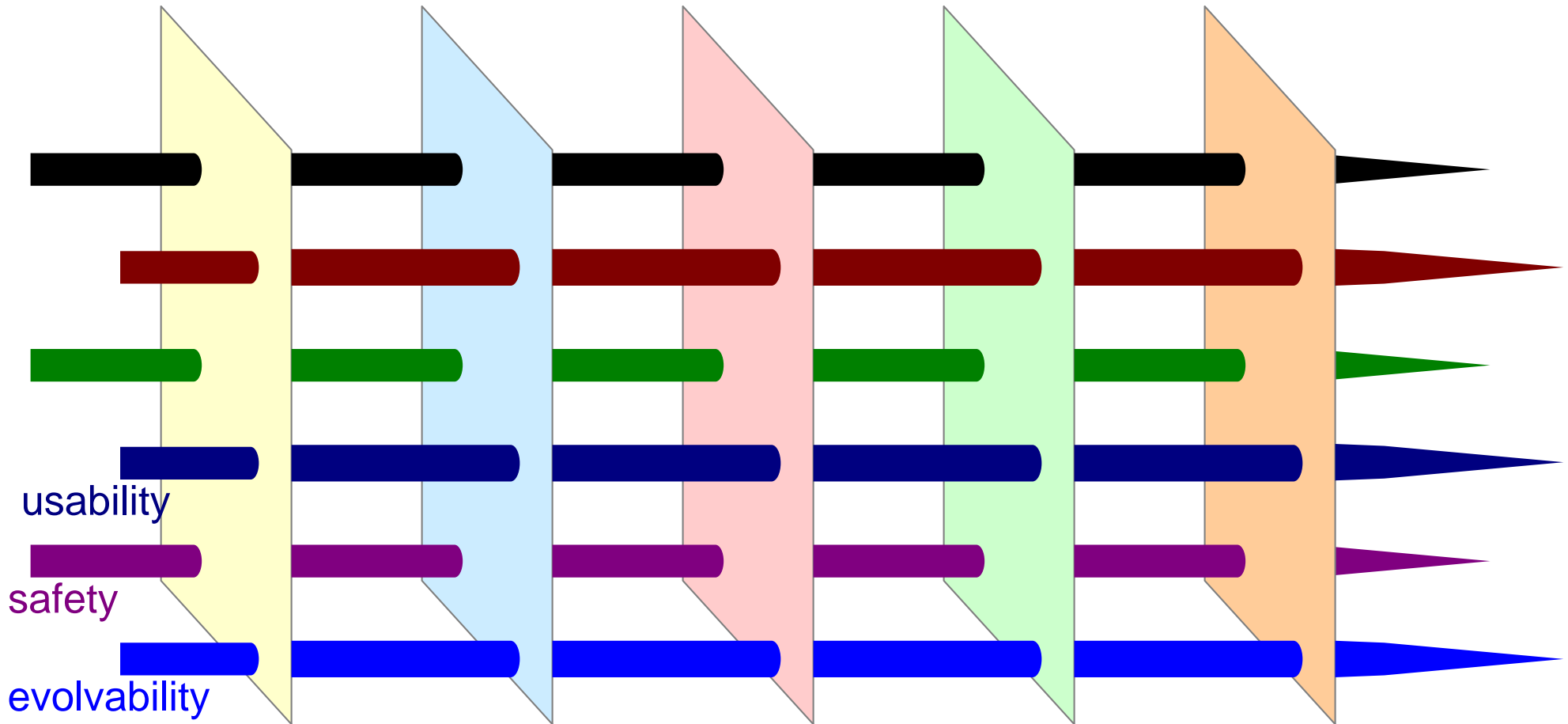
see reasoning

Exercise Life Cycle View

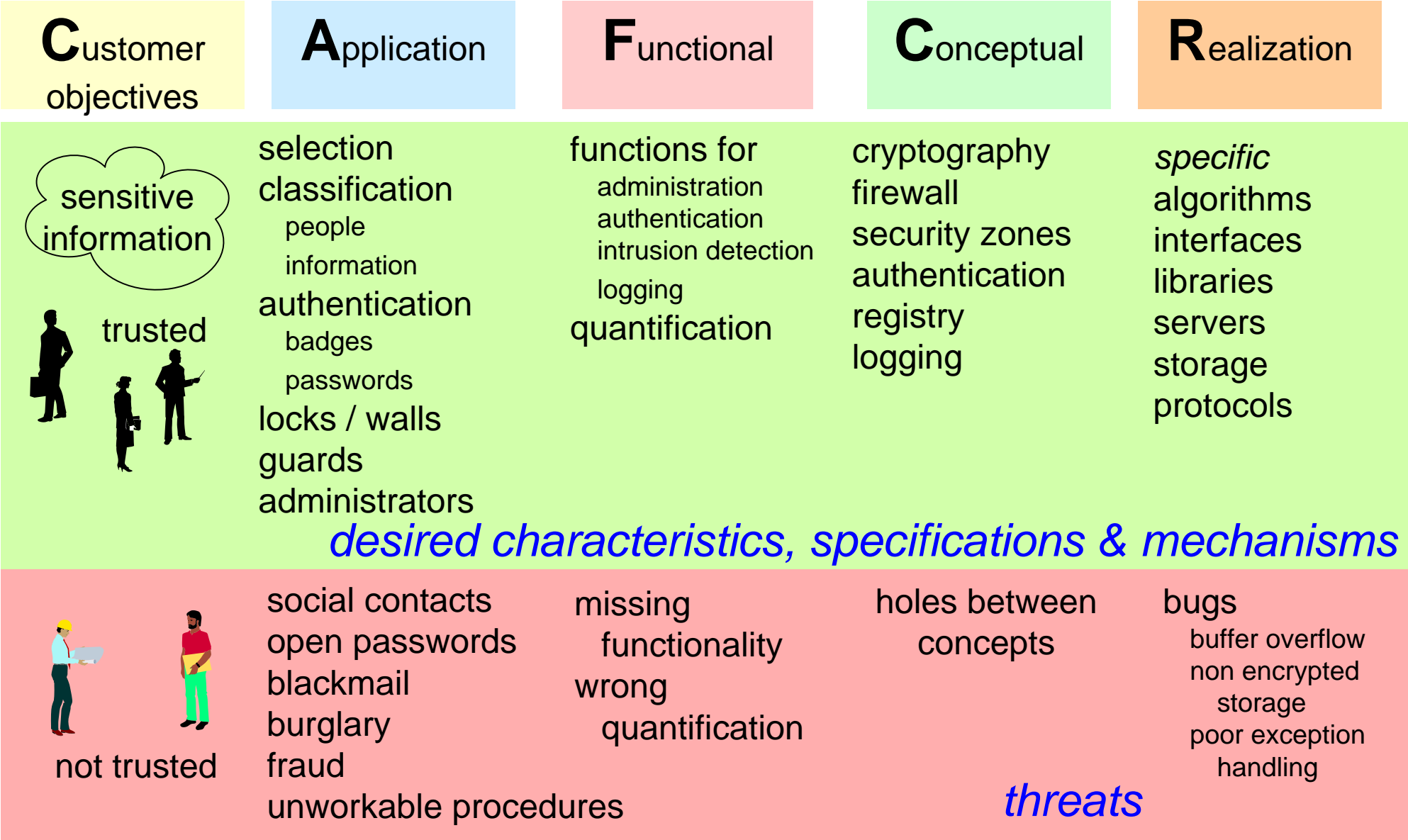
Analyze the **evolution** during the **lifecycle**.

