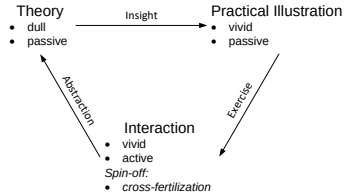
Masters Course The Context of Embedded System Design, Module 0, Information

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

Introduction to the Masters Course The Context of Embedded System Design

January 22, 2023
status: draft
version: 0

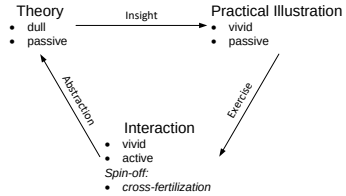


Information Masters Course The Context of Embedded System Design

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

The Masters Course The Context of Embedded System Design is a course for students following the masters “Embedded Systems”. The course material is based on the SARCH course *Systems Architecting*. However, more and shorter exercises are added, and a common case is used thorughout the course.
The course addresses a wide spectrum of issues in relation with system architecture, such as: processes, business, role and task of the system architect (team), generic Developments (re-use, platforms) requirements, roadmapping, and skills.



January 22, 2023
status: draft
version: 0.1

Program

session	subject
lecture 1	introduction, requirements capturing
lecture 2	story telling, customer views
lecture 3	product creation in business context
lecture 4	roles and tasks in product creation
lecture 5	how to: document, present
lecture 6	roadmapping
lecture 7	product families, platforms
lecture 8	presentation by teams

Case: Intelligent Greenhouse

Teams of 3 to 5 students

Describe the context of the Intelligent Greenhouse,
one subject/section per week.
Every lecture one subject will be discussed.
Send the resulting section within one week to the teacher.
Filename: Team<Teamnumber>Subject<subjectnumber>
Filesize <100 kB prevent mailbox overflow :-(

At the end: present an overview to the Management Team.
Send complete description within two weeks to the teachers:
gerrit.muller@esi.nl; joris.van.den.aker@esi.nl

Exercises Requirements

- 1 Describe a “Intelligent Greenhouse”: What does it look like, what can it do?
- 2 Identify Stakeholders and concerns
- 3 Discuss the technological opportunities and challenges
- 4 Make a key driver map

Exercises Story Telling

- 1 Create a story
- 2 Improve the story, with the criteria for stories in mind
- 3 Derive a case description from the story
- 4 Make a design to satisfy the case description

Exercises Product Creation

- 1 Identify the processes within your own company.
- 2 Make a design of the product
- 3 Make a work breakdown structure
- 4 Propose an organizational structure, quantify the size of the groups.

Exercises Roles and Tasks

- 1 Determine the most critical system functions and performance aspects
- 2 Propose an integration plan
- 3 Perform a risk assessment
- 4 Improve the organizational structure

Exercises Documentation and Presentation

- 1 Analyse the costs of the product creation, manufacturing and sales
- 2 Analyse the income
- 3 Make multi-year business forecast
- 4 Make a presentation outline for the presentation to the Board of Management

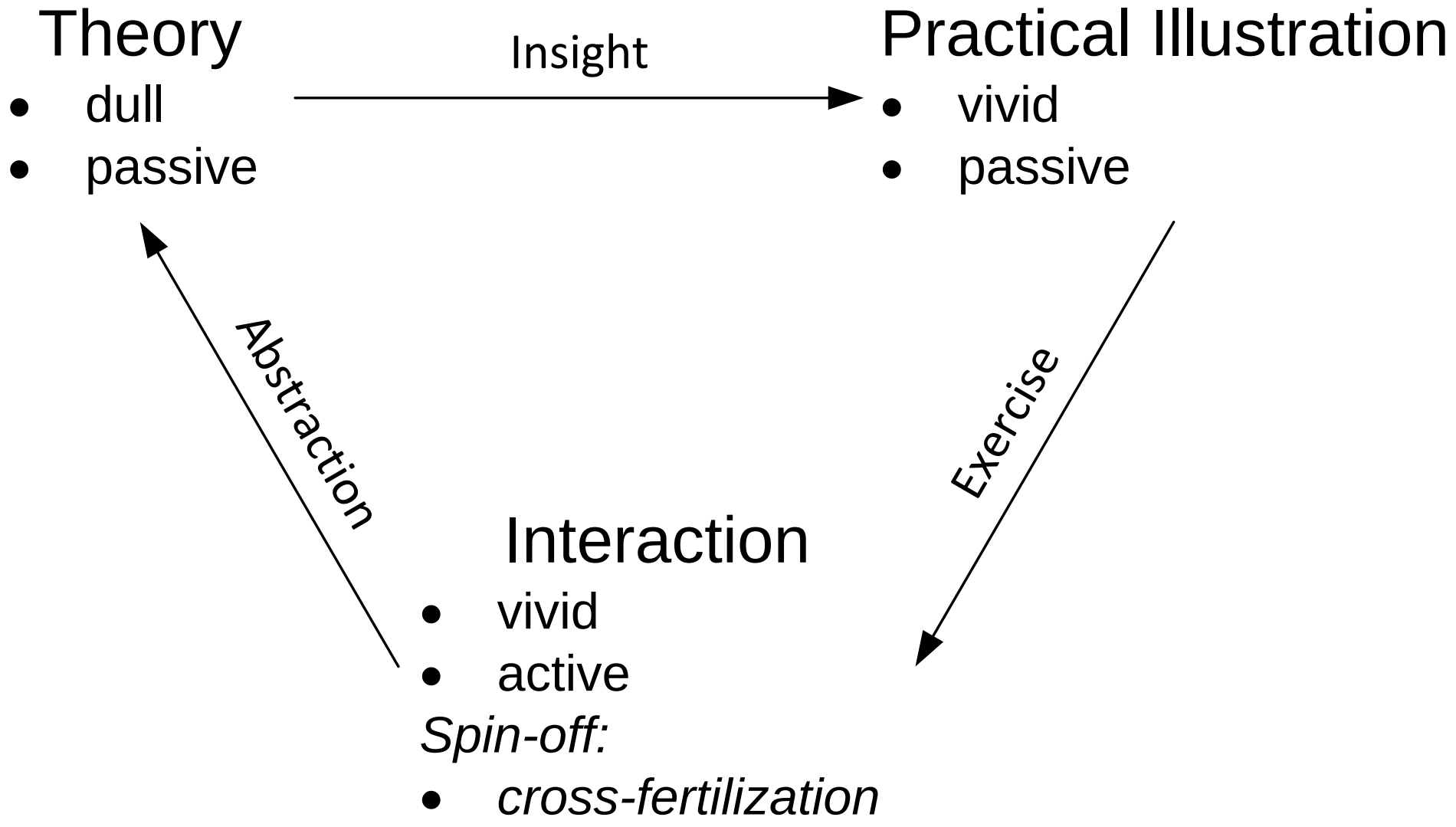
Exercises Roadmapping

- 1 Identify Market trends
- 2 Identify Technology Trends
- 3 Make a product roadmap proposal
- 4 Integrate Market, products, technology into 1 roadmap and identify Process and People issues

Exercises Product Families

- 1 Identify the members of the product family
- 2 Identify the synergy between the members of the family
- 3 Identify the member specific functionality
- 4 Propose a balanced product family approach

Structure


Theory
dull
passive
Insight
Practical Illustration
vivid
passive
Abstraction
Exercise
Interaction
vivid
active
Spin-off:
cross-fertilization

Rules of the Interactive Parts

- Your contribution is essential.
- Don't monopolize the time, everyone also the quiet people should have the opportunity to contribute;
  The facilitator will intervene if the contribution is limited to a small group of participants.
- Respect the contribution of others;
  Opinions can't be wrong, difference of opinion is normal and called pluriformity.
- The course format is highly experimental and based on improvisation, constructive proposals are welcome;
  it is your course! Regular evaluations will give the opportunity to influence the rest of the course.

Rules of the Broadcast Parts

- Please write your questions/remarks/statements on yellow stickers and attach them at the end on the P-flip.
  These will be used in the interactive section for discussion and to increase insight.
- Short clarification questions are welcome,
  discussion will take place in the interactive part.
- Stupid questions don't exist. Learning is based on **safe** and **open** interaction.
  Very individual oriented questions can be referred to a break or after the session.

Module Requirements

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses requirements: What are requirements? How to find, select, and consolidate requirements?

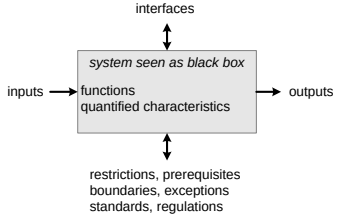
January 22, 2023
status: concept
version: 1.4

Fundamentals of Requirements Engineering

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

Requirements engineering is one of the systems engineering pillars. In this document we discuss the fundamentals of systems engineering, such as the transformation of needs into specification, the need to prescribe *what* rather than *how*, and the requirements when writing requirements.



January 22, 2023
status: concept
version: 0.1

Definition of “Requirement”

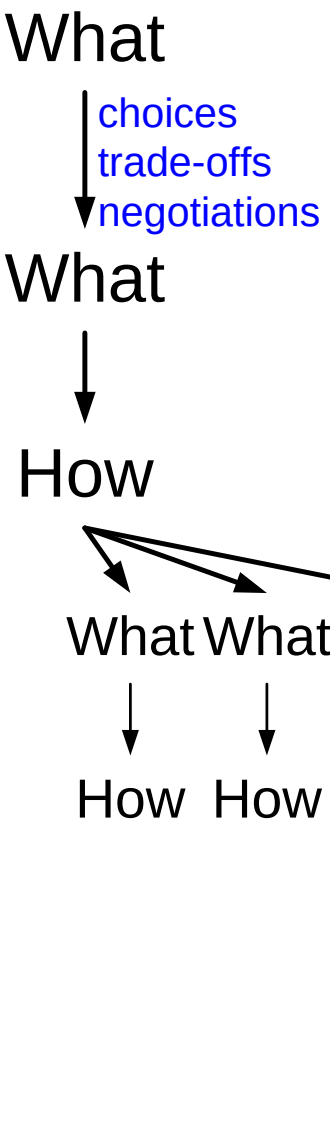
Requirements describing the needs of the customer: ***Customer Needs***

Requirements describing the characteristics of the final resulting system (product): ***System (Product) Specification***

The ***requirements management process*** recursively applies this definition for every level of decomposition.

Requirements describing the needs of the company itself over the life cycle: ***Life Cycle Needs***

Flow of Requirements



customer needs:
What is needed by the customer?

product specification:
What are we going to realize?

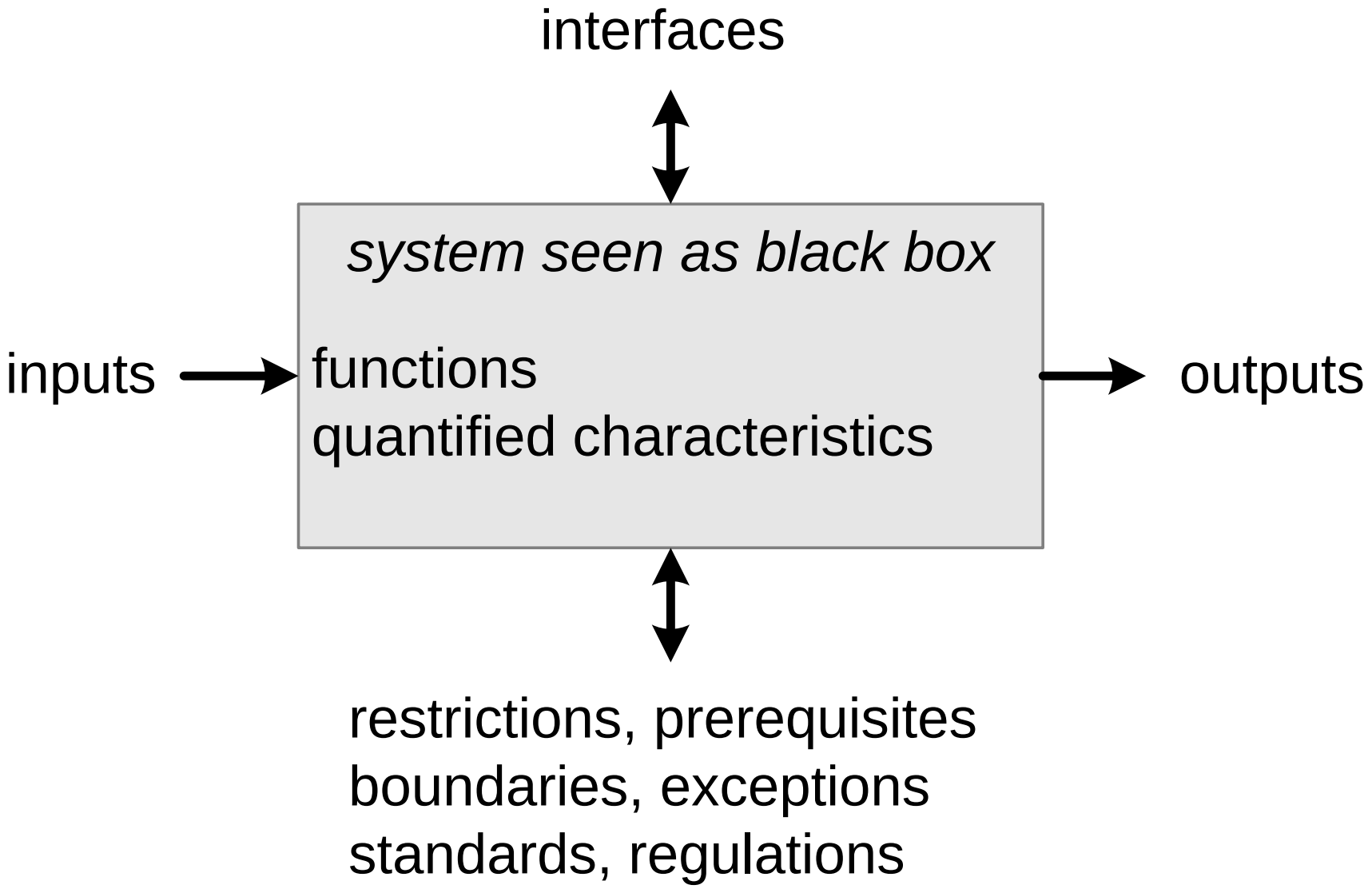
system design:
How are we going to realize the product?

What are the subsystems we will realize?

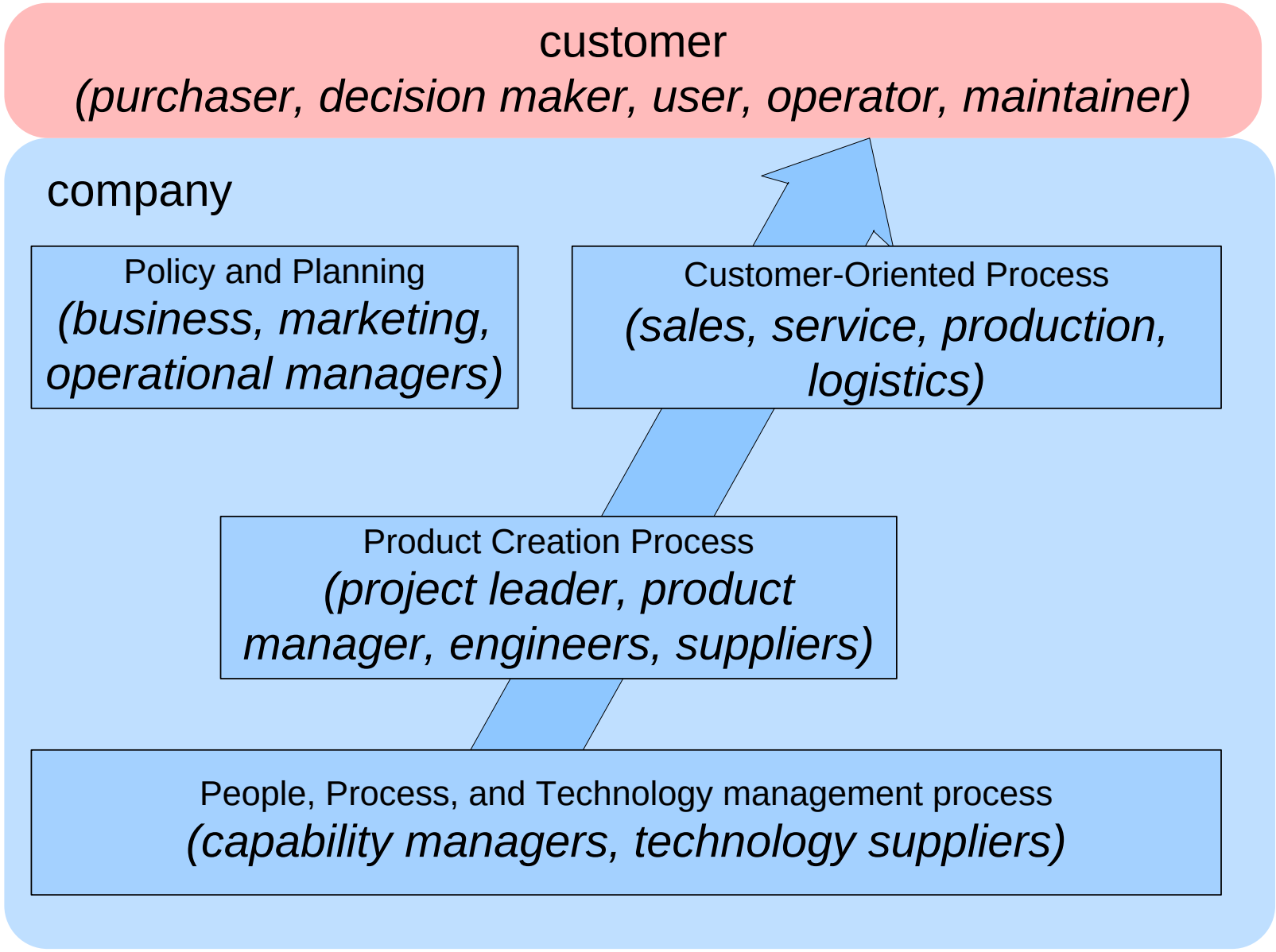
How will the subsystems be realized?

up to "atomic" components

System as a Black Box



Stakeholders w.r.t. Requirements



The “Formal” Requirements for Requirements

Specific

Unambiguous

Verifiable

Quantifiable

Measurable

Complete

Traceable

The Requirements to Enable Human Use

Accessible

Understandable

Low threshold

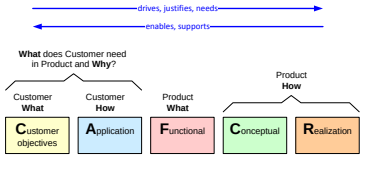
Short introduction to basic “CAFCR” model

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

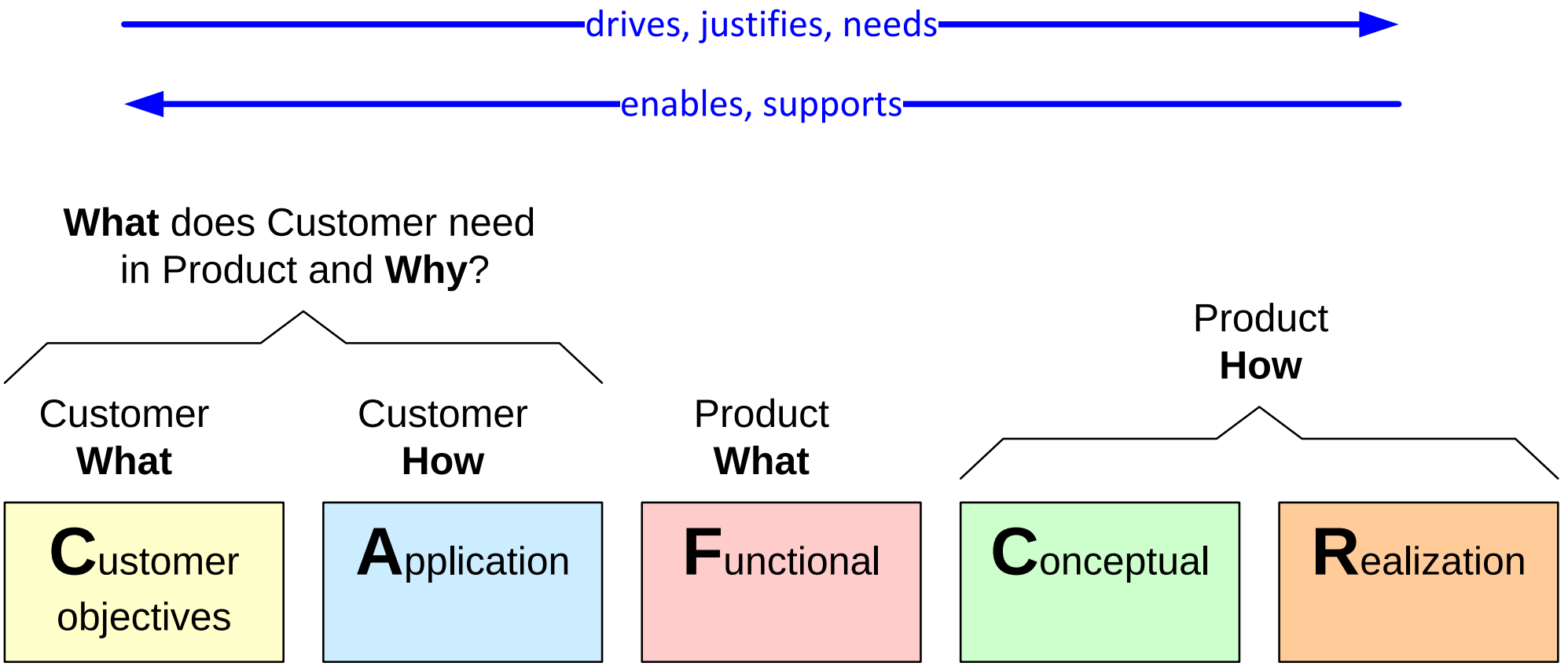
Abstract

The basic “CAFCR” reference model is described, which is used to describe a system in relation to its context. The main stakeholder in the context is the customer. The question “Who is the customer?” is addressed.

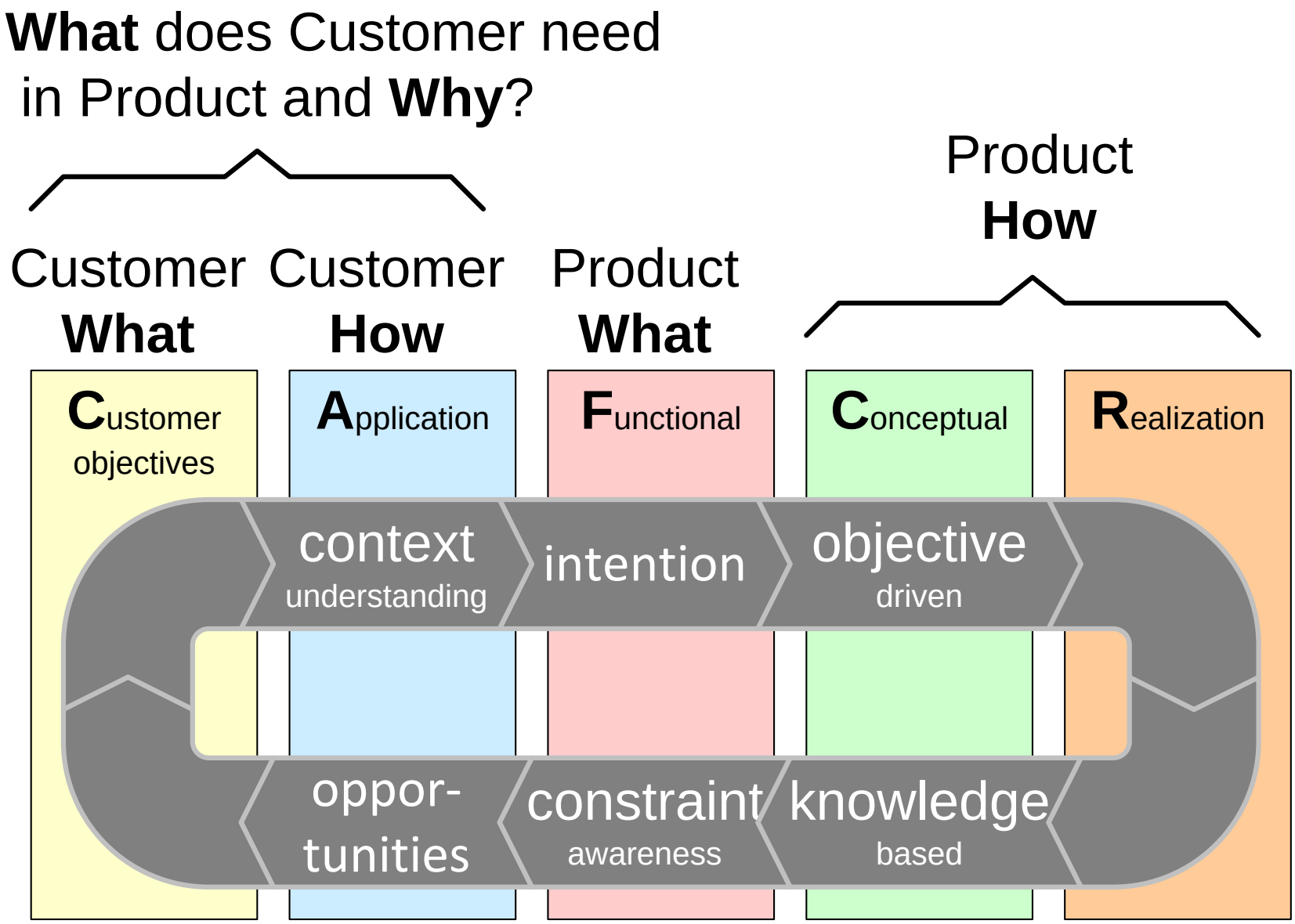
January 22, 2023
status: draft
version: 0.4



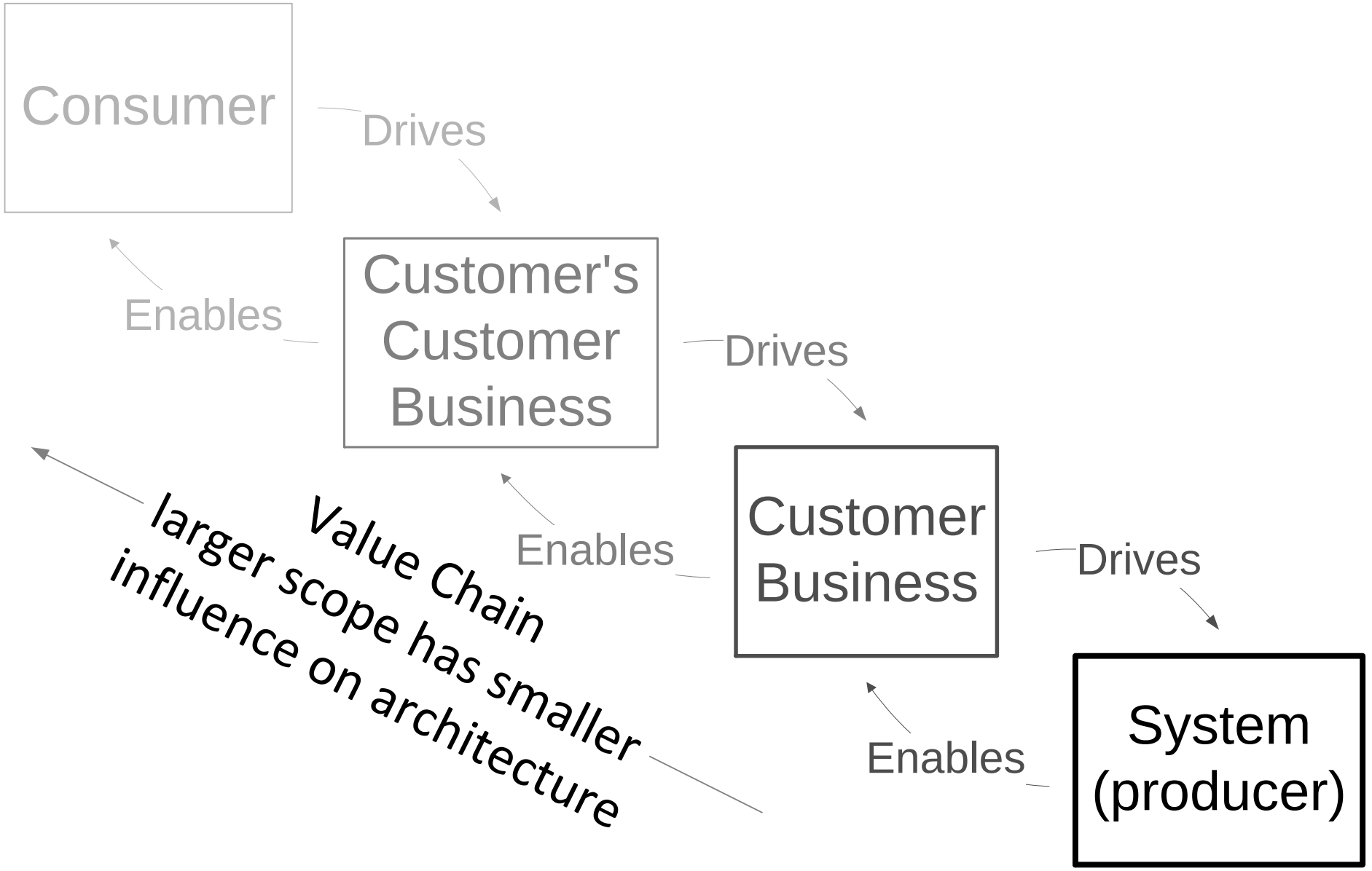
The “CAFCR” model



Integrating CAFCR



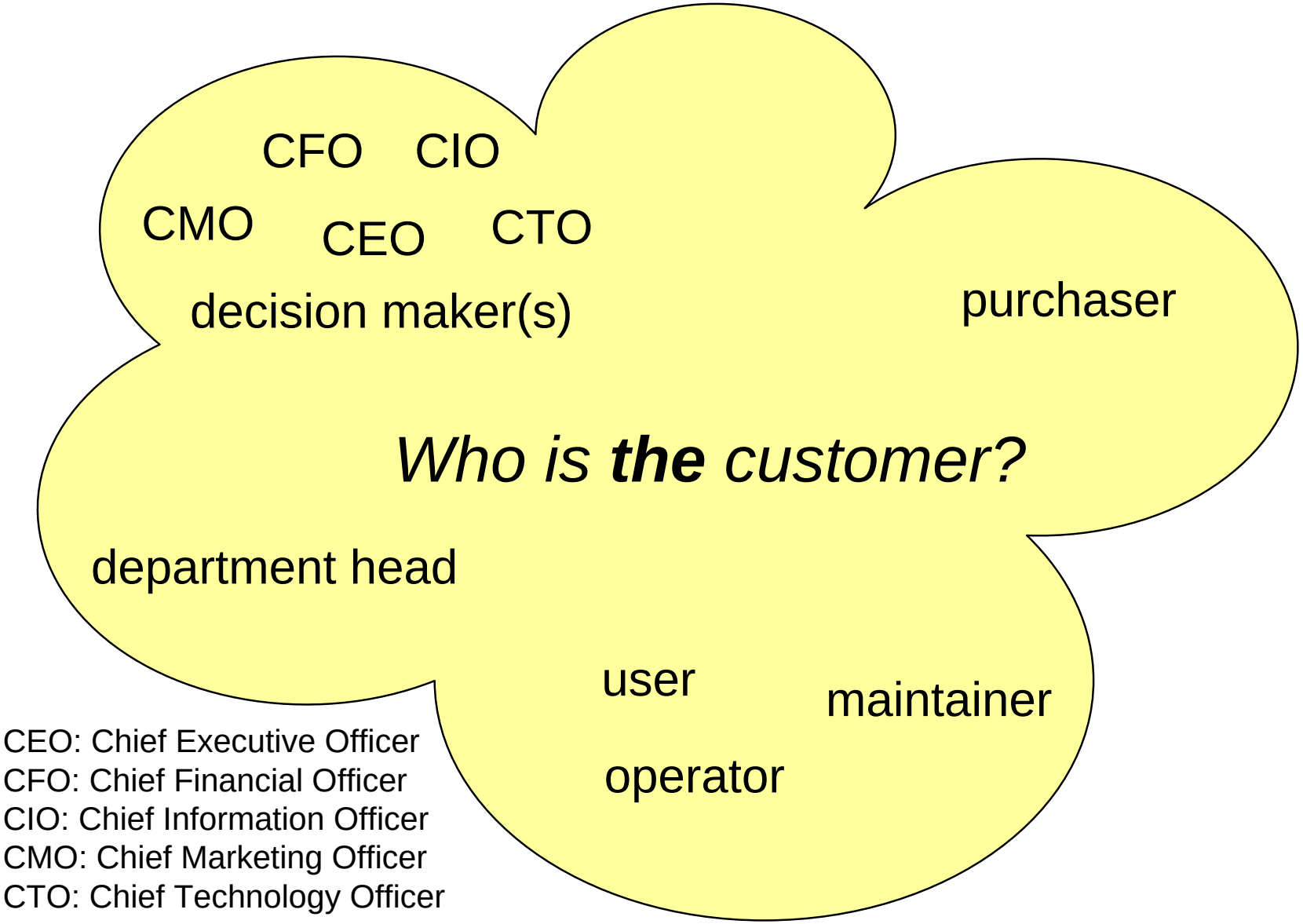
CAFCR can be applied recursively



Market segmentation

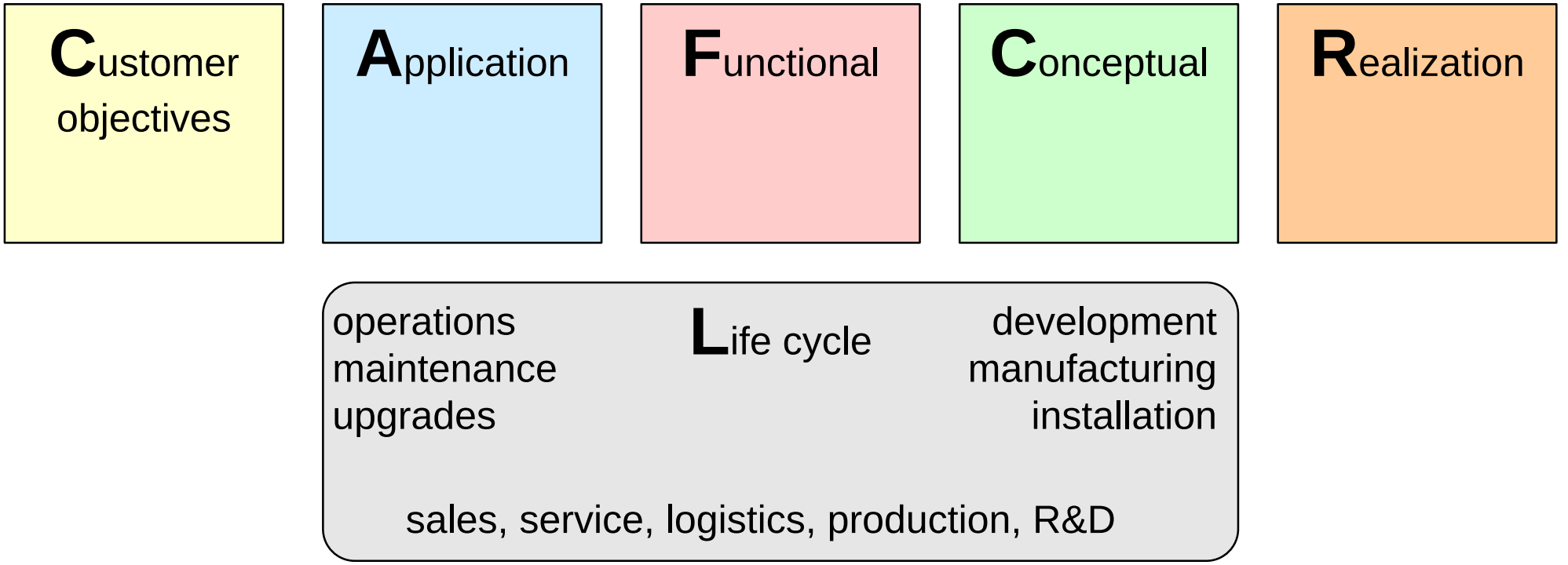
segmentation axis	examples
geographical	USA, UK, Germany, Japan, China
business model	profit, non profit
economics	high end versus cost constrained
consumers	youth, elderly
outlet	retailer, provider, OEM, consumer direct

Example of a small buying organization



CEO: Chief Executive Officer
CFO: Chief Financial Officer
CIO: Chief Information Officer
CMO: Chief Marketing Officer
CTO: Chief Technology Officer

CAFCR+ model; Life Cycle View

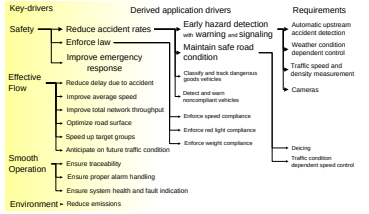


Key Drivers How To

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

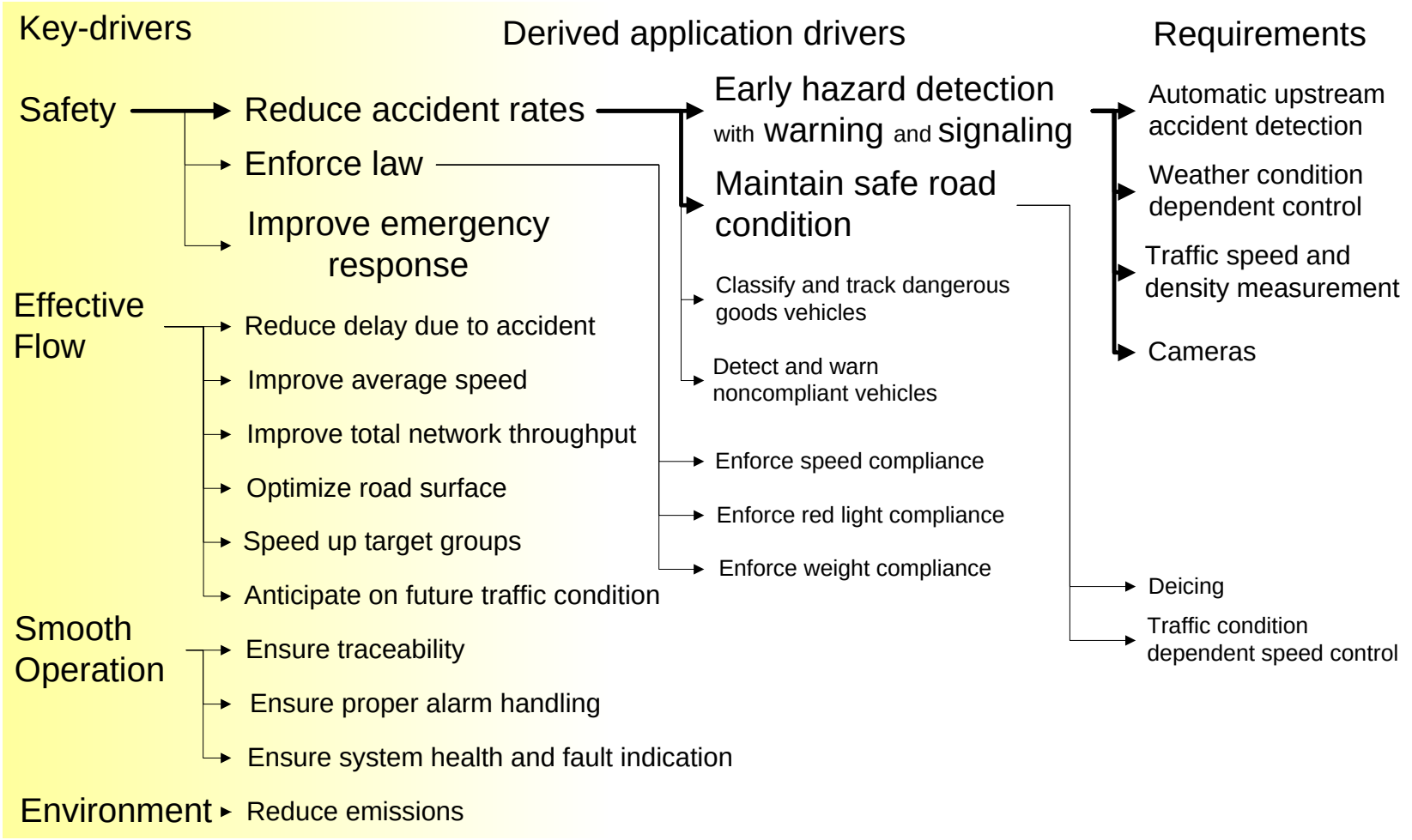
The notion of ”business key drivers” is introduced and a method is described to link these key drivers to the product specification.