- identify sources of change in customer context, life cycle context, and technology
- make a list of changes
- determine per change the expected rate of change and the required response time to the change
- optional: determine effort, impact, and risks per change

Quality needles as generic integrating concepts



Security as example through all views



Quality Checklist

usable

usability
attractiveness
responsiveness
image quality
wearability
storability
transportability

dependable

safety
security
reliability
robustness
integrity
availability

effective

throughput or
productivity

interoperable

connectivity
3rd party extendible

liable

liability
testability
traceability
standards compliance

efficient

resource utilization
cost of ownership

consistent

reproducibility
predictability

serviceable

serviceability
configurability
installability

future proof

evolvability
portability
upgradeability
extendibility
maintainability

logistics friendly

manufacturability
logistics flexibility
lead time

ecological

ecological footprint
contamination
noise
disposability

down to earth attributes

cost price
power consumption
consumption rate
(water, air,
chemicals,
et cetera)
size, weight
accuracy

Make a **line of reasoning** for one of the dominant qualities.

- in the CA views; determine what customers do to achieve their goal
- in the F view determine the specification of your system supporting this quality
- in the CR views determine the relevant concepts and technologies
- Take the reverse viewpoints as well: what threatens this quality?

Simplistic Financial Computations for System Architects.

by *Gerrit Muller* USN-SE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

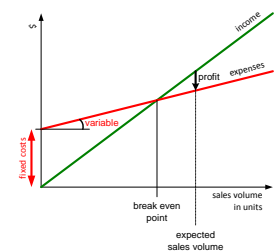
Abstract

This document explains how simple financial estimates can be made by system architects. These simplistic estimates are useful for an architect to perform sanity checks on proposals and to obtain understanding of the financial impact of proposals. Note that architects will never have full fledged financial controller know how and skills. These estimates are zero order models, but real business decisions will have to be founded on more substantial financial proposals.

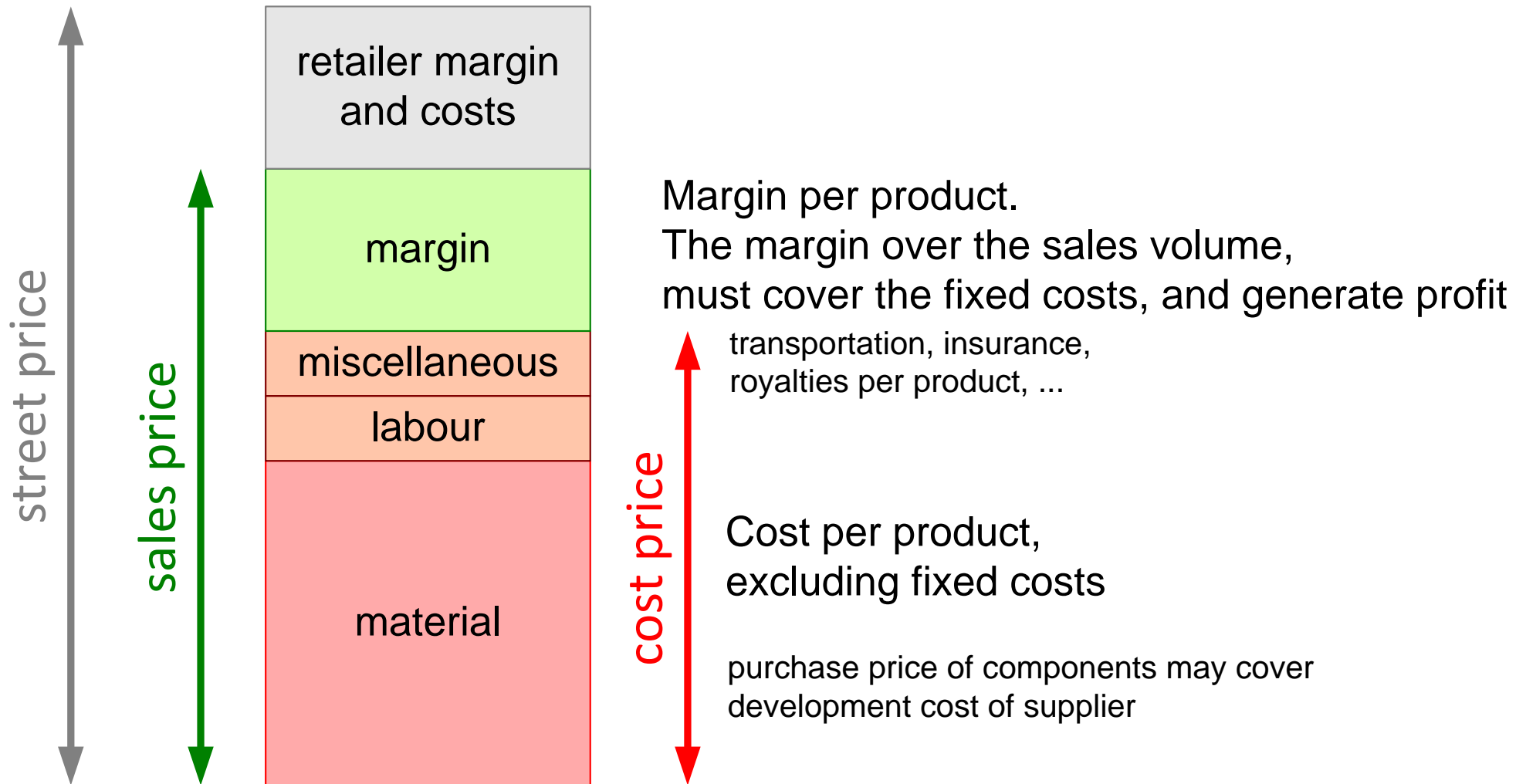
Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

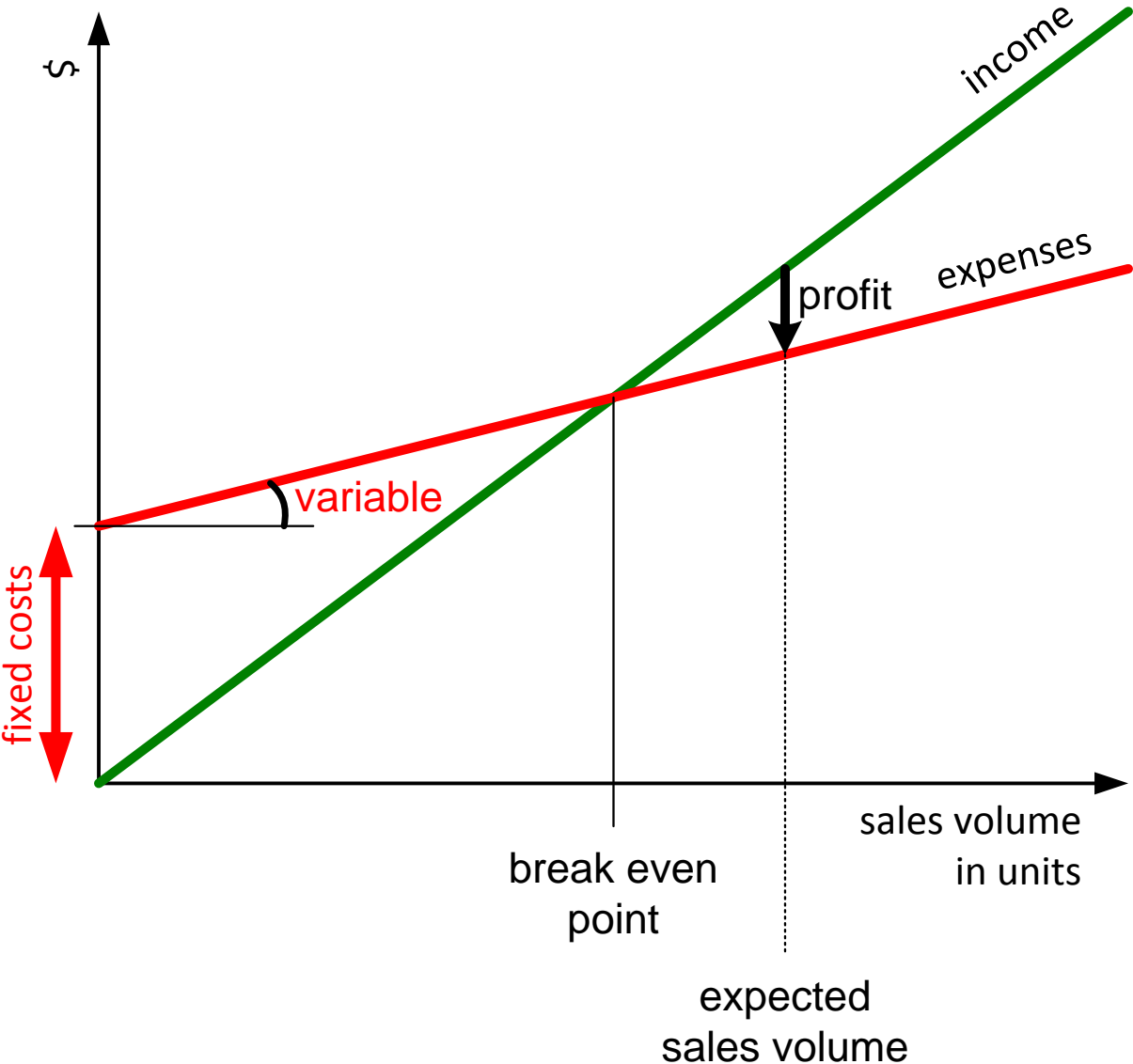
January 27, 2026
status: preliminary
draft
version: 1.3