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

January 22, 2023
status: draft
version: 0.2

Example Motorway Management Analysis



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

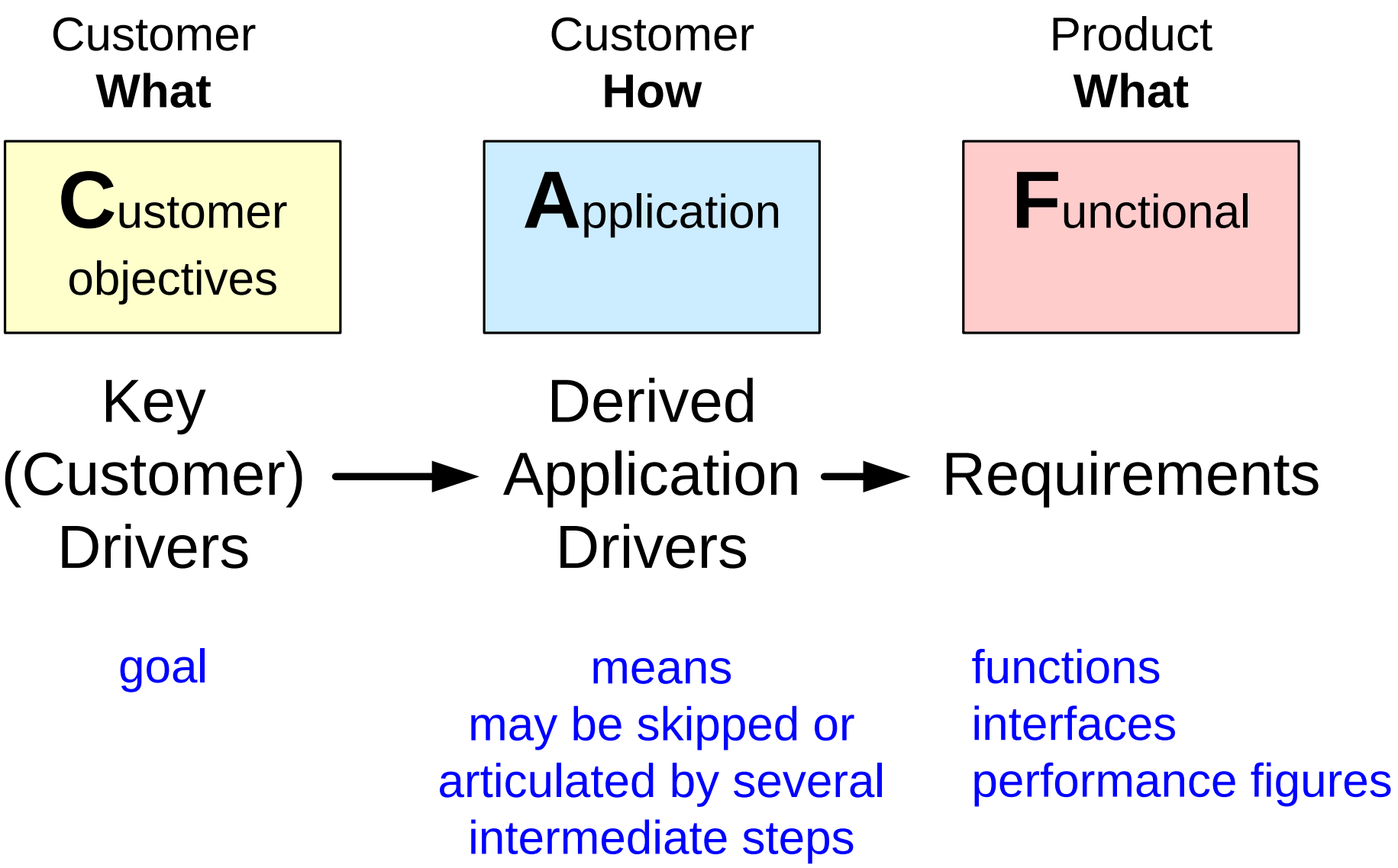
Method to create Key Driver Graph

• 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

Transformation of Key Drivers into Requirements



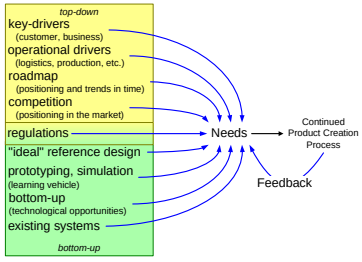
Requirements Elicitation and Selection

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

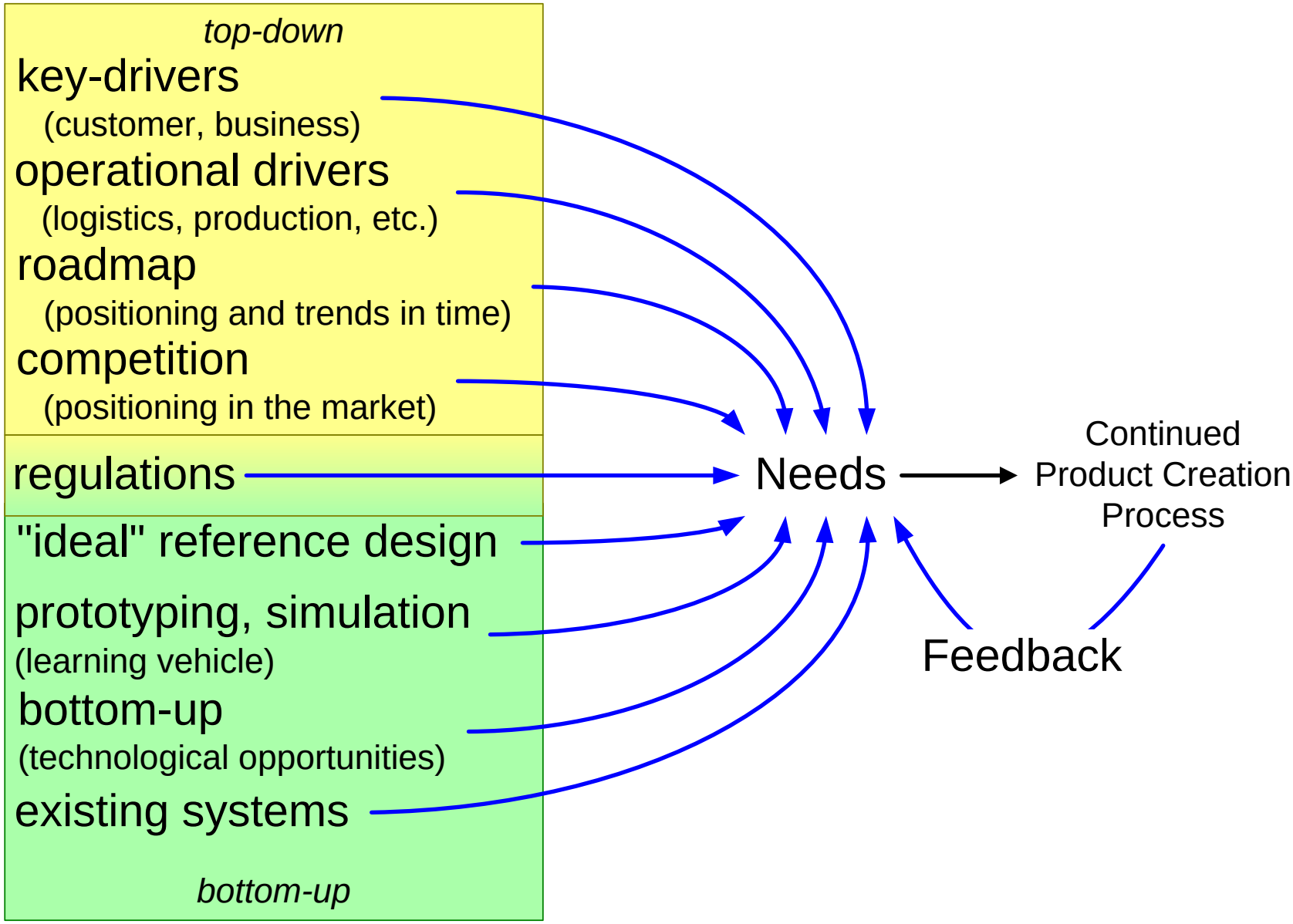
Abstract

An elicitation method for needs is described using many different viewpoints. A selection process with a coarse and a fine selection is described to reduce the specification to an acceptable and feasible subset.

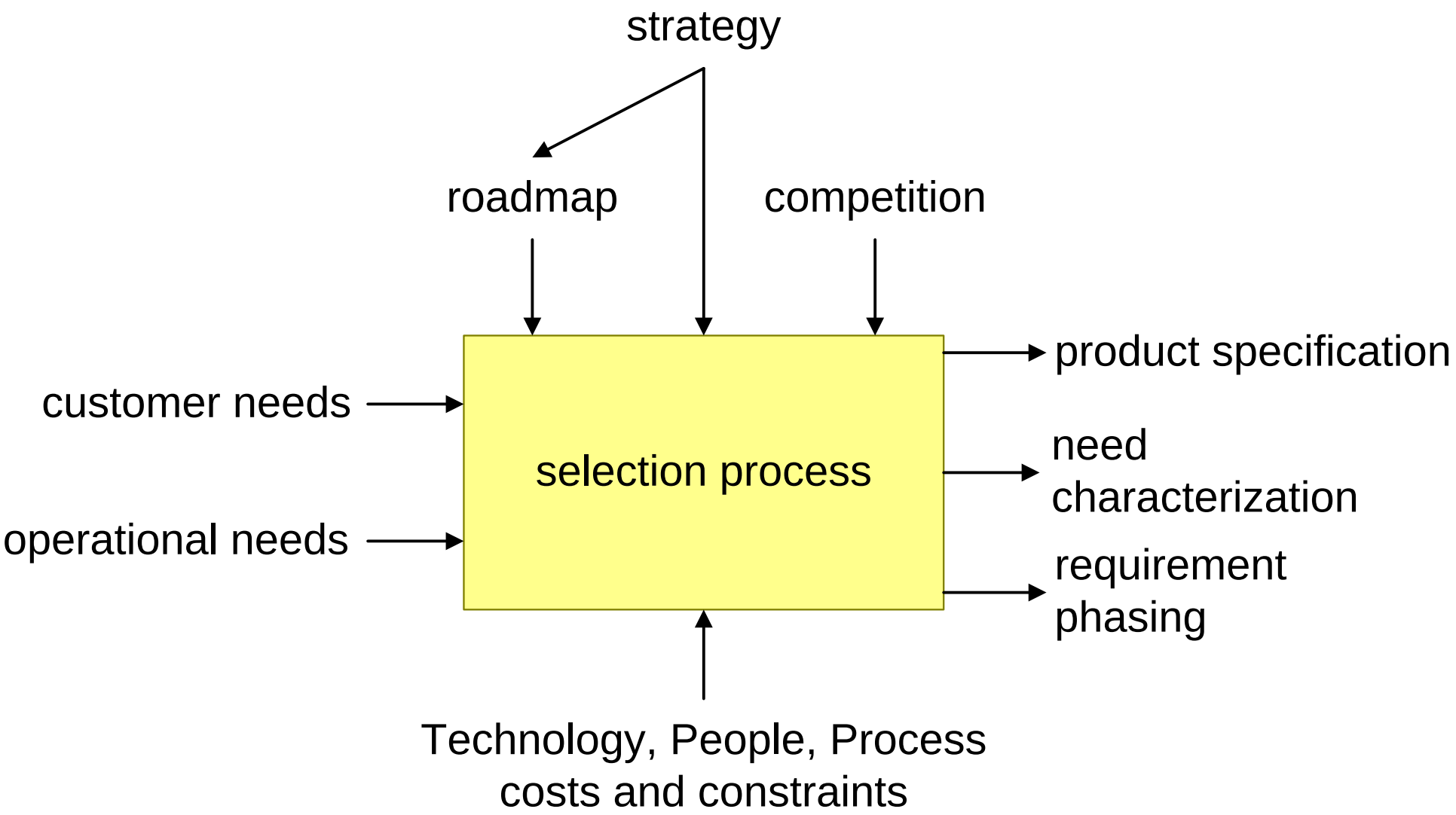
January 22, 2023
status: draft
version: 0



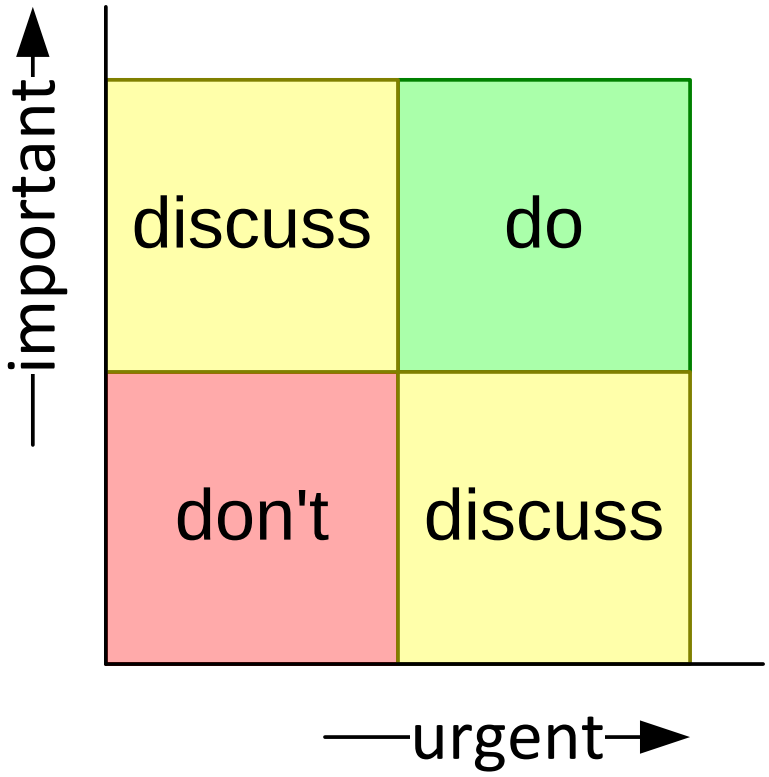
Complementary Viewpoints to Capture Requirements

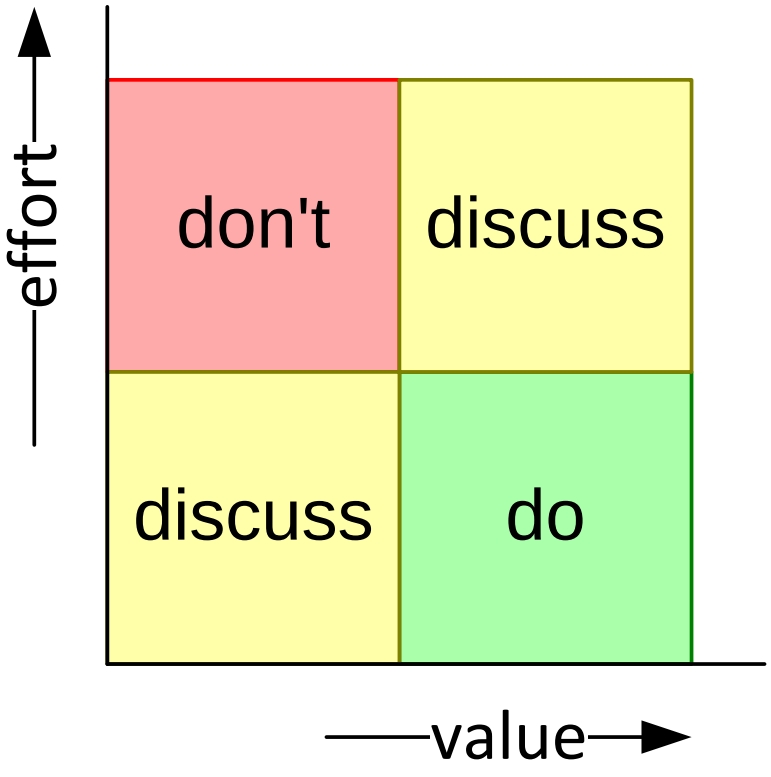


Requirement Selection Process



Simple Qualification Method





Examples of Quantifiable Aspects

- Value for the customer
- (dis)satisfaction level for the customer
- Selling value (How much is the customer willing to pay?)
- Level of differentiation w.r.t. the competition
- Impact on the market share
- Impact on the profit margin

Use relative scale, e.g. 1..5 1=low value, 5 -high value

Ask several knowledgeable people to score

Discussion provides insight (don't fall in spreadsheet trap)

Exercise Requirements Capturing

- Determine the key drivers for one particular product family.
- Translate these drivers into application drivers and derive from them the requirements.

Needs and Requirements

Needs, Specification, Requirements

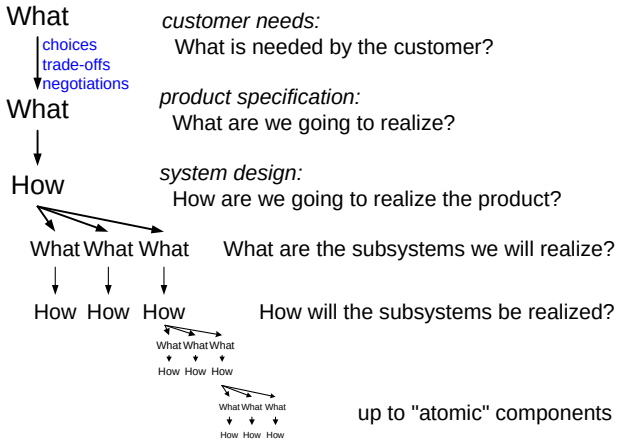
Requirements describing the needs of the customer: ***Customer Needs***

Requirements describing the characteristics of the final resulting system (product): ***System (Product) Specification***

The ***requirements management process*** recursively applies this definition for every level of decomposition.

Requirements describing the needs of the company itself over the life cycle: ***Life Cycle Needs***

Flow of Requirements



Requirements for Requirements

Specific

Unambiguous

Verifiable

Quantifiable

Measurable

Complete

Traceable

Enable Human Use

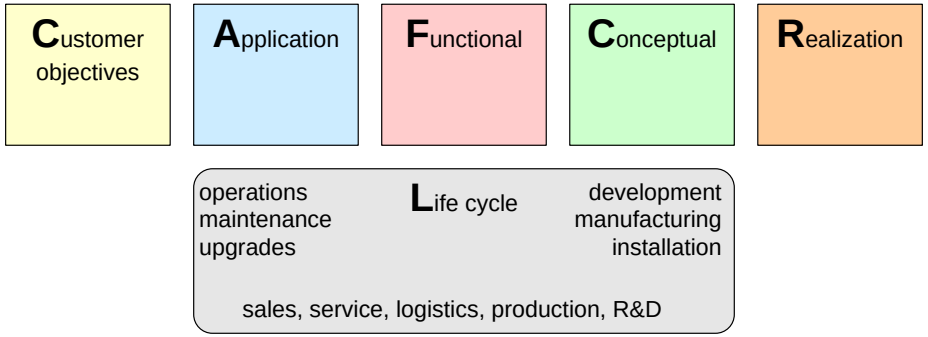
Accessible

Understandable

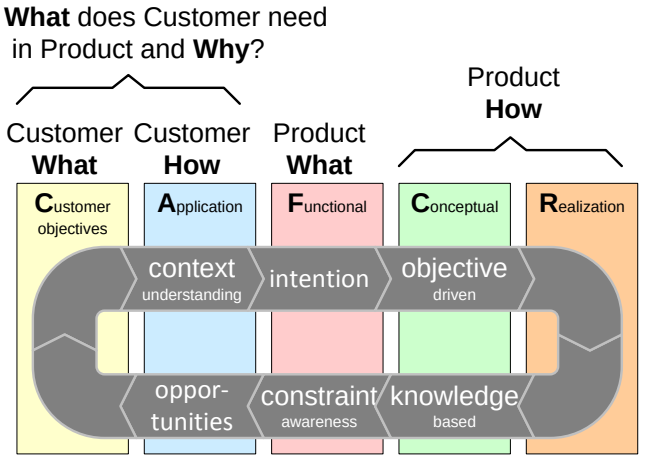
Low threshold

CAFCR, Customer Key Driver Graph

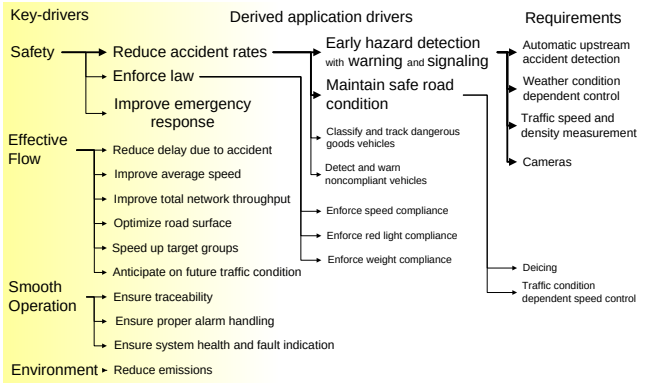
CAFCR+ Model



Iterate over Views

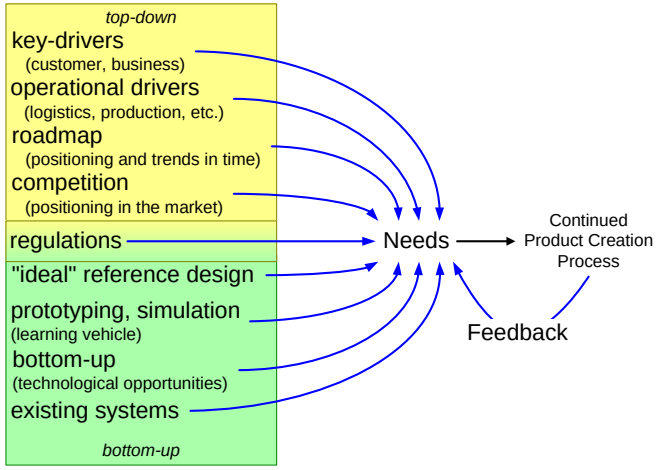


Example Key Driver Graph



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

Complementary Viewpoints



Module Story Telling

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses Story Telling as a means to explore customer needs and as a means for communication.