Product Margin = Sales Price - Cost



Profit as function of sales volume



Investments, more than R&D



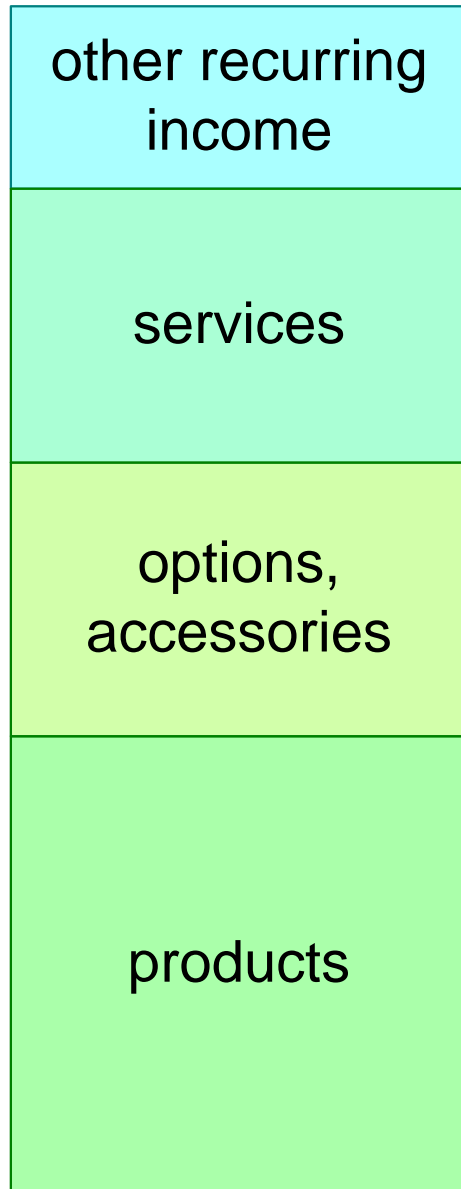
business dependent:
pharmaceuticals industry
sales cost >> R&D cost

strategic choice:
NRE or per product

including:
staff, training, tools, housing
materials, prototypes
overhead
certification

often a standard staffing rate is used
that covers most costs above:
 $R\&D\ investment = Effort * rate$

Income, more than product sales only



$$\sum_{\text{services}} \text{income}_{\text{service}}$$

$$\sum_{\text{options}} \text{sales price}_{\text{option}} * \text{volume}_{\text{option}}$$

$$\text{sales price}_{\text{product}} * \text{volume}_{\text{product}}$$

license fees
pay per movie

content, portal
updates
maintenance

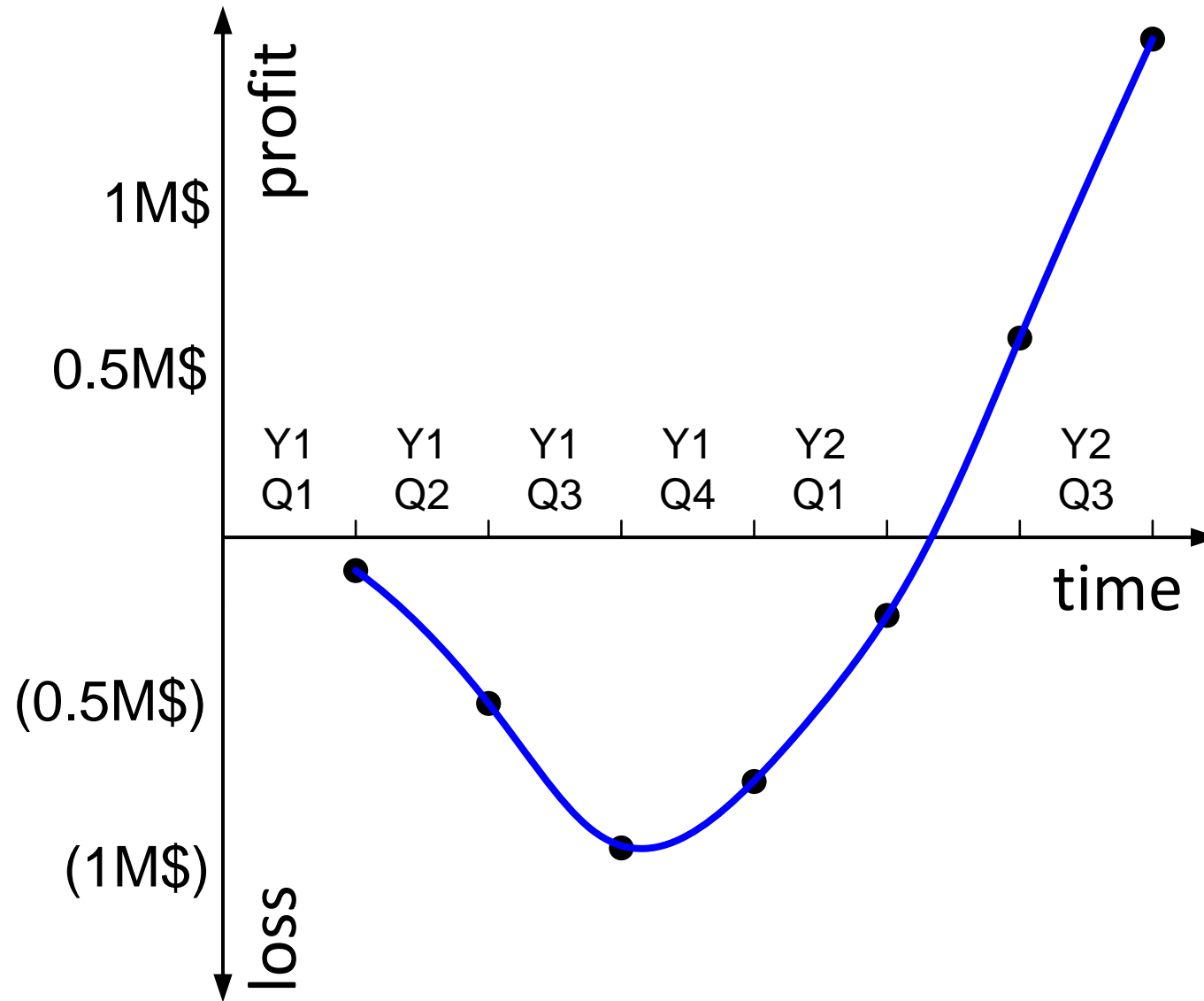
The Time Dimension

	Y1 Q1	Y1 Q2	Y1 Q3	Y1 Q4	Y2 Q1	Y2 Q2	Y2 Q3
investments	100k\$	400k\$	500k\$	100k\$	100k\$	60k\$	20k\$
sales volume (units)	-	-	2	10	20	30	30
material & labour costs	-	-	40k\$	200k\$	400k\$	600k\$	600k\$
income	-	-	100k\$	500k\$	1000k\$	1500k\$	1500k\$
quarter profit (loss)	(100k\$)	(400k\$)	(440k\$)	200k\$	500k\$	840k\$	880k\$
cumulative profit	(100k\$)	(500k\$)	(940k\$)	(740k\$)	(240k\$)	600k\$	1480k\$

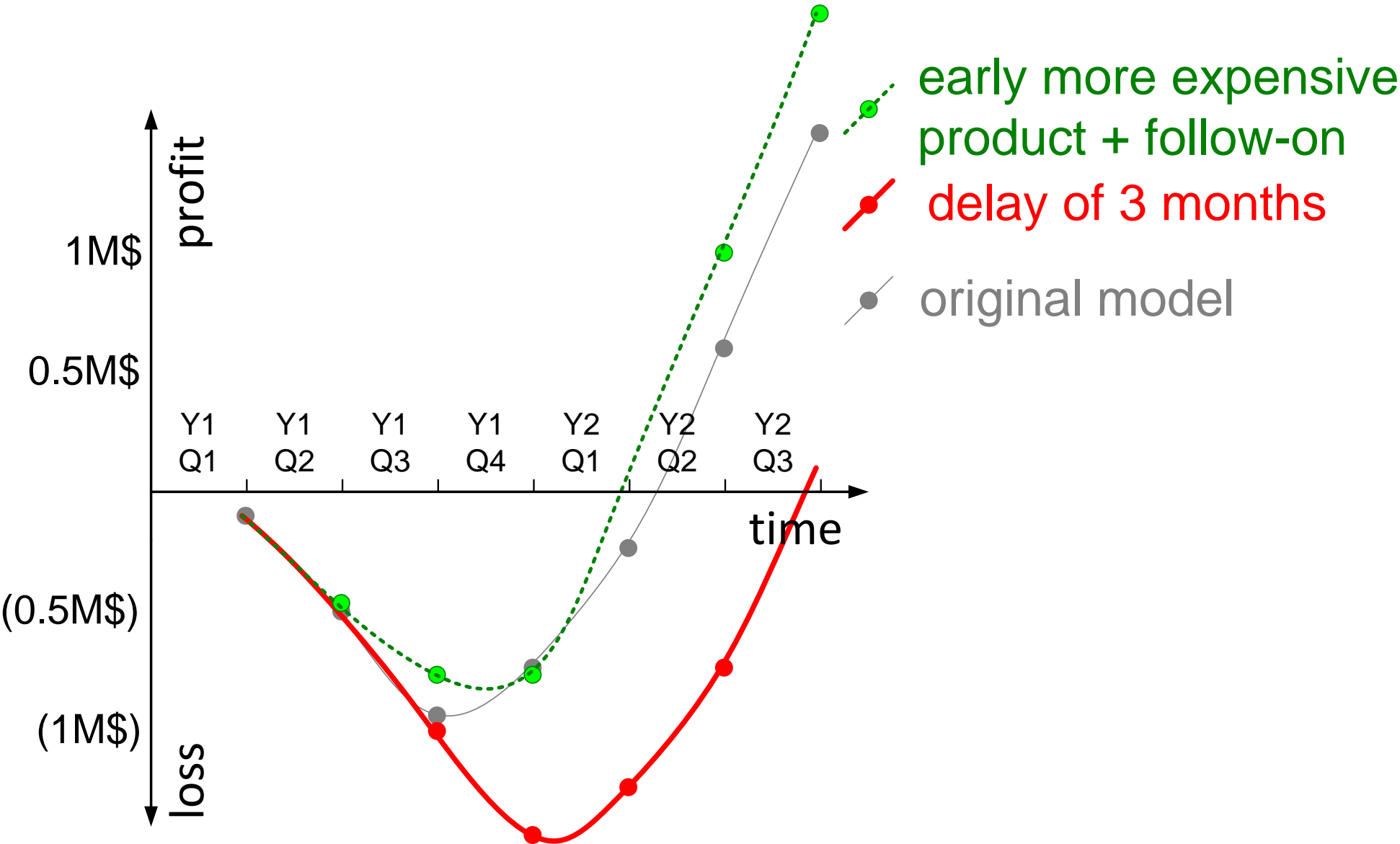
cost price / unit = 20k\$
sales price / unit = 50k\$

*variable cost = sales volume * cost price / unit*
*income = sales volume * sales price / unit*
quarter profit = income - (investments + variable costs)

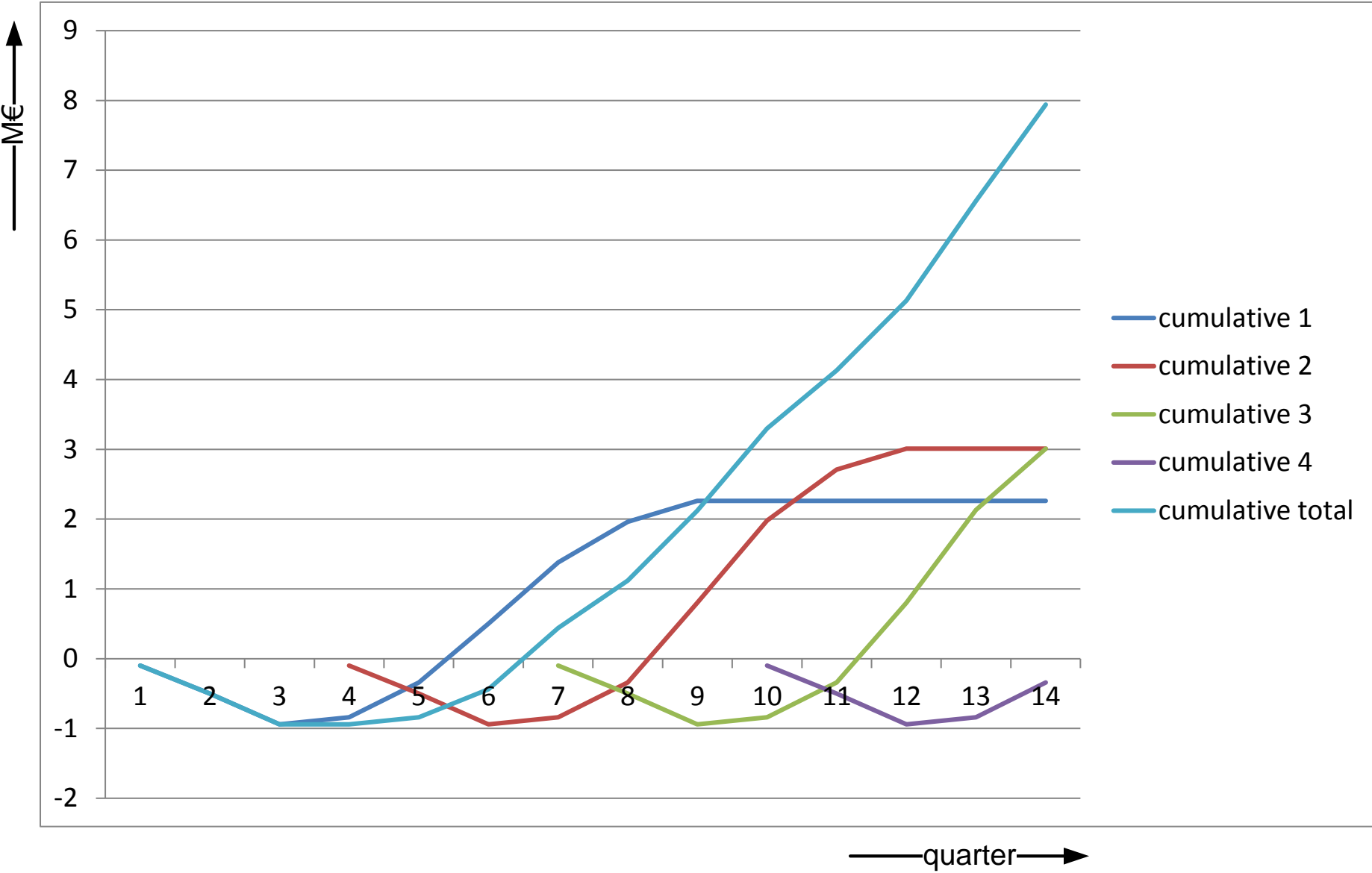
The "Hockey" Stick



What if ...?