January 22, 2023
status: draft
version: 0

logo
TBD

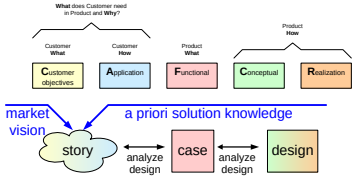
Story How To

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

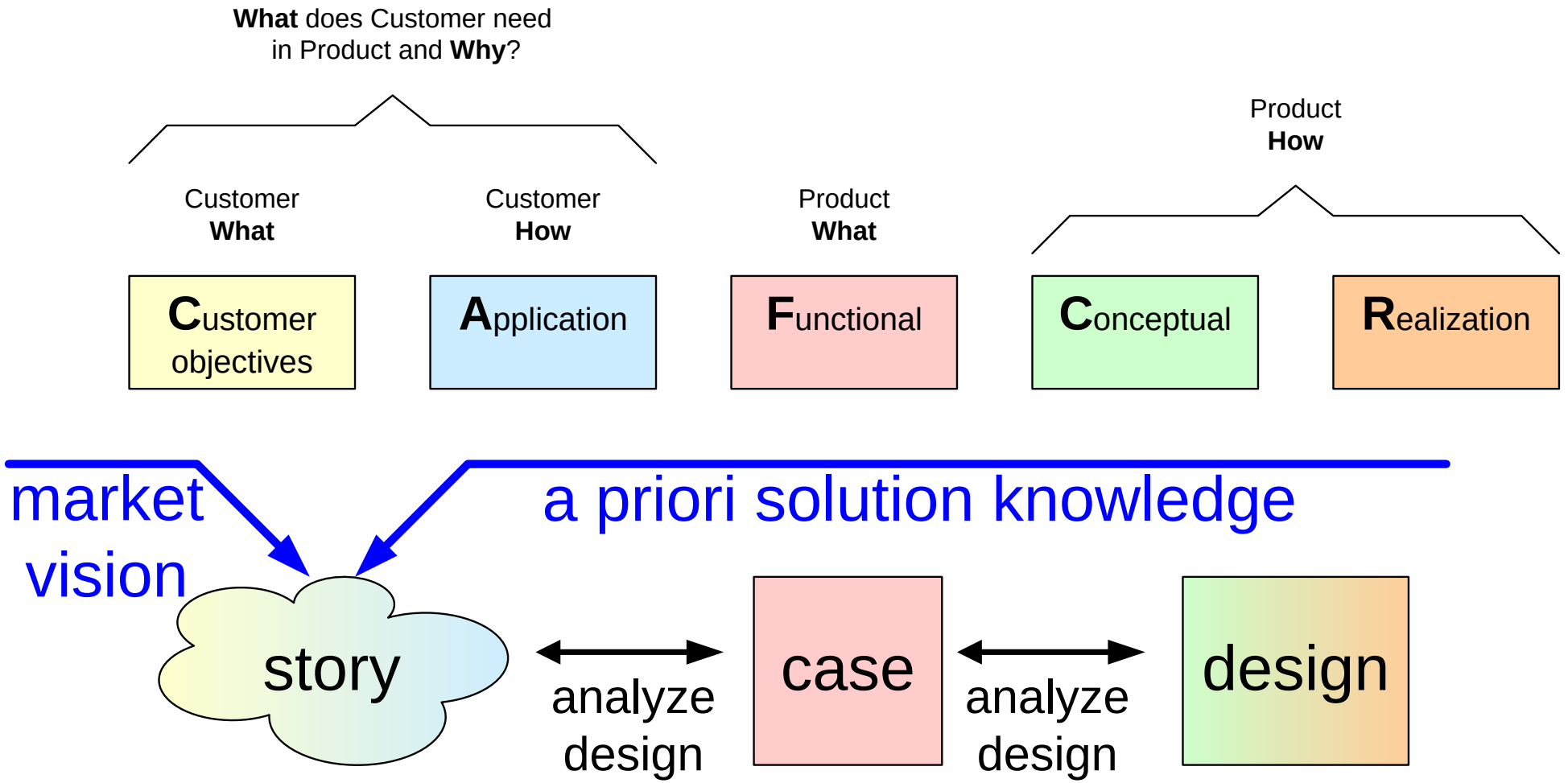
Abstract

A story is an easily accessible story or narrative to make an application live. A good story is highly specific and articulated entirely in the problem domain: the native world of the users. An important function of a story is to enable specific (*quantified, relevant, explicit*) discussions.

January 22, 2023
status: concept
version: 1.2



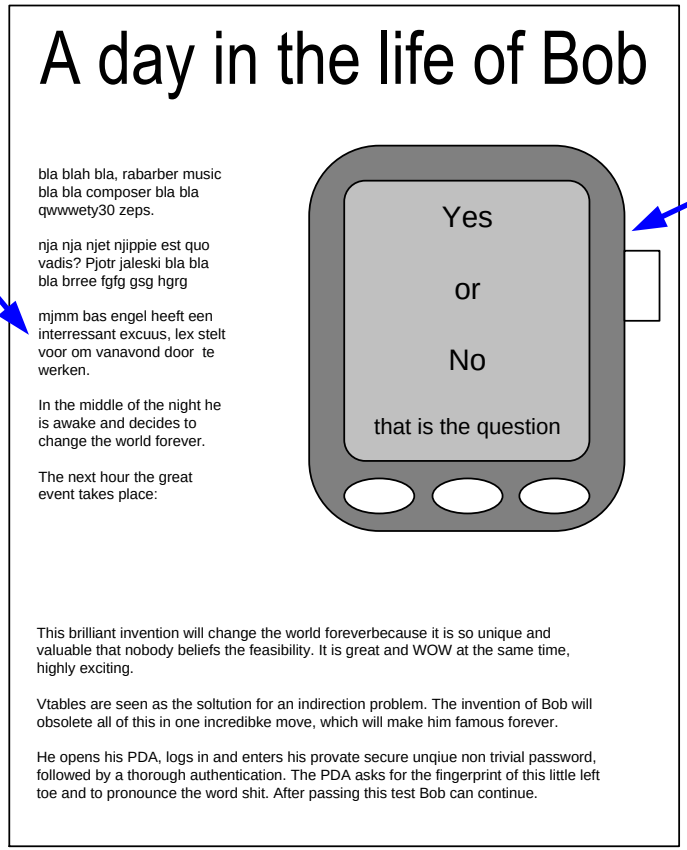
From story to design



Example story layout

ca. half a page of plain English text

draft or sketch of some essential appliance



Points of attention

- purpose — What do you need to know for specification and design?
- scope — “umbrella” or specific event?
- viewpoint, stakeholders — Define your stakeholder and viewpoint f.i. user, maintainer, installer
- visualization — Sketches or cartoon Helps to share and communicate ideas
- size (max 1 A4) — Can be read or told in few minutes
- recursive decomposition, refinement

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, ...

Example of a story

Betty is a 70-year-old woman who lives in Eindhoven. Three years ago her husband passed away and since then she lives in a home for the elderly. Her 2 children, Angela and Robert, come and visit her every weekend, often with Betty's grandchildren Ashley and Christopher. As so many women of her age, Betty is reluctant to touch anything that has a technical appearance. She knows how to operate her television, but a VCR or even a DVD player is way to complex.

When Betty turned 60, she stopped working in a sewing studio. Her work in this noisy environment made her hard-of-hearing with a hearing-loss of 70dB around 2kHz. The rest of the frequency spectrum shows a loss of about 45dB. This is why she had problems understanding her grandchildren and why her children urged her to apply for hearing aids two years ago. Her technophobia (and her first hints or arthritis) inhibit her to change her hearing aids' batteries. Fortunately her children can do this every weekend.

This Wednesday Betty visits the weekly Bingo afternoon in the meetingplace of the old-folk's home. It's summer now and the tables are outside. With all those people there it's a lot of chatter and babble. Two years ago Betty would never go to the bingo: "I cannot hear a thing when everyone babbles and clatters with the coffee cups. How can I hear the winning numbers?!". Now that she has her new digital hearing instruments, even in the bingo cacophony, she can understand everyone she looks at. Her social life has improved a lot and she even won the bingo a few times.

That same night, together with her friend Janet, she attends Mozart's opera The Magic Flute. Two years earlier this would have been one big low rumbly mess, but now she even hears the sparkling high piccolos. Her other friend Carol never joins their visits to the theaters. Carol also has hearing aids, however hers only "work well" in normal conversations. "When I hear music it's as if a butcher's knife cuts through my head. It's way too sharp!". So Carol prefers to take her hearing aids out, missing most of the fun. Betty is so happy that her hearing instruments simply know where they are and adapt to their environment.

source: Roland Mathijssen
Embedded Systems Institute
Eindhoven

Value and Challenges in this story

Customer objectives

Application

Value proposition in this story:

quality of life:

active participation in different social settings

usability for nontechnical elderly people:

"intelligent" system is simple to use

loading of batteries

Conceptual

Realization

Challenges in this story:

Intelligent hearing instrument

Battery life — at least 1 week

No buttons or other fancy user interface on the hearing instrument, other than a robust On/Off method

The user does not want a technical device but a solution for a problem

Instrument can be adapted to the hearing loss of the user

Directional sensitivity (to prevent the so-called cocktail party effect)

Recognition of sound environments and automatic adaptation (adaptive filtering)

source: Roland Mathijssen, Embedded Systems Institute, Eindhoven

Exercise Story Telling

- Create a story using the criteria.
- Transform the story into a case (functional, as well as quantitative).
- Perform a short design exploration based on the case.
- Improve the story based on the use in the case and the design.
- Use time boxes to ensure that you make all the indicated steps.

Module System Architecture Context

by *Gerrit Muller* Buskerud University College and Buskerud University College
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

The system architecture process is positioned in a wider context: First in the business context, then in the Product Creation Process context.

January 22, 2023
status: concept
version: 1.1

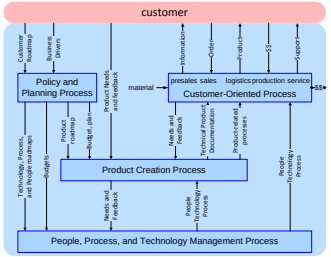
USN ESI

Process Decomposition of a Business

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

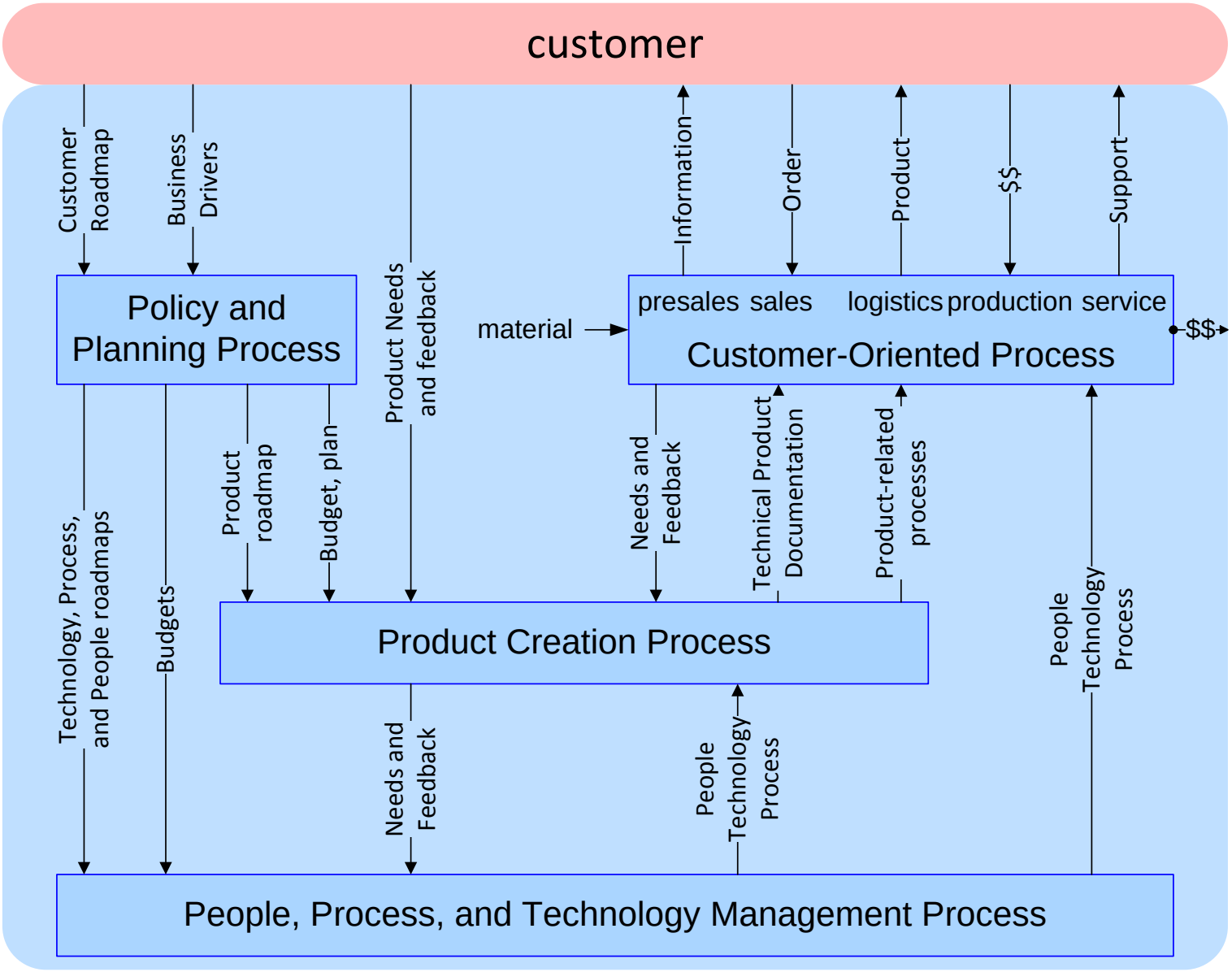
Abstract

This article positions the system architecture process in a wider business scope. This positioning is intended to help understanding the processes in which the system architect (or team of system architects) is involved.
It focuses on an organization that creates and builds systems consisting of hardware and software. Although other product areas such as solution providers, services, courseware, et cetera also need system architects, the process structure will deviate from the structure as presented here.

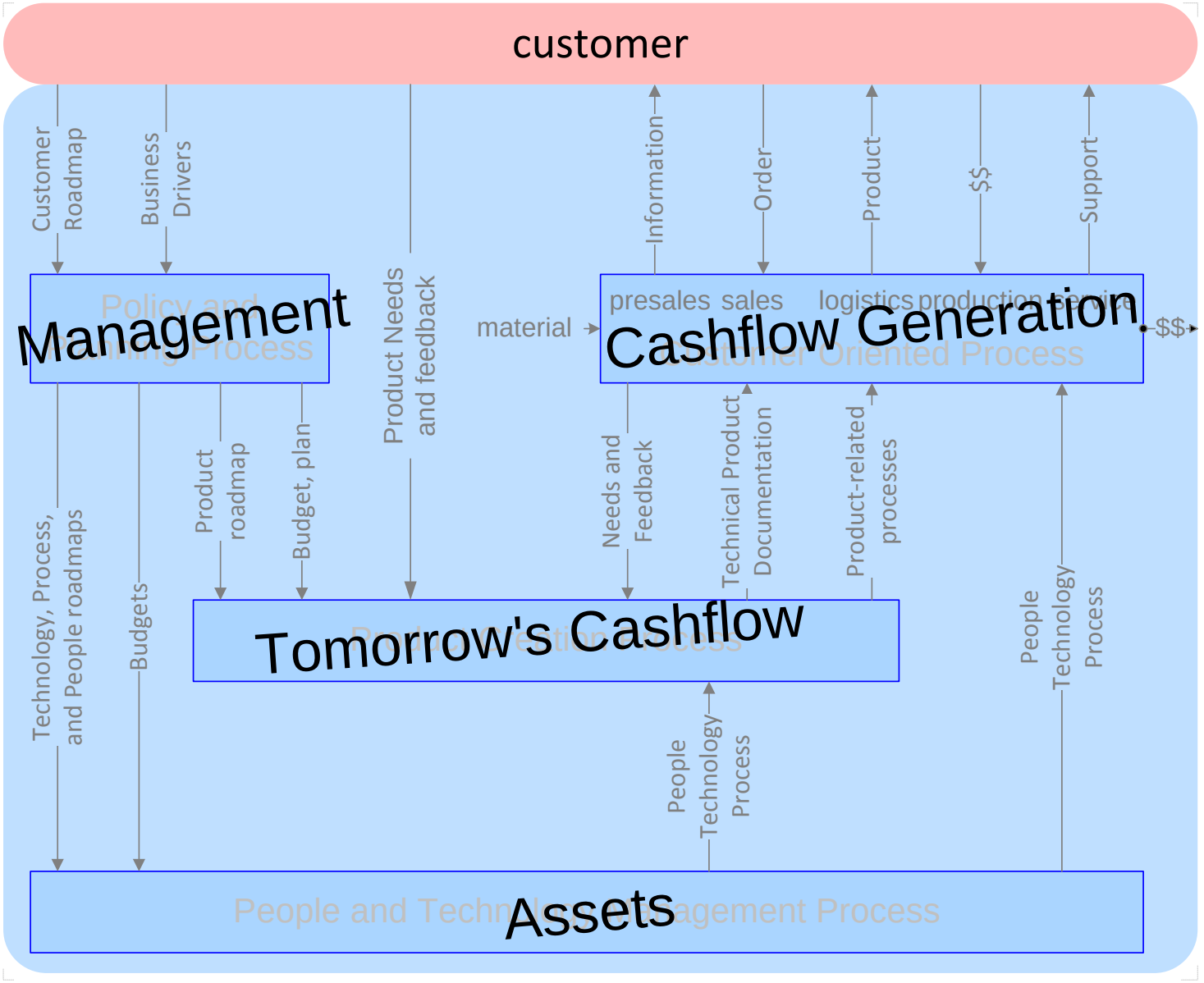


January 22, 2023
status: concept
version: 1.1

Simplified Decomposition of the Business



Financial Characterization of Decomposition



Multiple Instances per Process

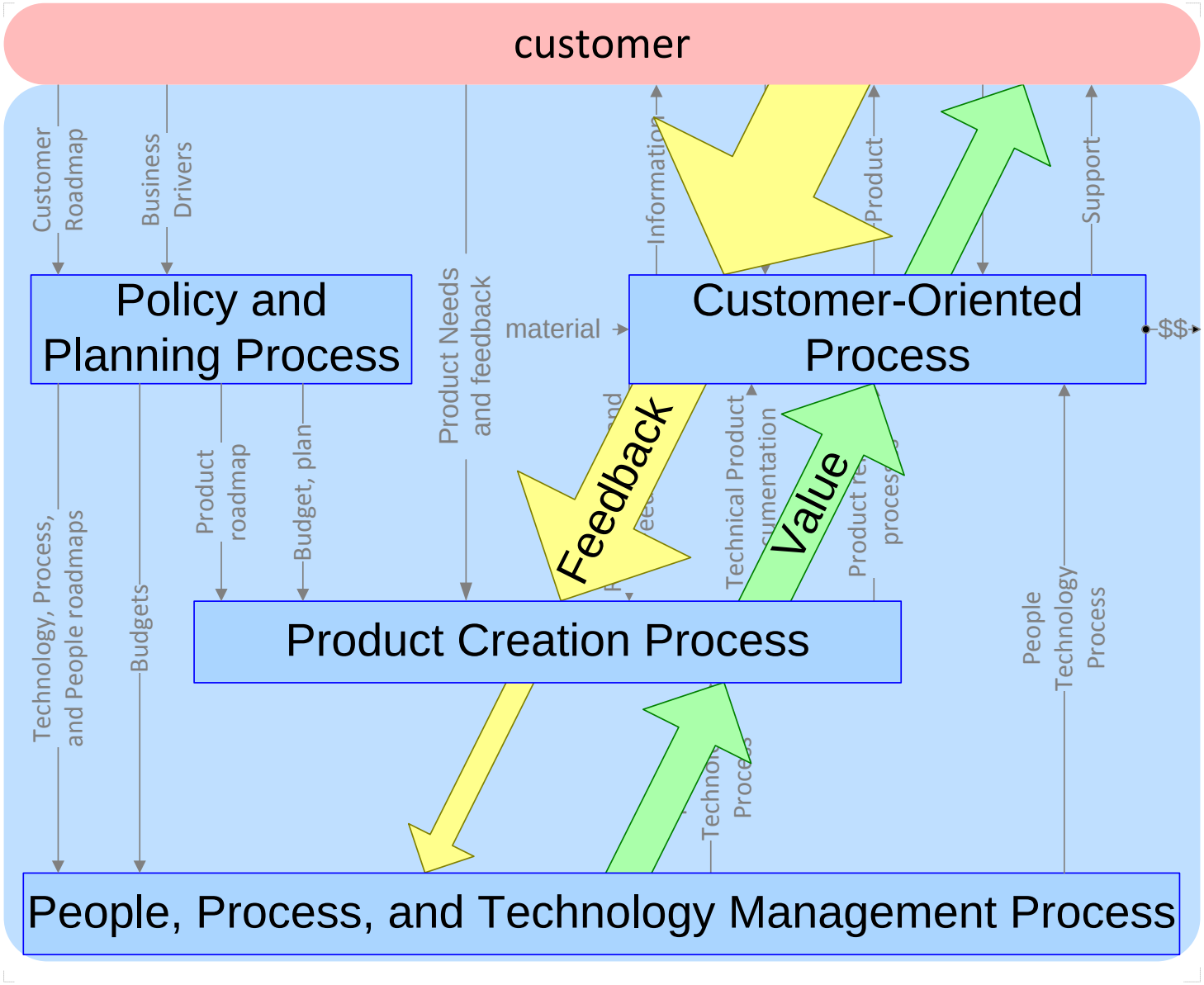
Customer Oriented Process: Depends on geography, customer base, and supply chain.