Stacking Multiple Developments



Fashionable financial yardsticks

Return On Investments (ROI)

Net Present Value

Return On Net Assets (RONA) leasing reduces assets, improves RONA

turnover / fte outsourcing reduces headcount, improves this ratio

market ranking (share, growth) "only numbers 1, 2 and 3 will be profitable"

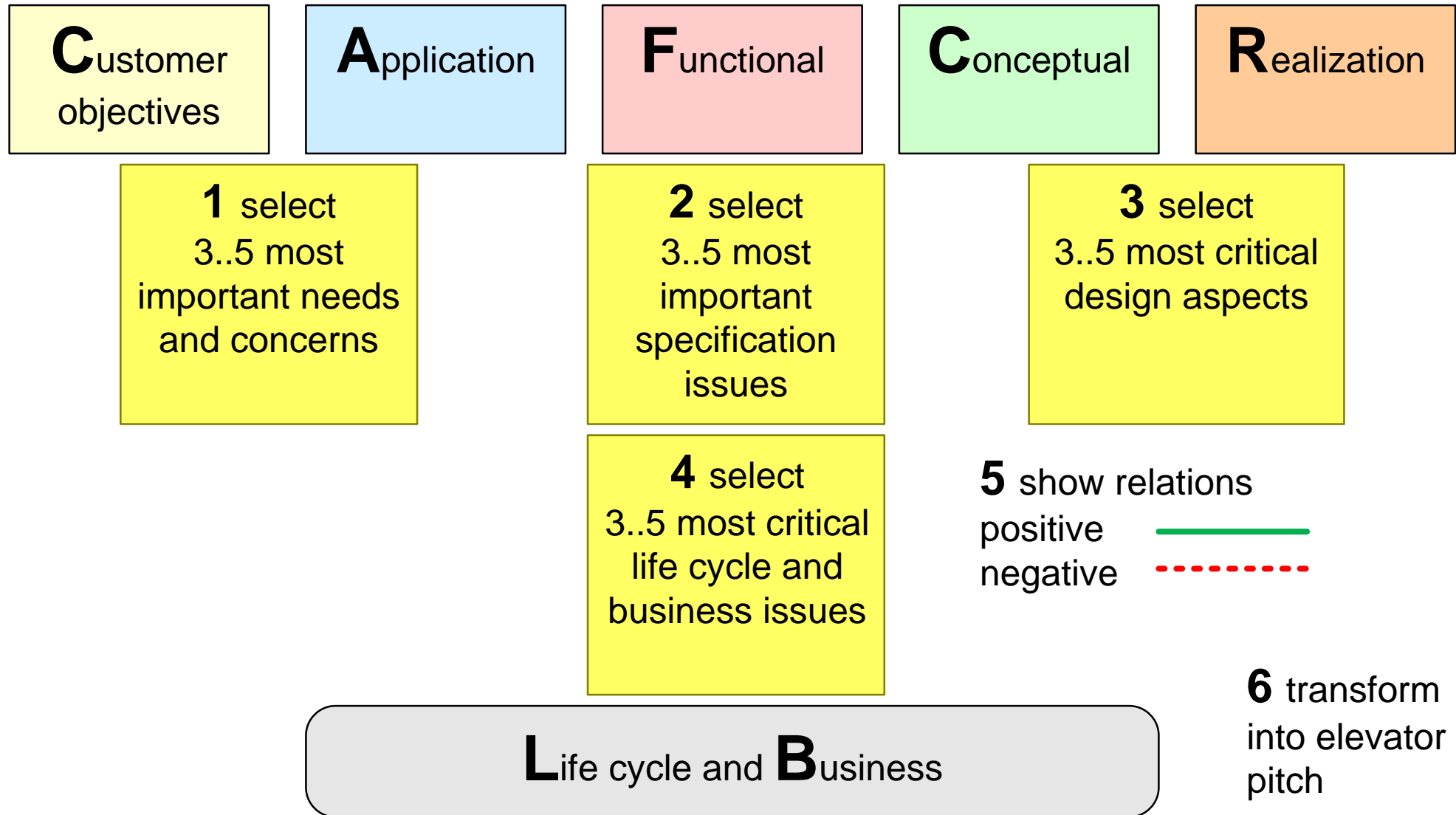
R&D investment / sales in high tech segments 10% or more

cash-flow fast growing companies combine profits with negative cash-flow,
risk of bankruptcy

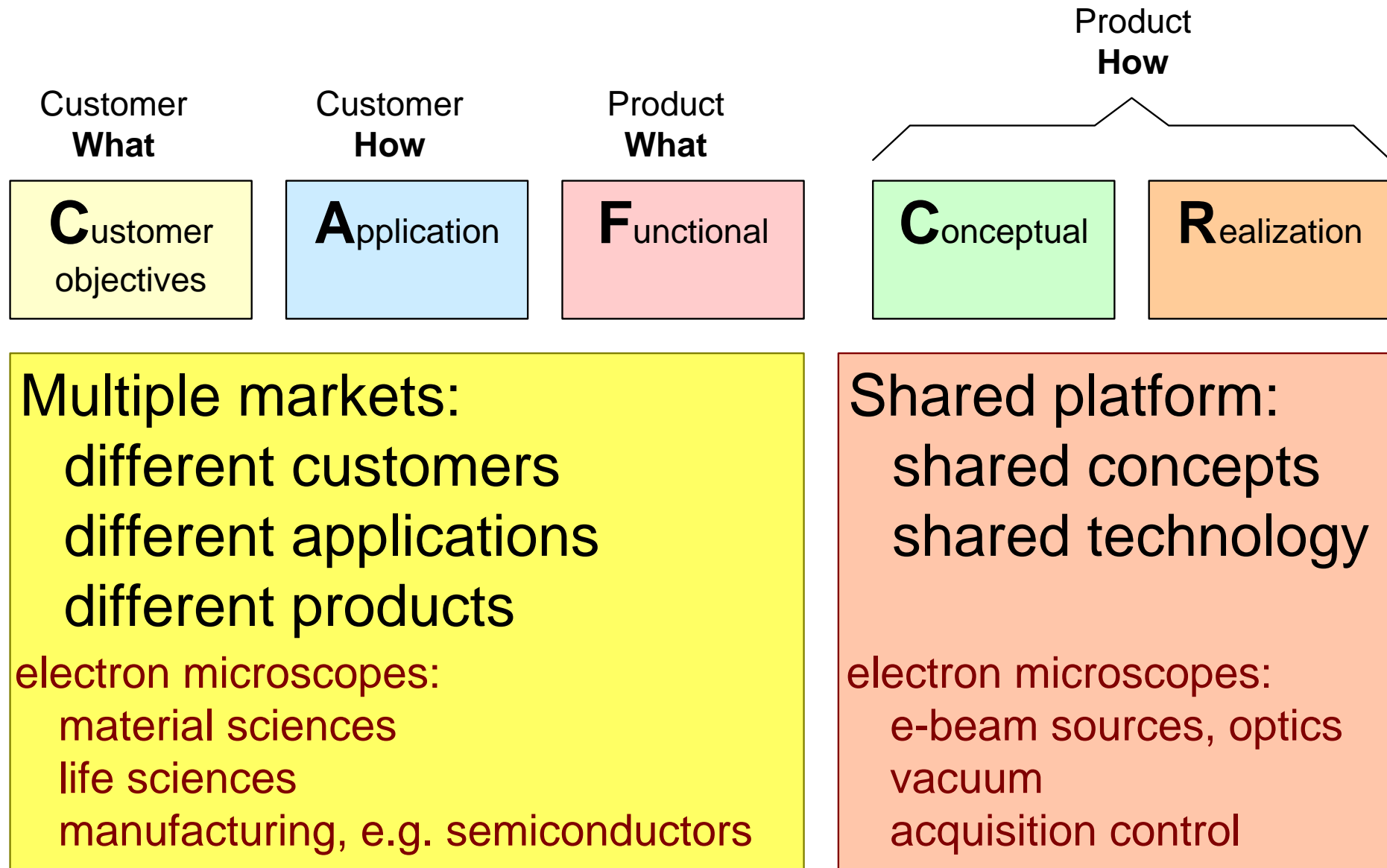
Make a **business plan** for the mid to long-term future.

- determine business model
- determine investments, sales volume, sales price, and costs
- estimate the cash flow and accumulated profit
- include at least 3 releases or generations of systems

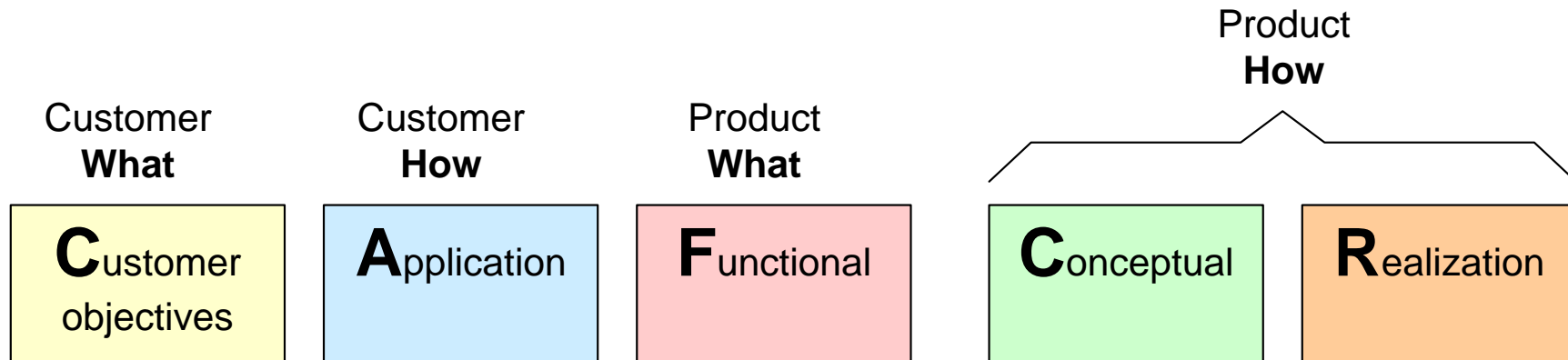
Exercise Threads of Reasoning



Multiple Markets



Complementing Systems for Same Market



Single market:
different stakeholders
different applications
interoperable products

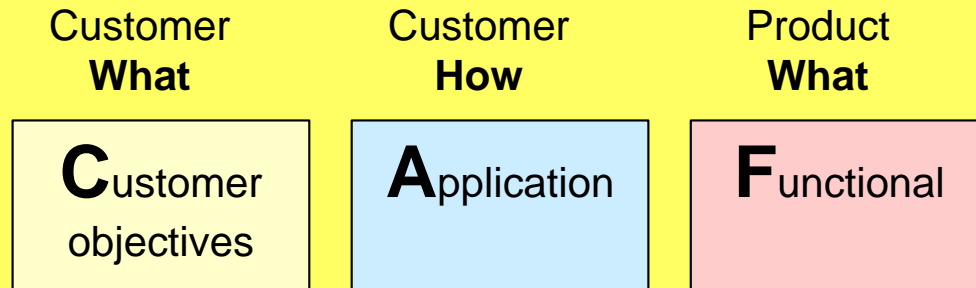
health care, e.g. cardiology:
analysis
diagnosis
treatment
administration

Shared components:
shared concepts
shared technology

health care, e.g. cardiology:
patient support
patient information
image information
storage & communication
user interface

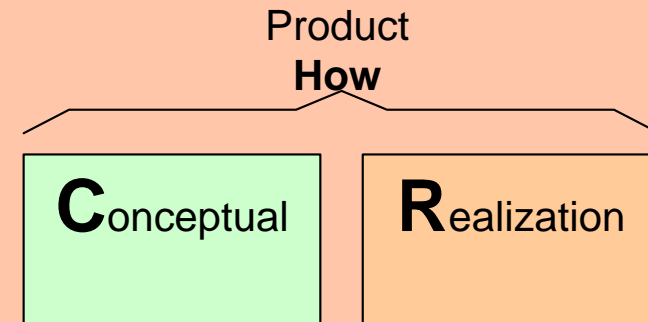
Scope Analysis

market segmentation



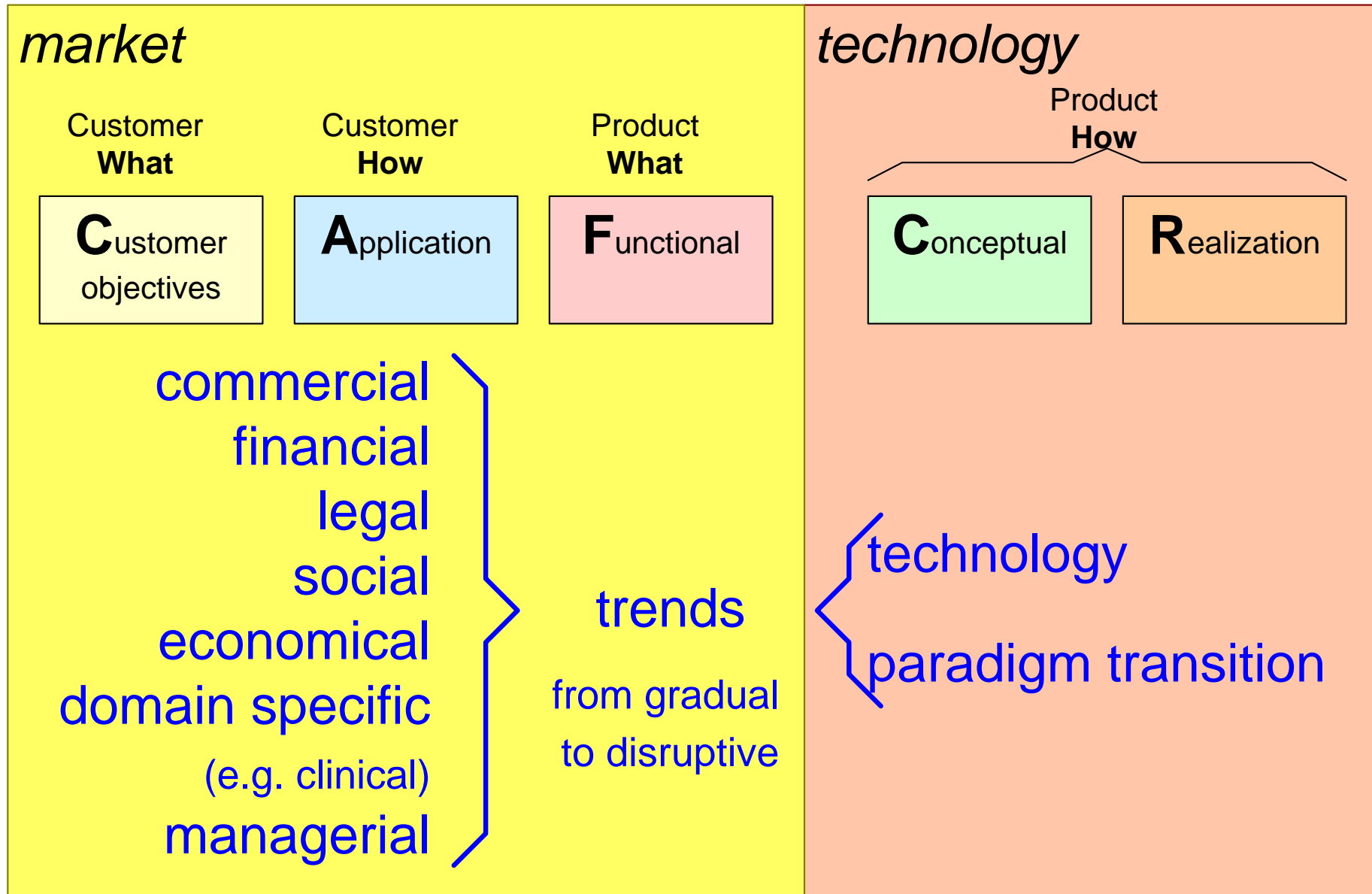
market taxonomy
customer classification
stakeholder classification
inventarization applications
inventarization
functions
features
performance