Product Creation Process: One per entity to be developed, where such an entity can be a product family, a product, or a subsystem.

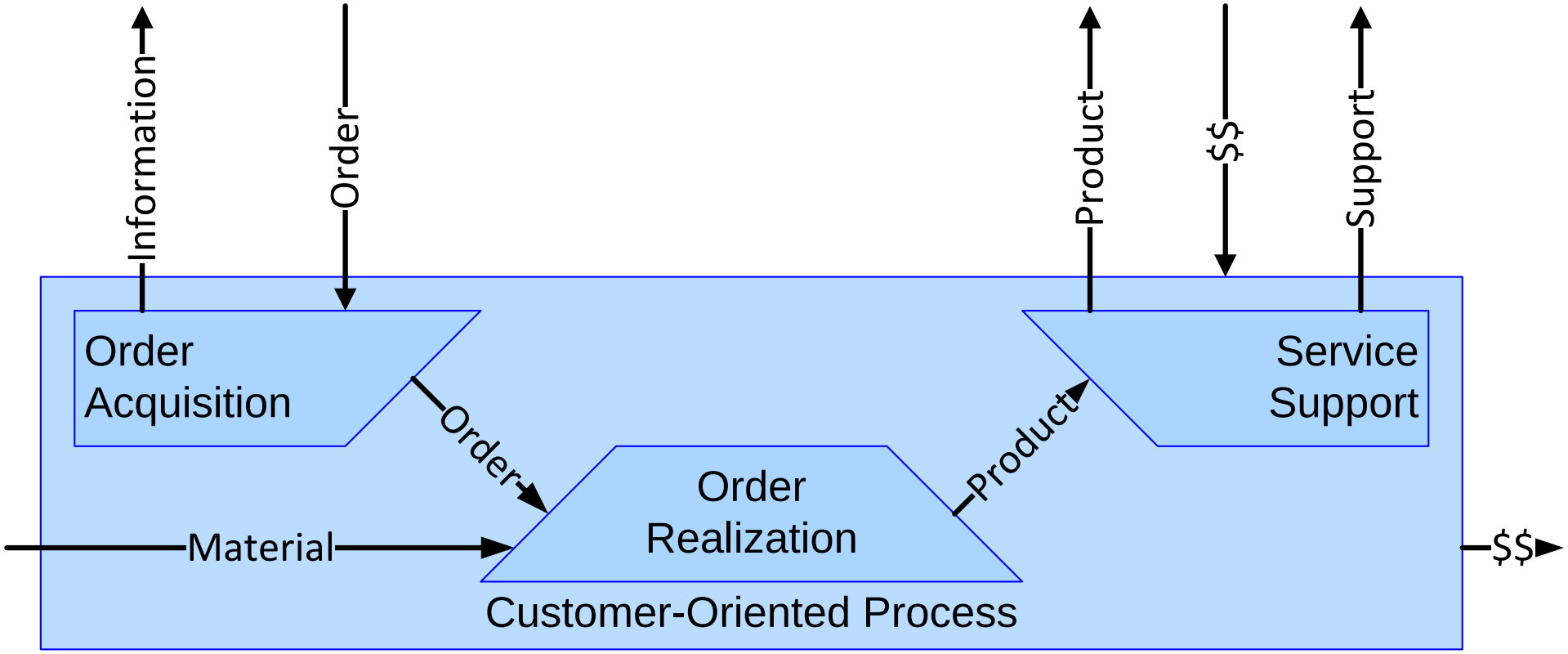
People and Technology Management Process: One per “competence”, where a competence is a cohesive set of technologies and methods.

Policy and Planning Process: One per business. This is the pro-active integrating process.

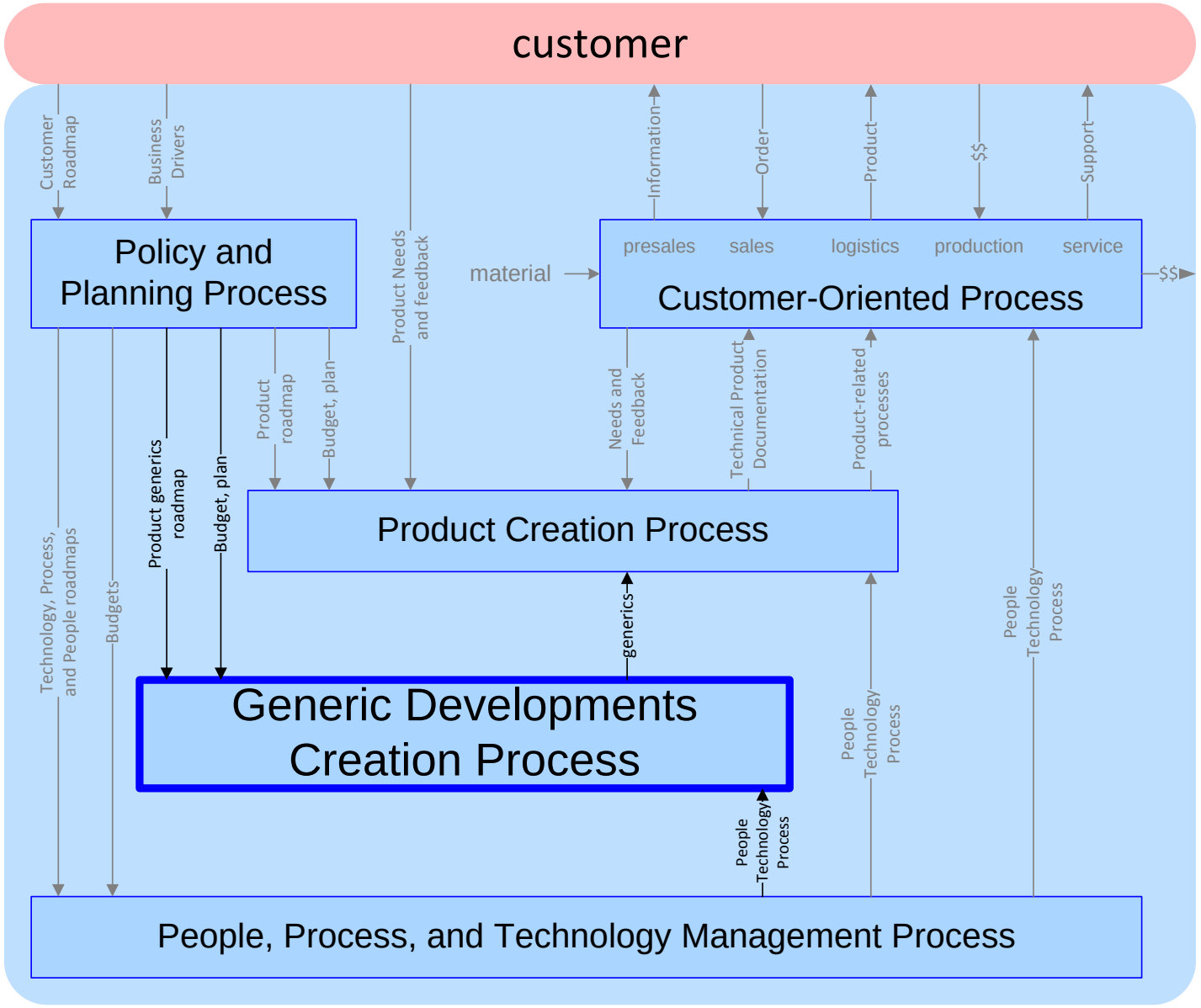
The Value Chain and the Opposite Feedback Flow



Decomposition of the Customer Oriented Process



Extended with Generic Developments



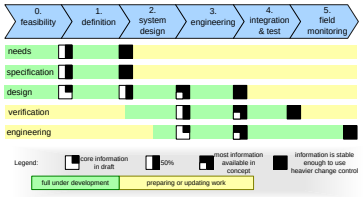
The Product Creation Process

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

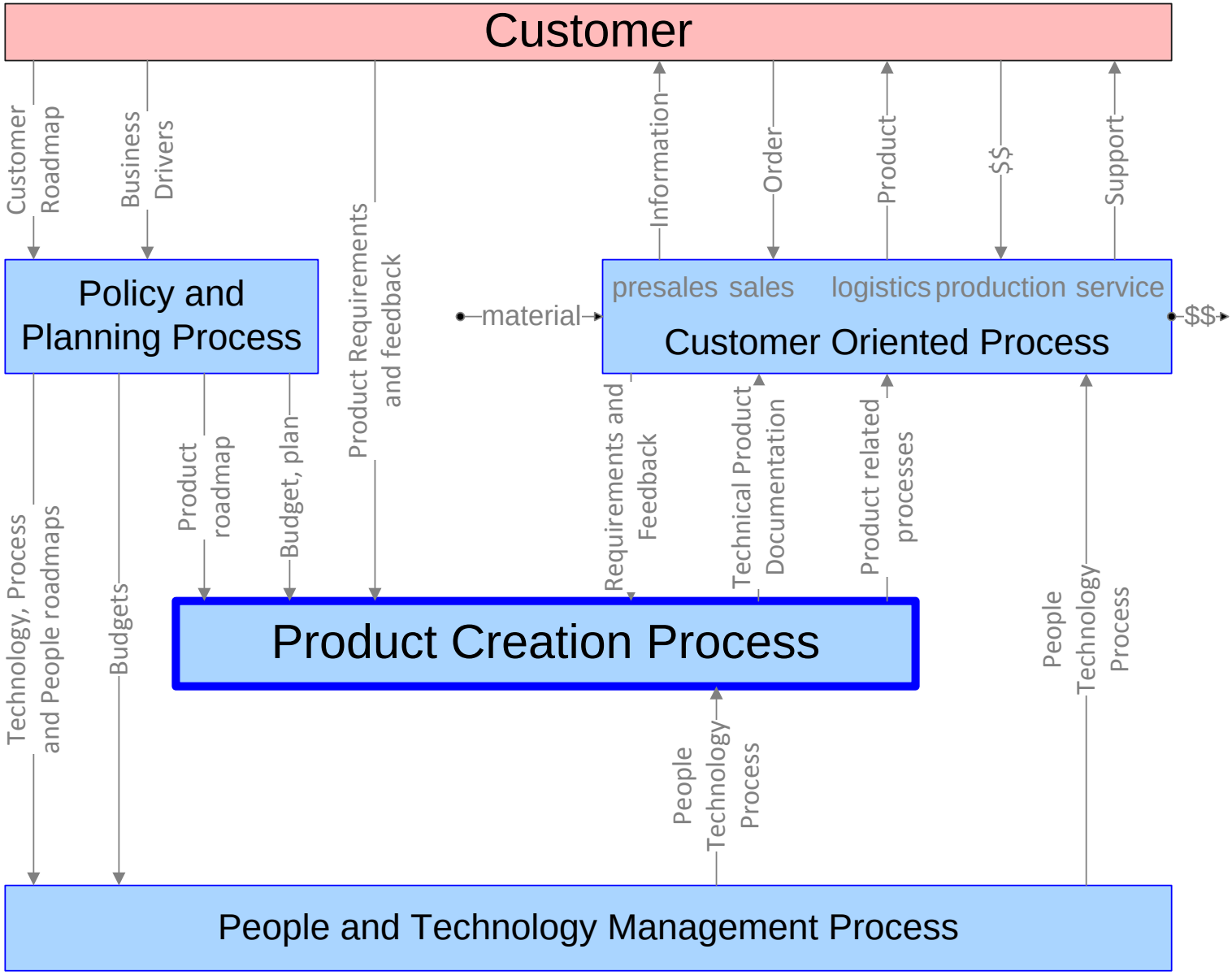
Abstract

The Product Creation Process is described in its context. A phased model for Product Creation is shown. Many organizations use a phased model as blueprint for the way of working. The operational organization of the product creation process is discussed, especially the role of the operational leader.

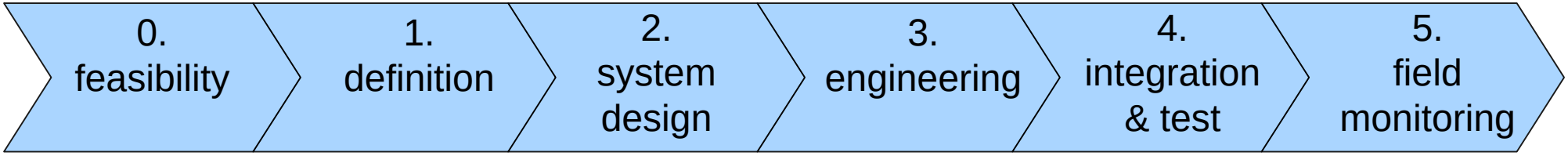
January 22, 2023
status: concept
version: 2.2



The Product Creation Process in Business Context



Phasing of the PCP at Business Level



sales

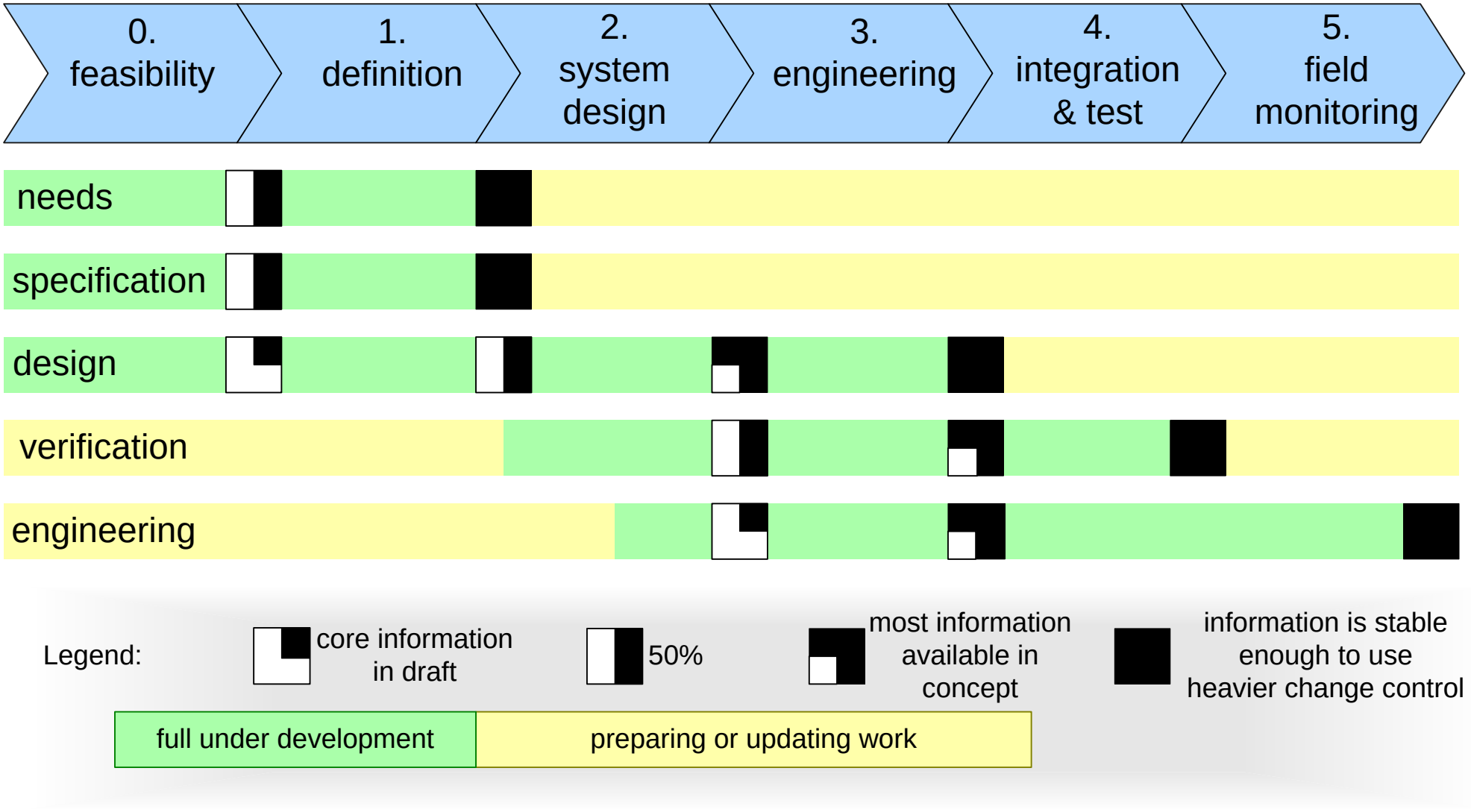
logistics

production

service

development & engineering: marketing, project management, design

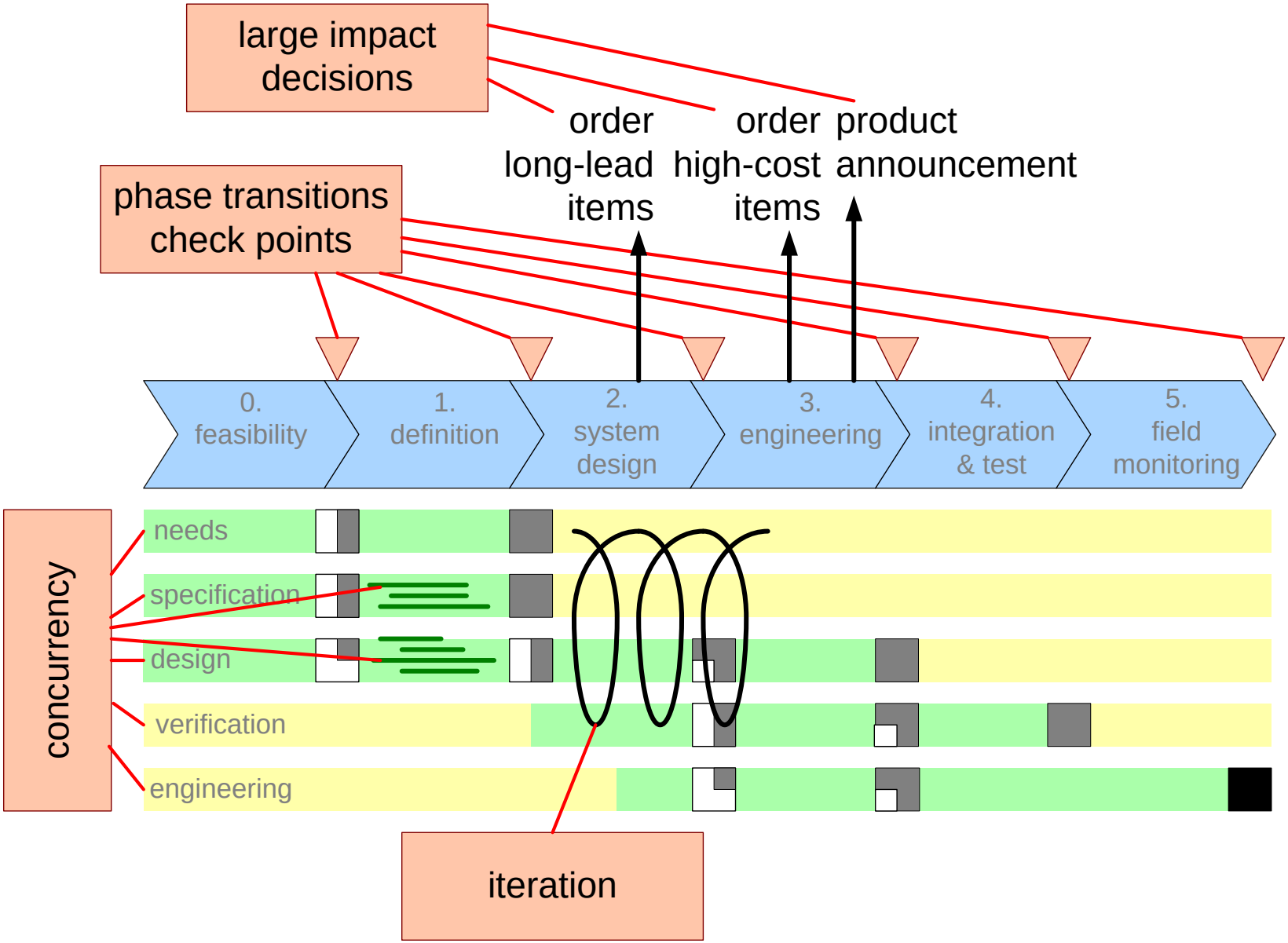
Phasing the Design Control Process



Advantages and Disadvantages of a Phased Process

benefits	*disadvantages*
blueprint: how to work	following blueprint blindly
reuse of experience	too bureaucratic
employees know *what* and *when*	transitions treated black and white
reference for management	

Characteristics of a Phase Model



Decisions and Phase Transitions

Define a minimal set of *large-impact* decisions.