synergy analysis

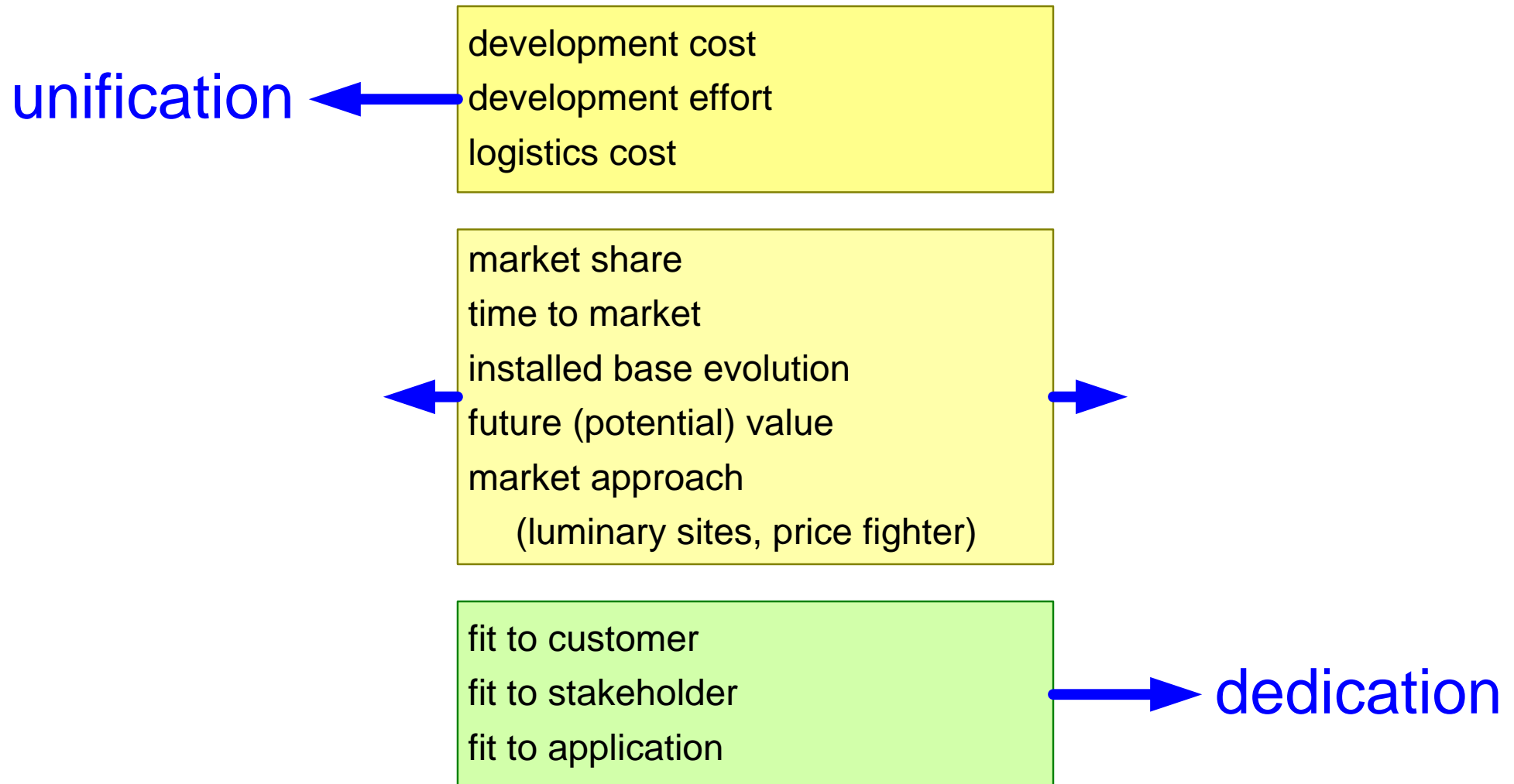


shared functionality
analyse characteristics
analyse differentiators
functionality
characteristics