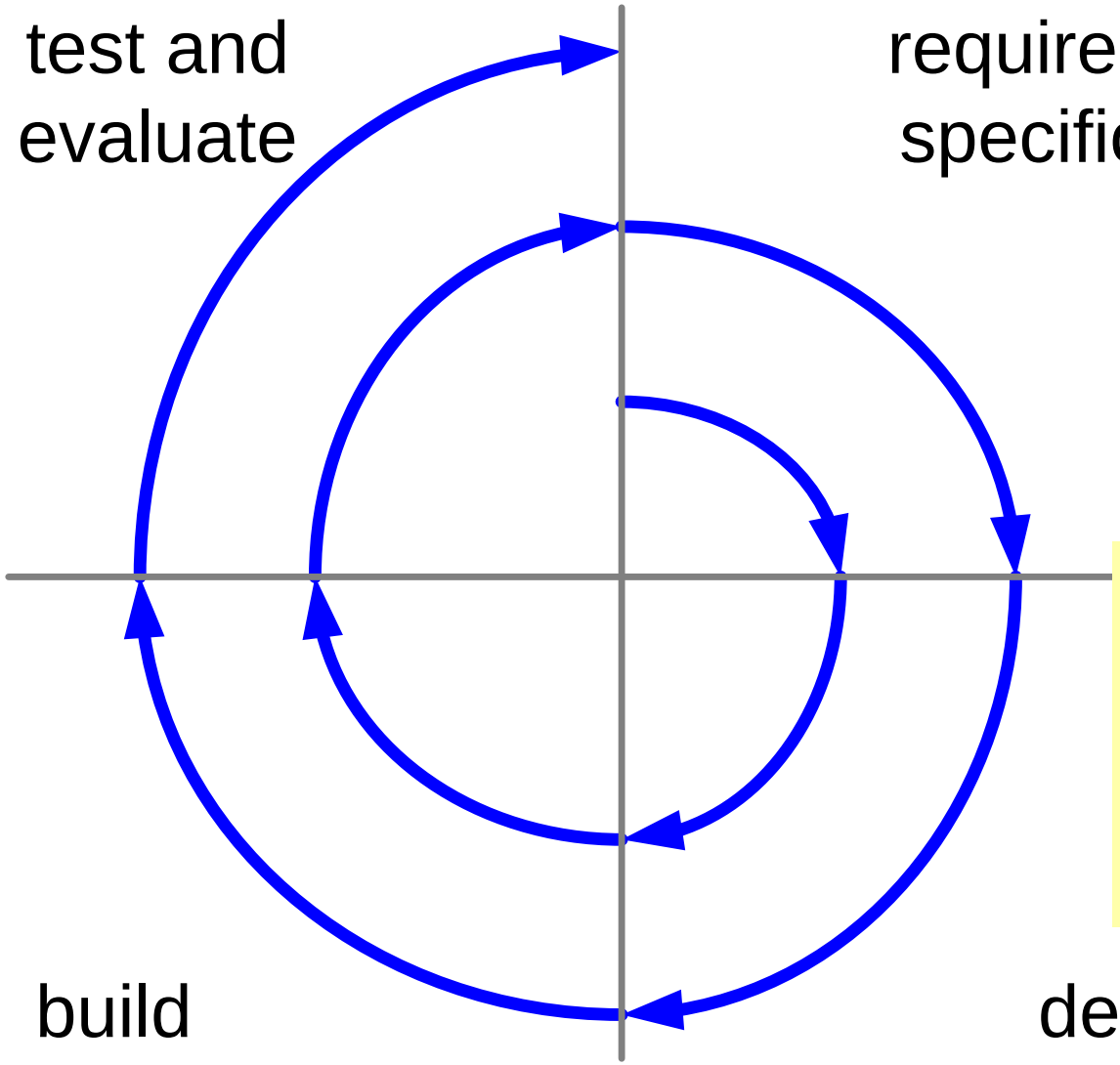
Define the mandatory and supporting information required for the decision.

Schedule a decision after the appropriate phase transition.

Decide explicitly.

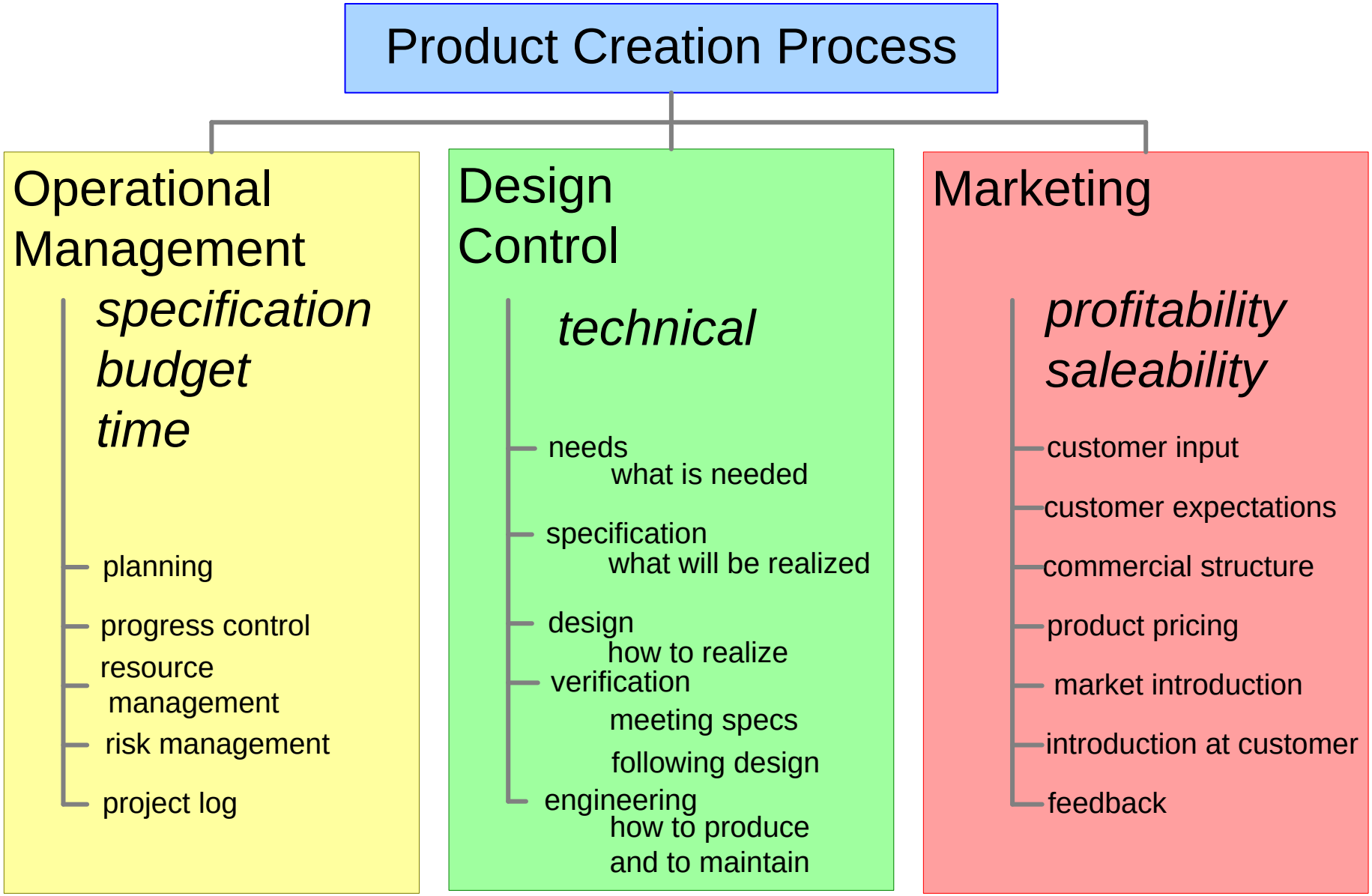
Communicate the decision clearly and widely.

Evolutionary PCP model

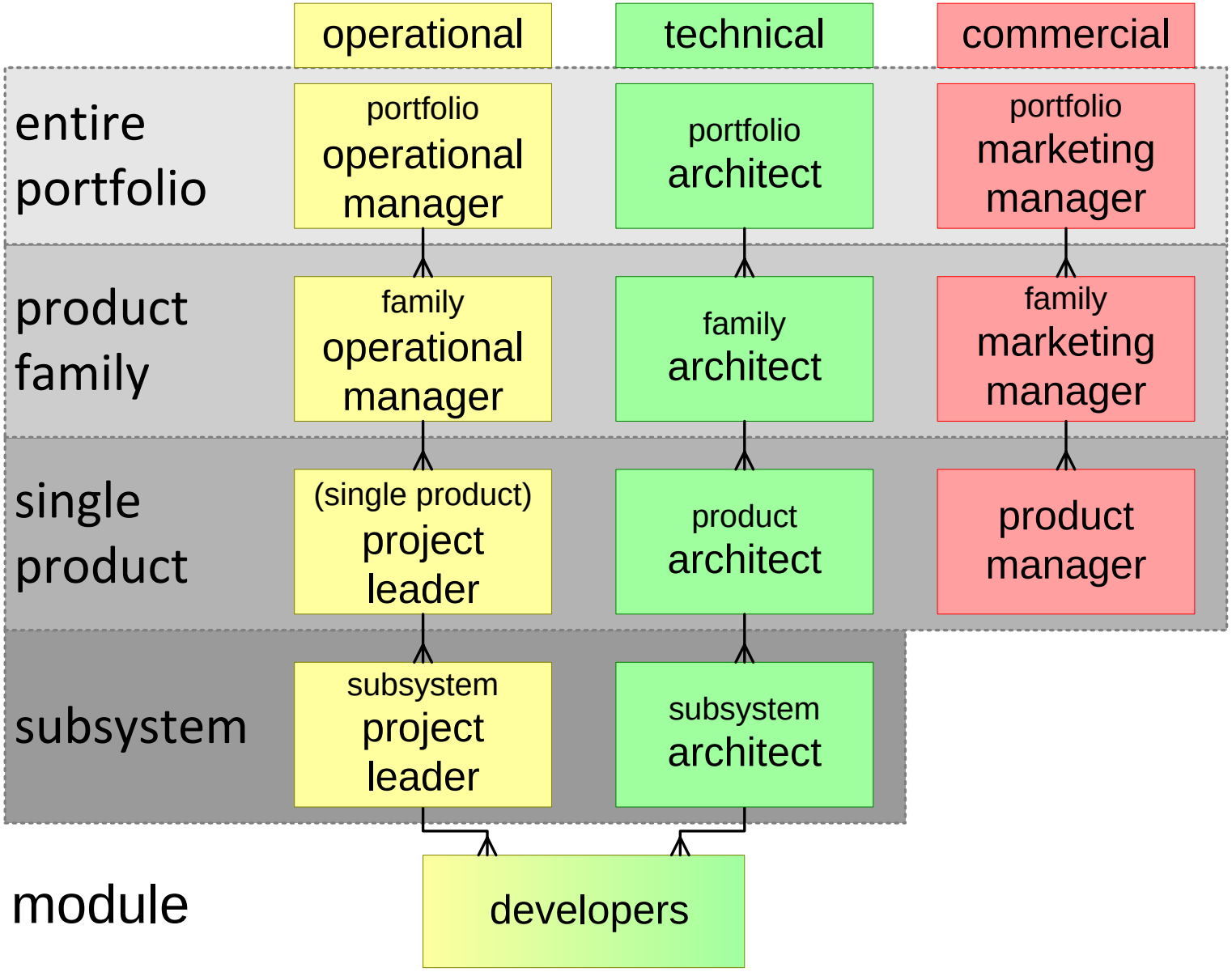


2% of budget (EVO)
2 weeks (XP)
up to 2 months
per cyclus

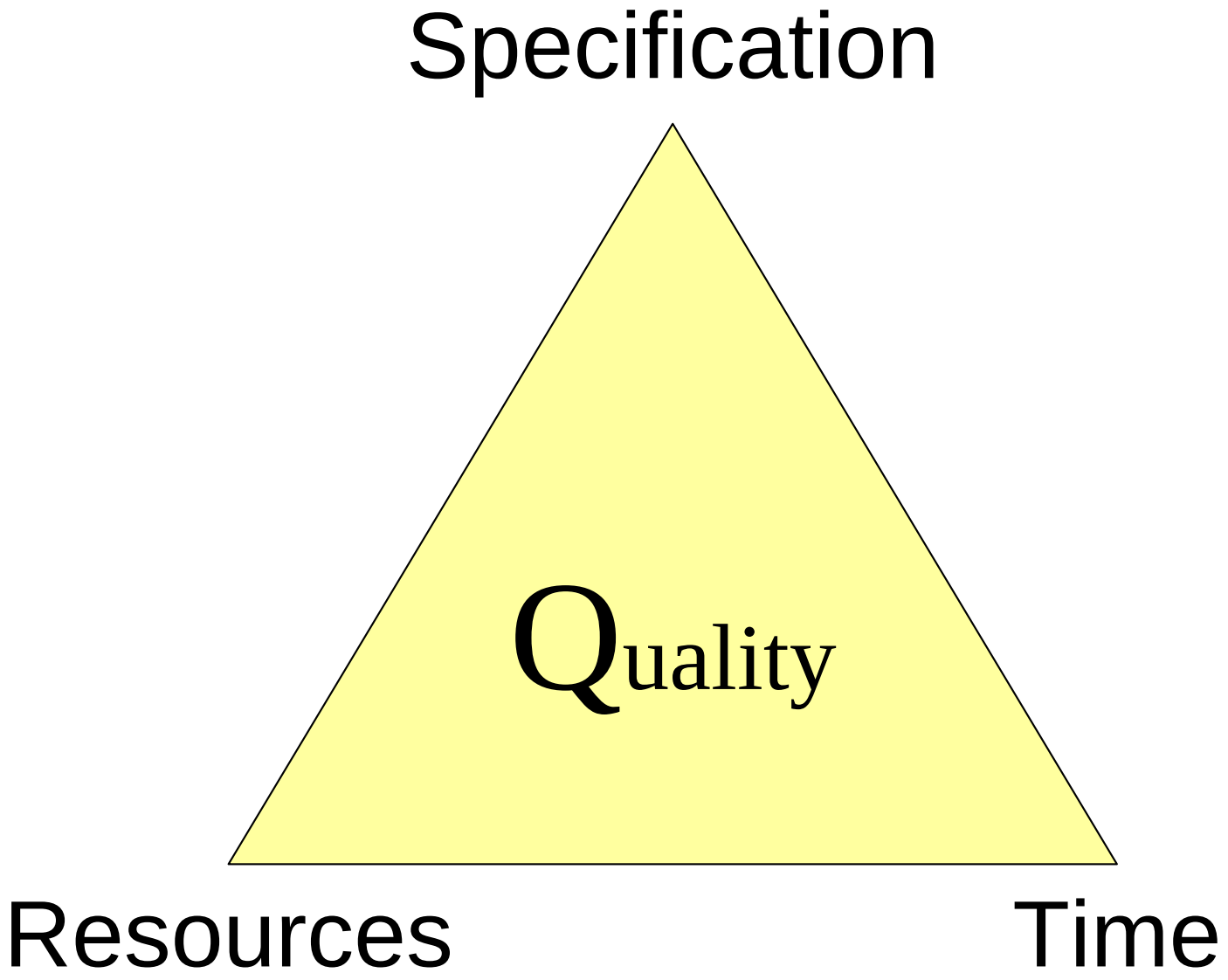
Decomposition of the Product Creation Process



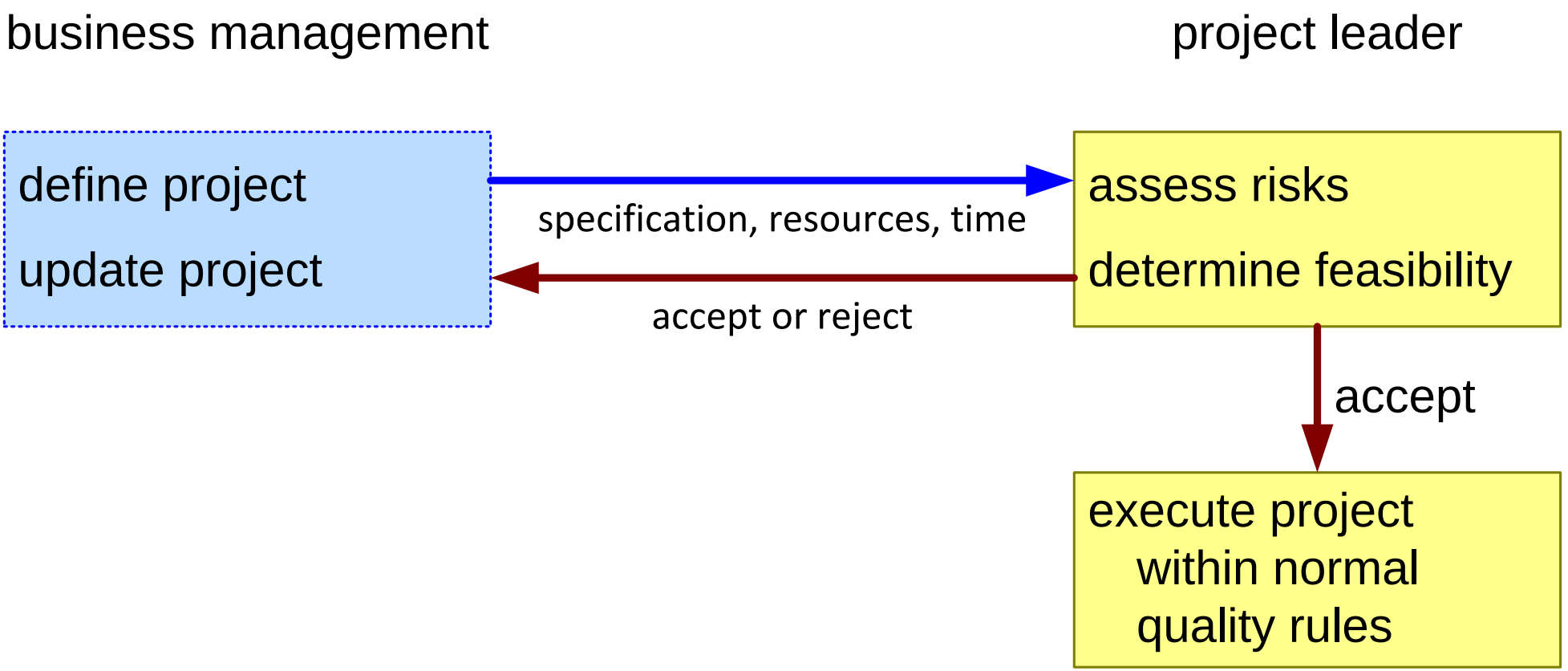
Operational Organization of the PCP



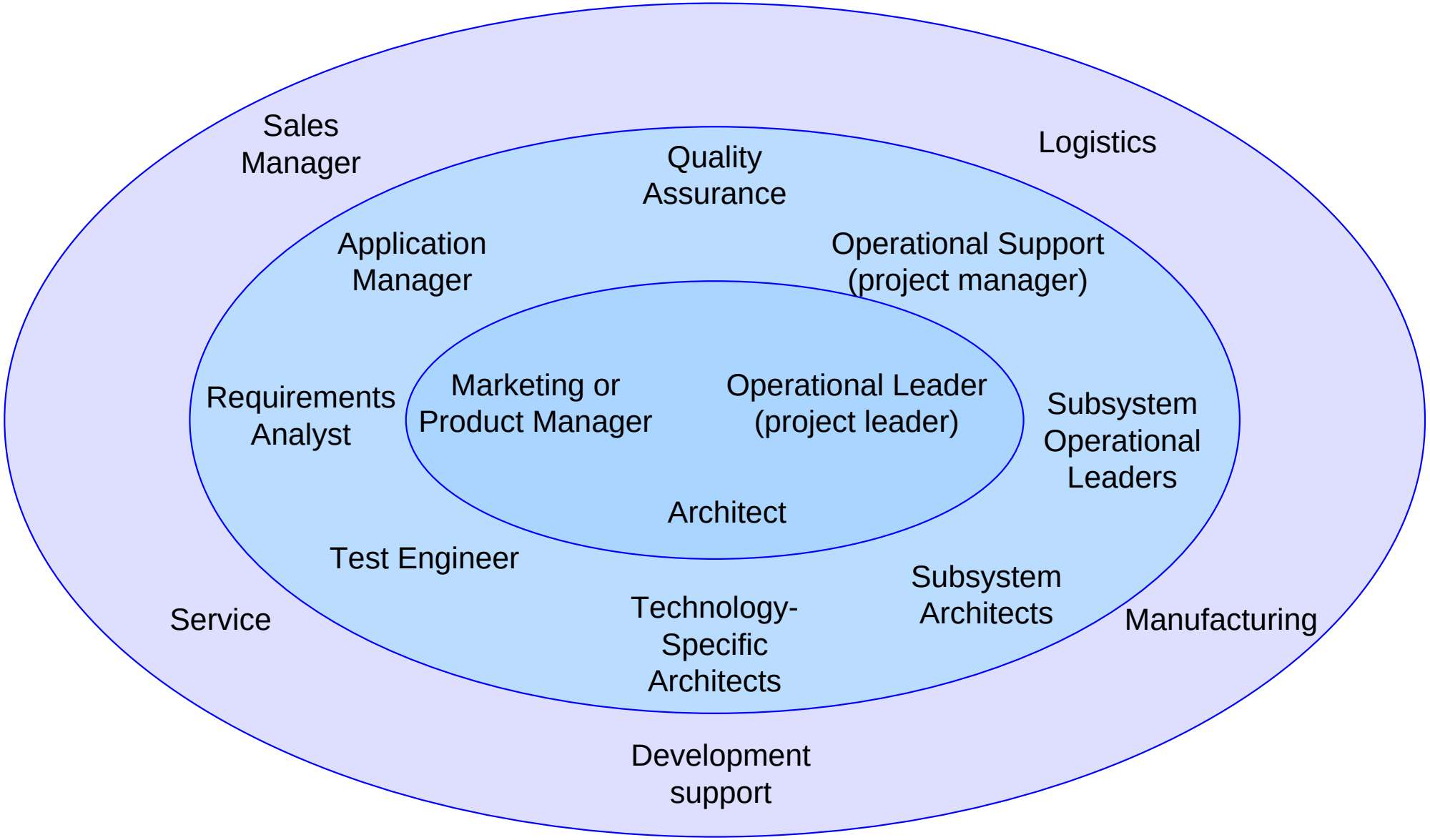
Prime Responsibilities of the Operational Leader



The Rules of the Operational Game



Operational Teams

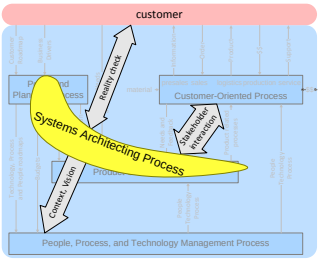


The System Architecture Process

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

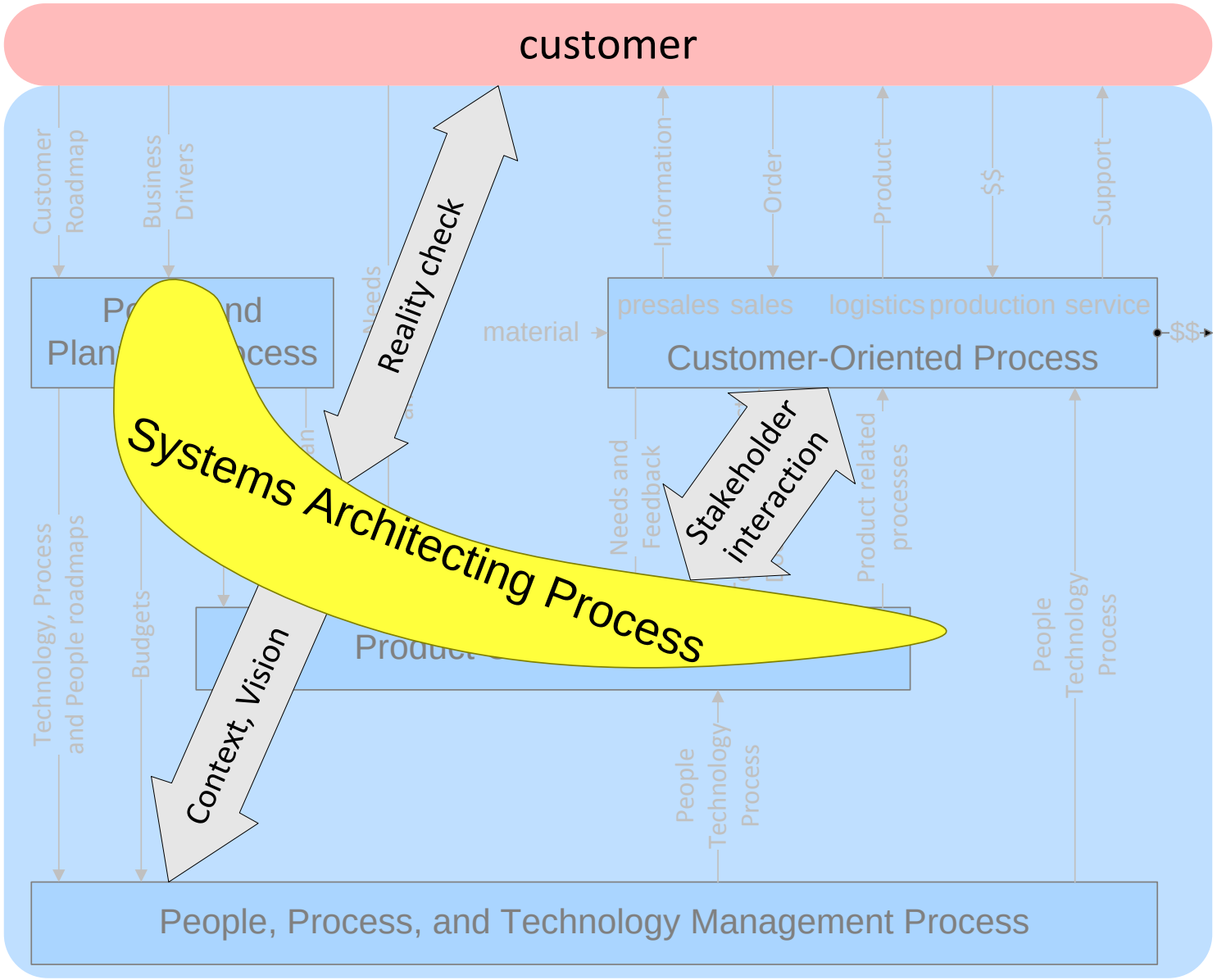
Abstract

The System Architecture Process is positioned in the business context. This process bridges the gap between the Policy and Planning Process and the Product Creation Process.
The purpose of the System Architecture Process is to provide the Integral Technical overview and consistency, and to maintain the integrity over time. Subjective characteristics as elegance and simplicity are key elements of a good architecture.
The scope of the system architecture process is illustrated by showing 5 views used in a reference architecture, ranging from Customer Business to Realization.

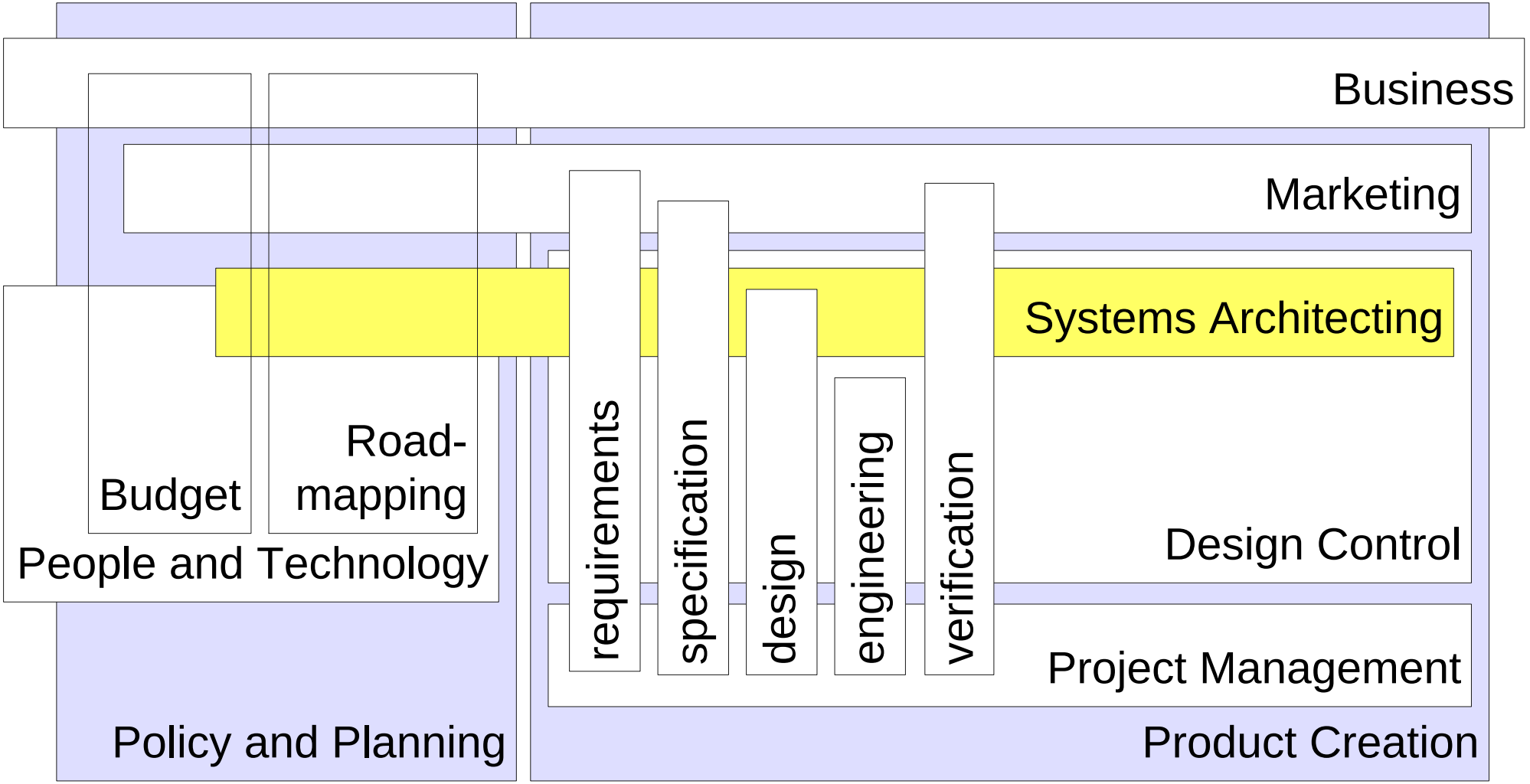


January 22, 2023
status: concept
version: 2.3

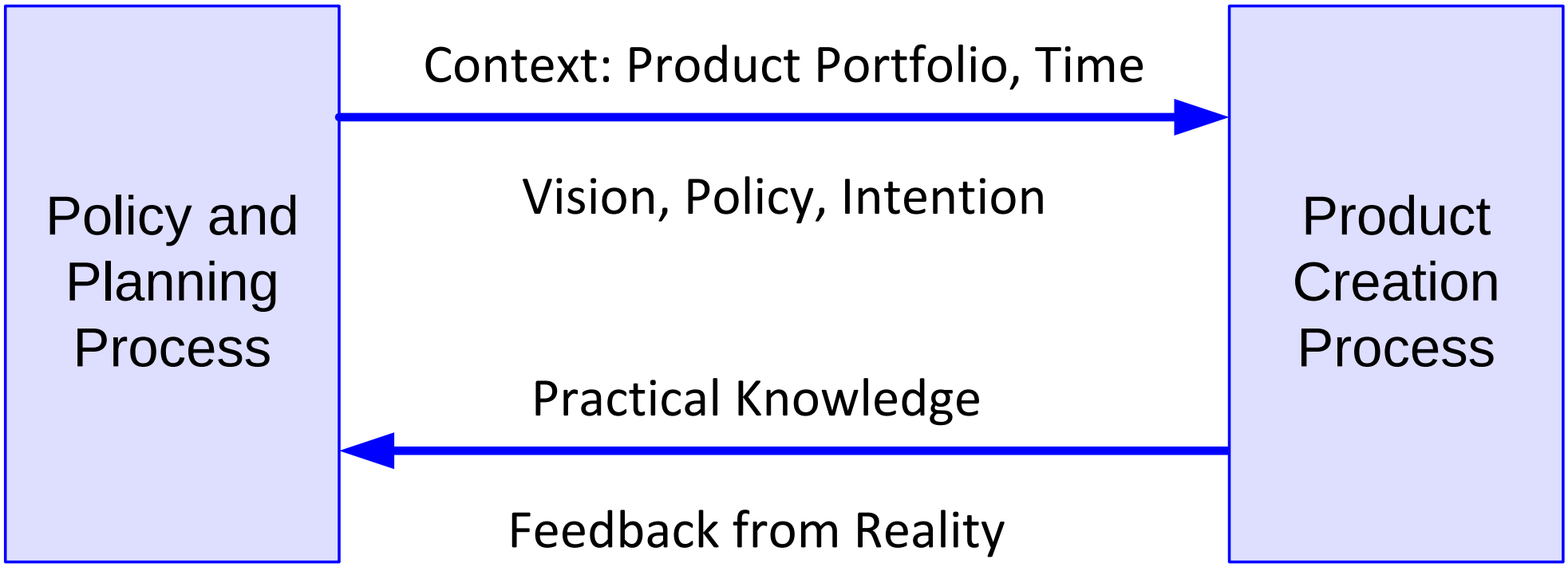
System Architecting Process in Business Context



Map of System Architecting Process and Neighborhood



System Architecting Relation between PPP and PCP



System Architecting Key Issues

key words

balance

consistency

integrity

simplicity

elegance

stakeholder satisfaction

balancing acts

External ⟷ internal requirements

Short term needs ⟷ long term interests

Efforts ⟷ risks from requirements to verification

Mutual influence of detailed designs

Value ⟷ costs

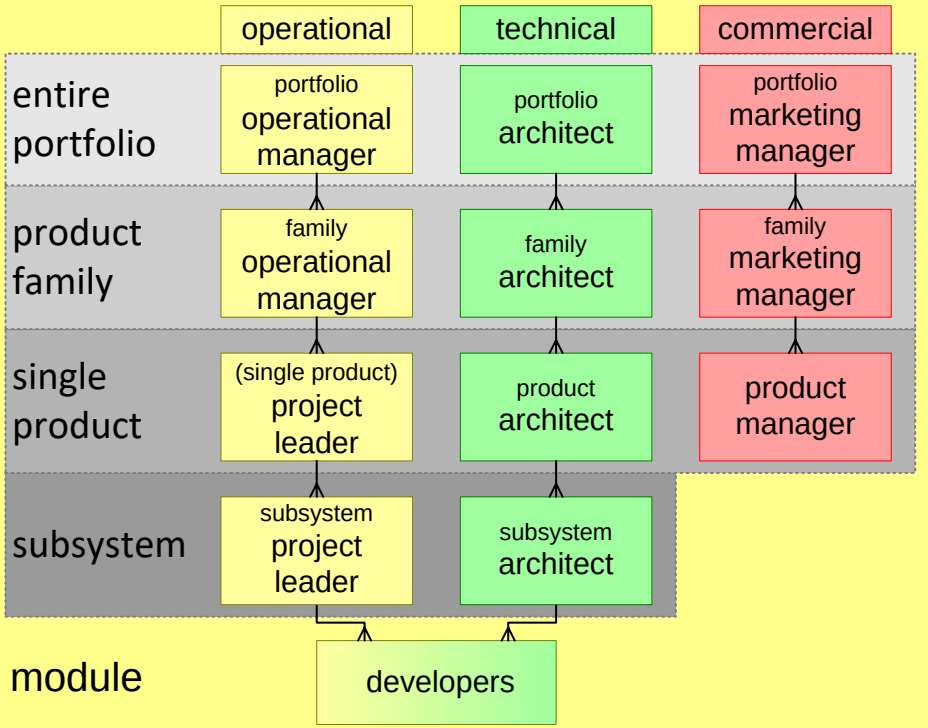
example trade-offs

Exercise Product Creation Process

Make a map with names of individuals in the **operational organization** of one project and its context

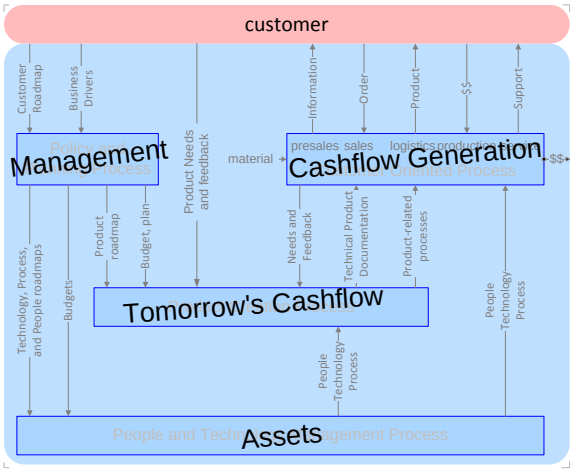
Identify the **relationships** of the **project core team**:

- **geographical**
- **organizational**
- **psychological**



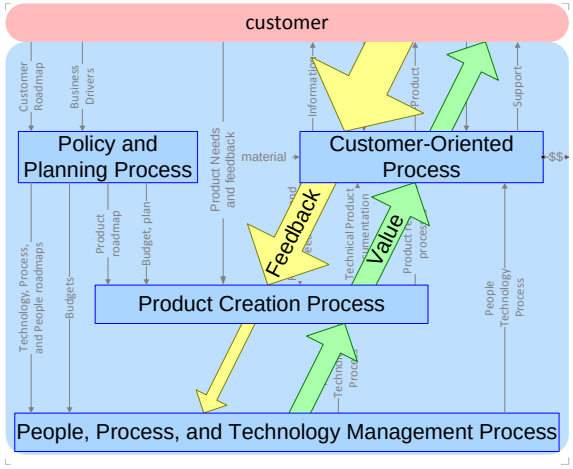
Process Decomposition of a Business

Importance in Financial terms



intentionally left blank

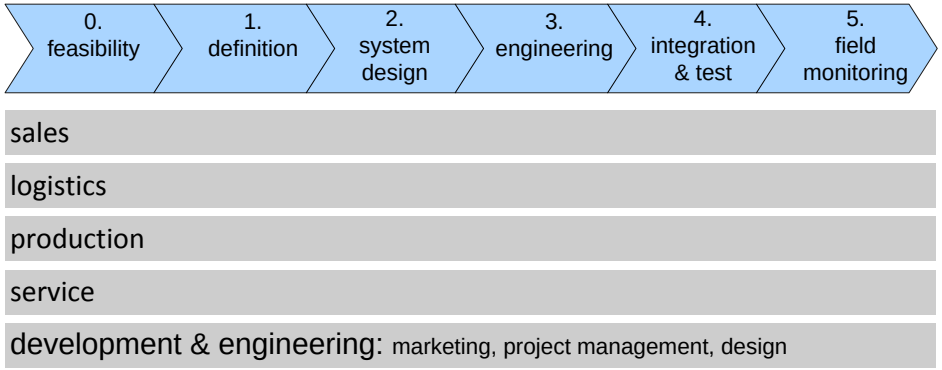
Value Chain and Feedback Flow



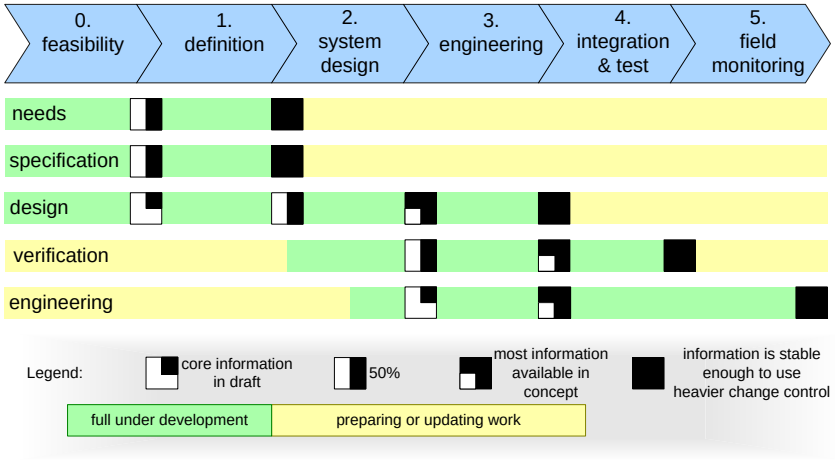
intentionally left blank

Product Creation Process

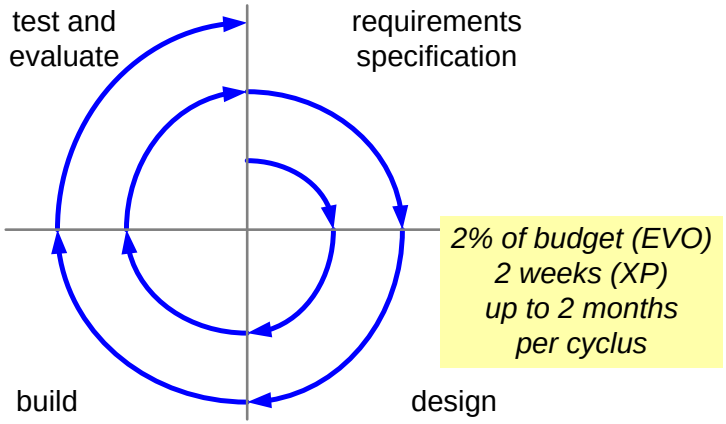
PCP involves **all** disciplines, much more than D&E