Roadmapping: Impact of Future



Criteria and Forces for Synergy



Possible Levels of Sharing

intangible assets

vision, objectives

specifications, interfaces

designs, concepts

processes

tangible assets

realized components

integrated (sub)systems

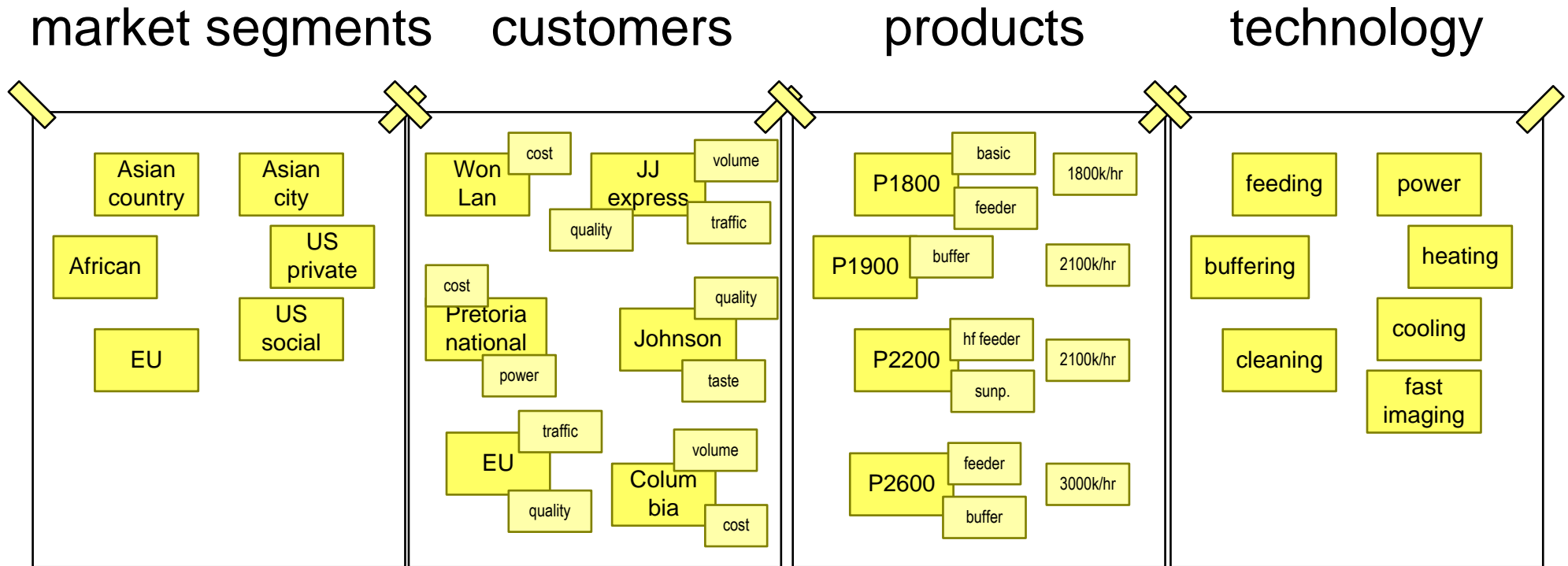
test suites

tools

infrastructure

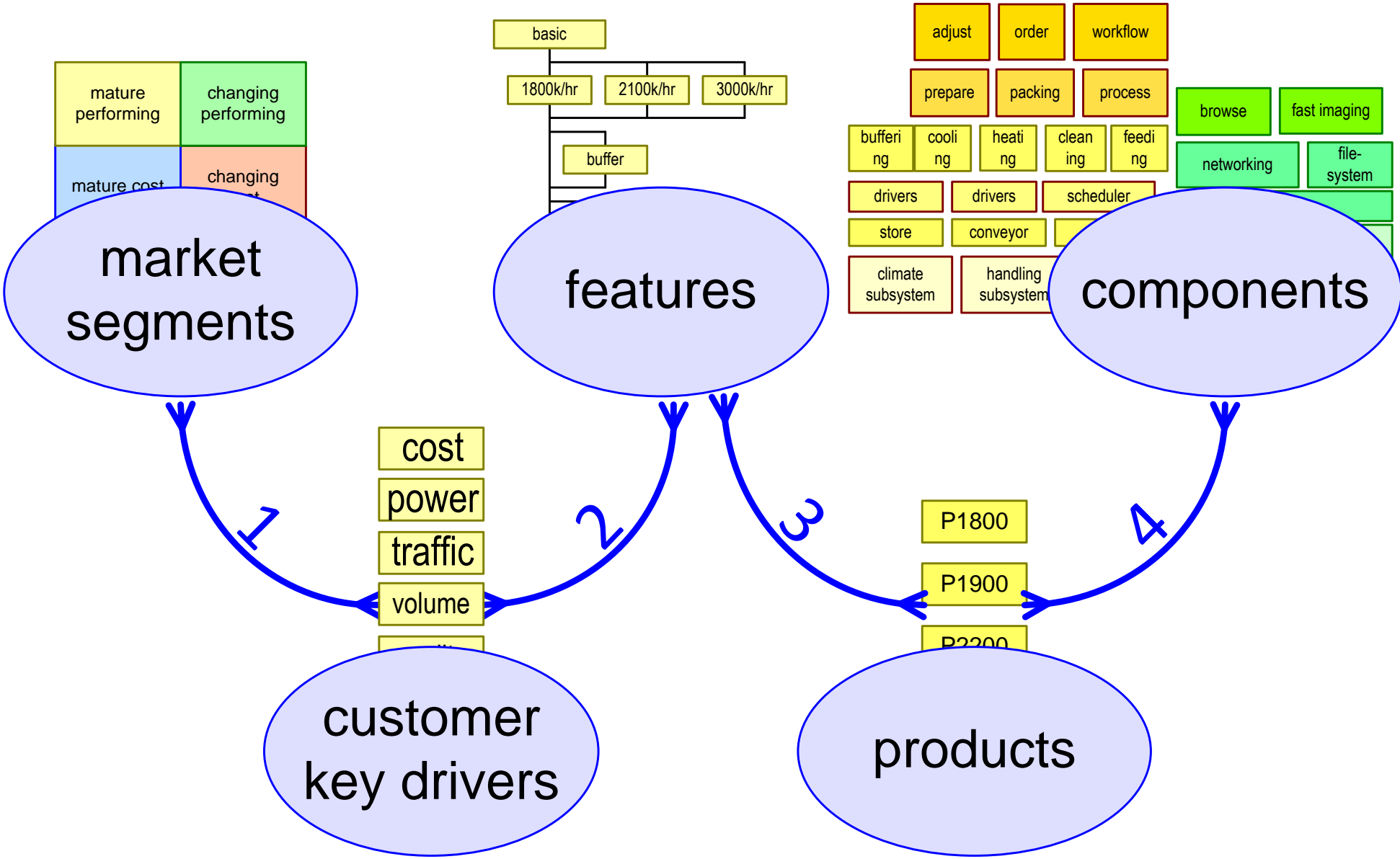
Not everything that can be shared should be shared!

Explore Markets, Customers, Products and Technologies

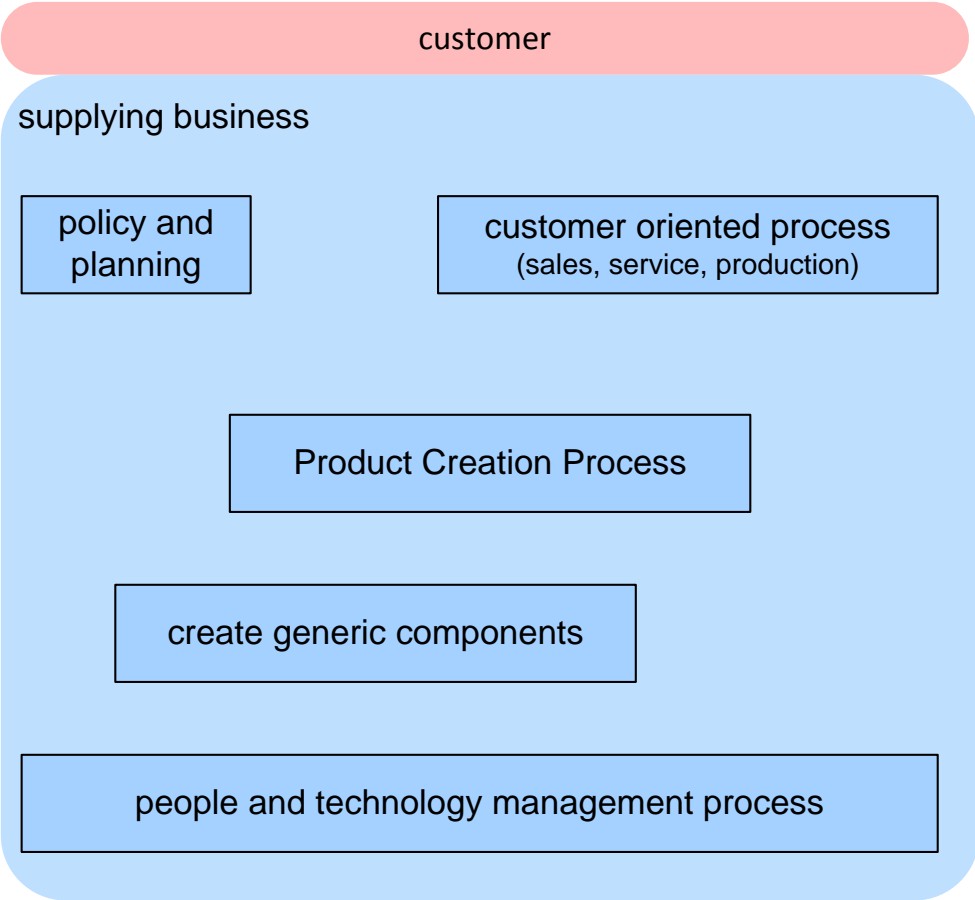


brain storm and discuss time-boxed

Mapping From Markets to Components



Models for Shared Asset Development



lead customer

direct feedback
too specific?

carrier product

product feedback
product specific?

platform

feedback problem
too generic

technology push

no feedback