Phased Process



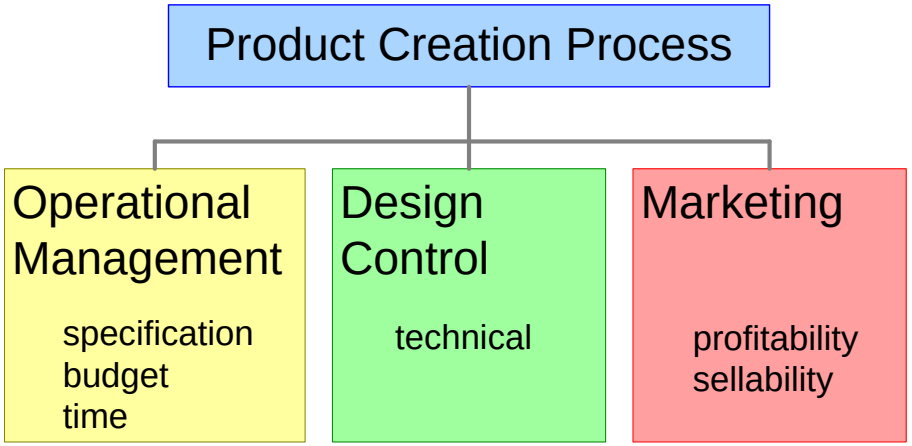
Incremental Development



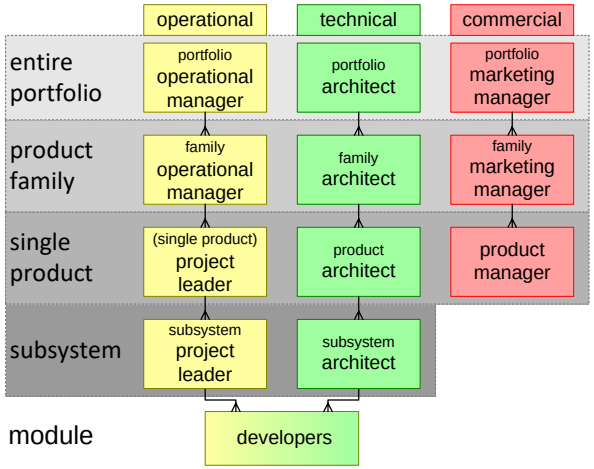
intentionally left blank

PCP Decomposition and Operational Management

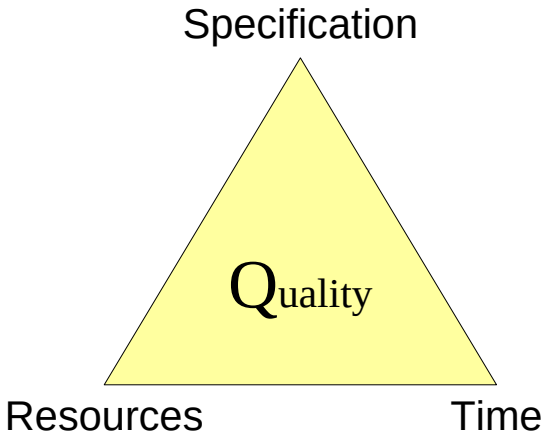
PCP decomposition



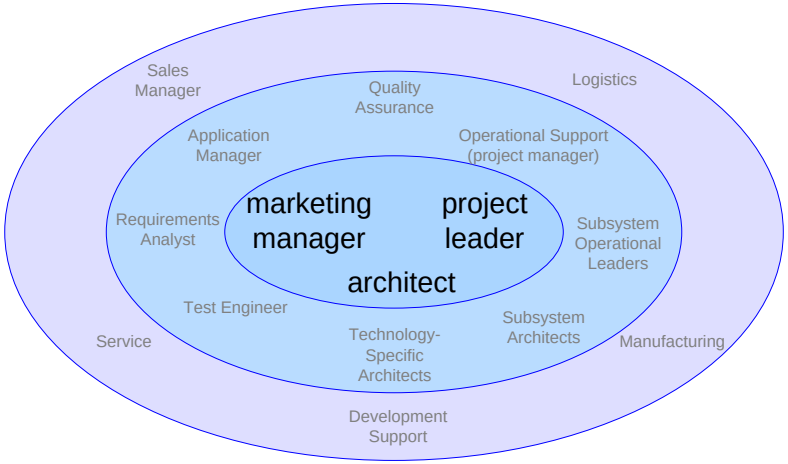
Architecture at all levels; From portfolio to subsystem



Operational Commitment

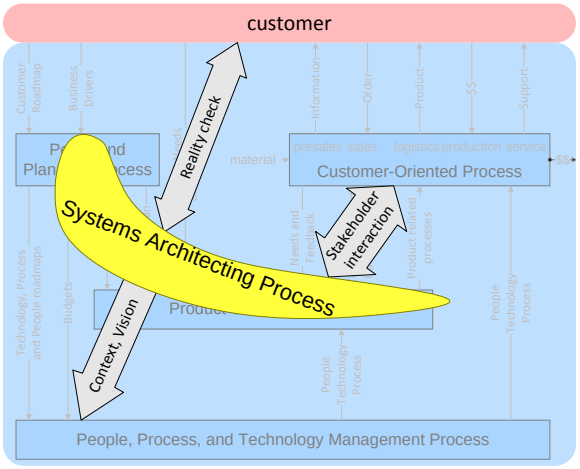


Core: Operational + Technical + Commercial



System Architecture Process

In Business Context



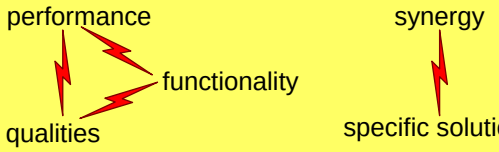
Key Issues

key words
balance
consistency
integrity
simplicity
elegance
stakeholder satisfaction

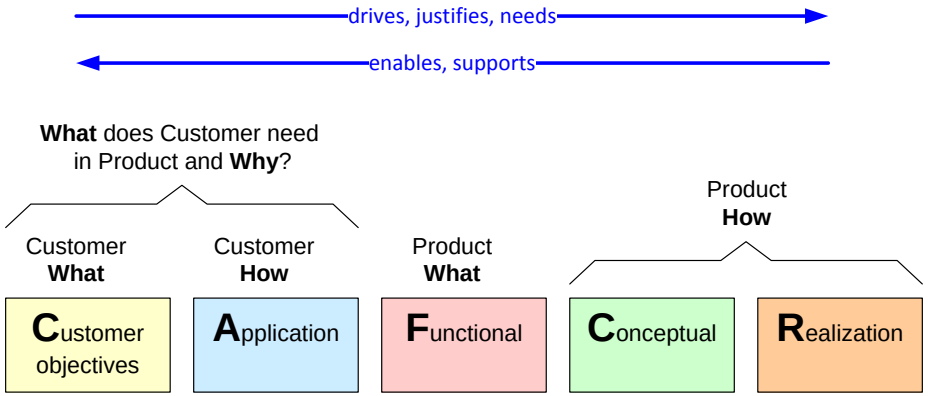
balancing acts

External ↔ internal requirements
Short term needs ↔ long term interests
Efforts ↔ risks from requirements to verification
Mutual influence of detailed designs
Value ↔ costs

example trade-offs



5 Views



intentionally left blank

Role and Task of the System Architect

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

The role and the task of the system architect are described in this module.

January 22, 2023
status: preliminary draft
version: 1.0

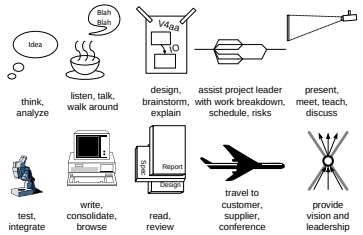
The Role and Task of the System Architect

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

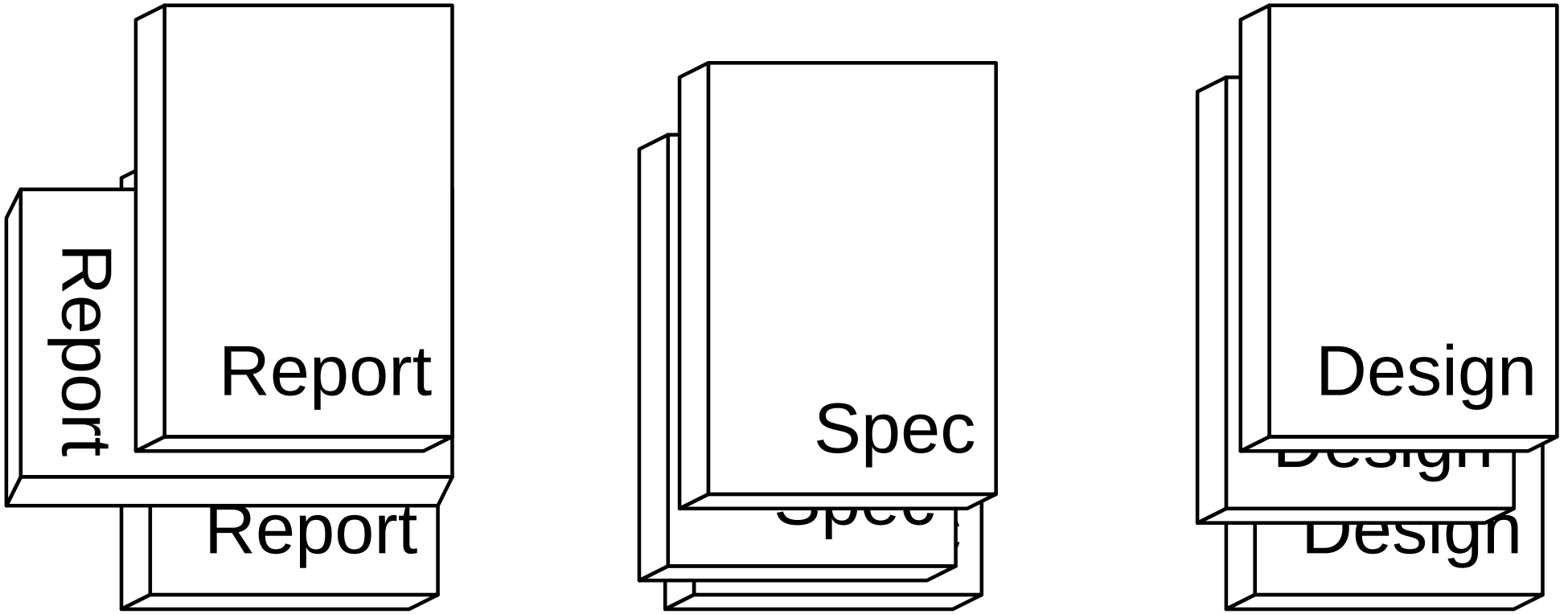
Abstract

The role of the system architect is described from three viewpoints: deliverables, responsibilities and activities. This description shows the inherent tension in this role: a small set of hard deliverables, covering a fuzzy set of responsibilities, hiding an enormous amount of barely visible day-to-day work.

January 22, 2023
status: concept
version: 2.0



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)*

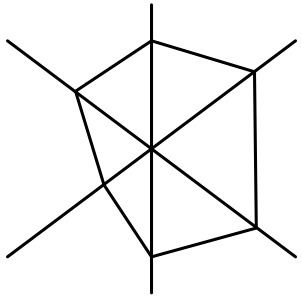
Verification Specification *(how the system will be verified)*

Verification Report *(the result of the verification)*

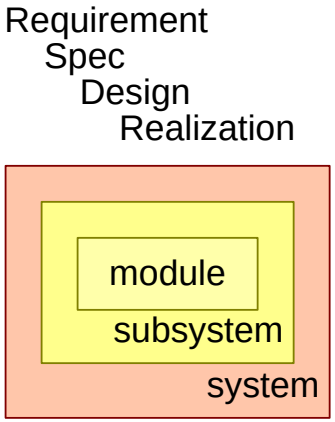
Feasibility Report *(the results of a feasibility study)*

Roadmap

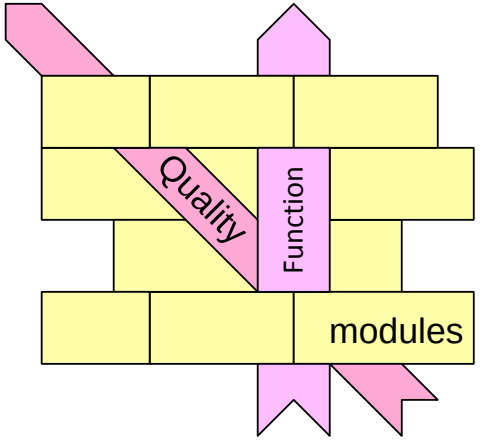
Responsibilities of the System Architect



Balance

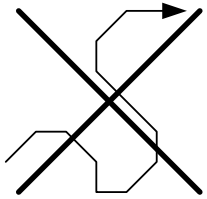


Consistency



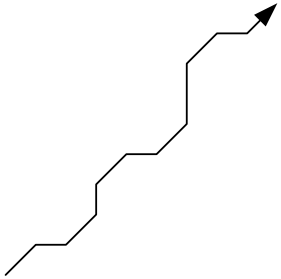
Decomposition
Integration

Overview

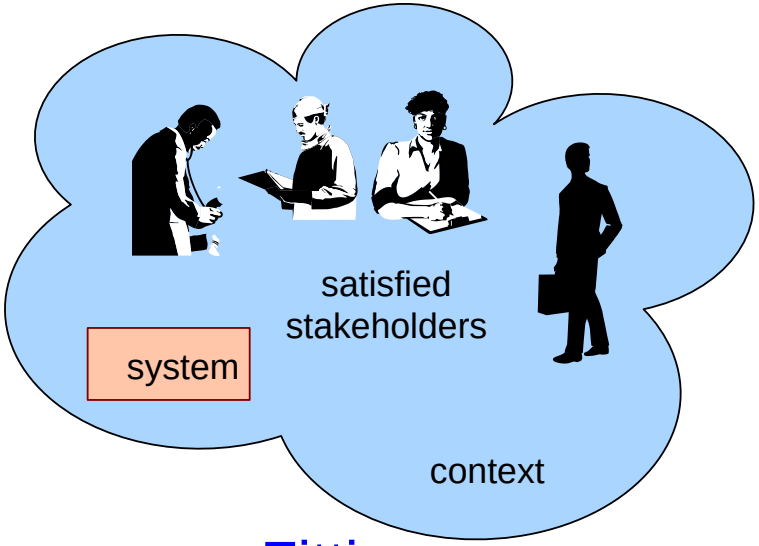


KISS

Elegance
Simple



Integrity

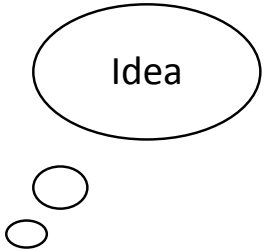


Fitting

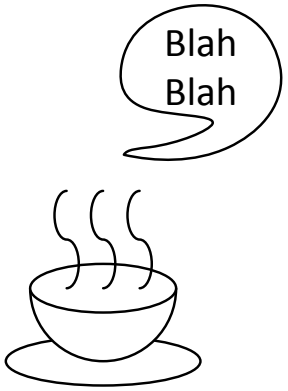
Examples of Secondary Responsibilities

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

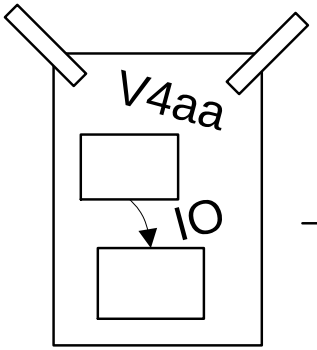
What does the System Architect do?



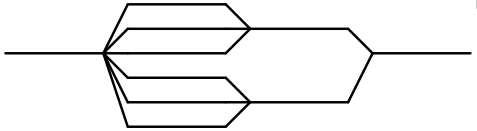
think, analyze



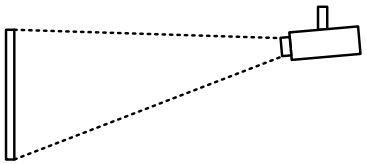
listen, talk, walk around



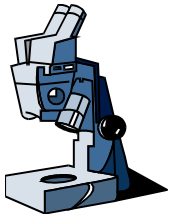
design, brainstorm, explain



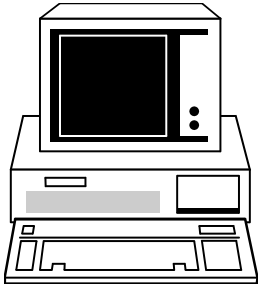
assist project leader with work breakdown, schedule, risks



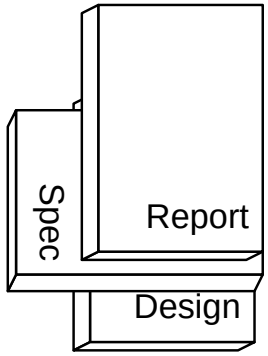
present, meet, teach, discuss



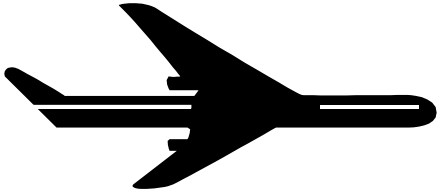
test, integrate



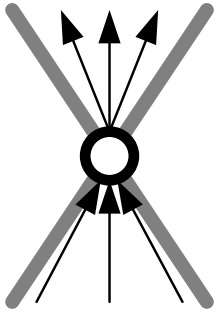
write, consolidate, browse



read, review

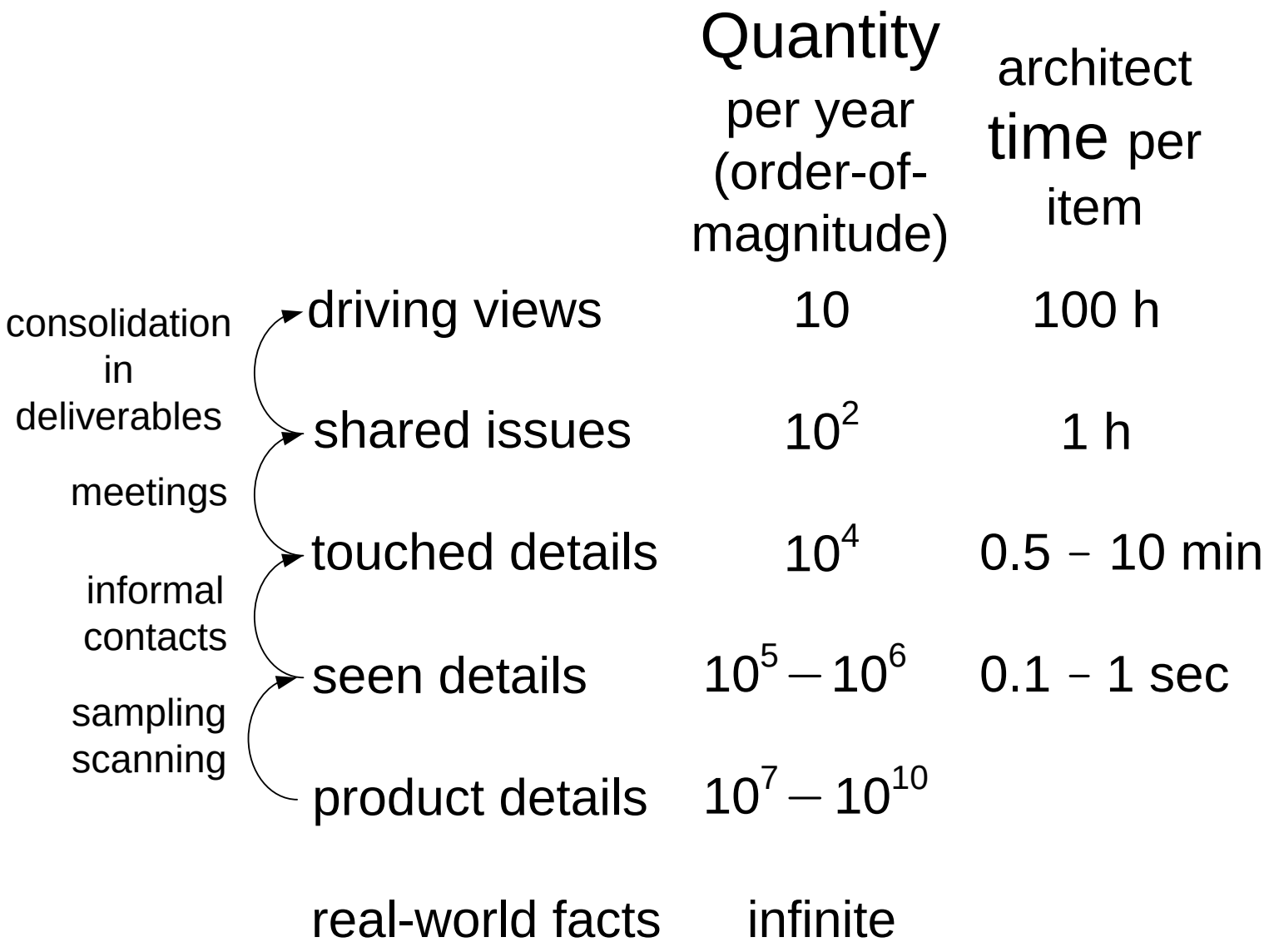


travel to customer, supplier, conference



provide vision and leadership

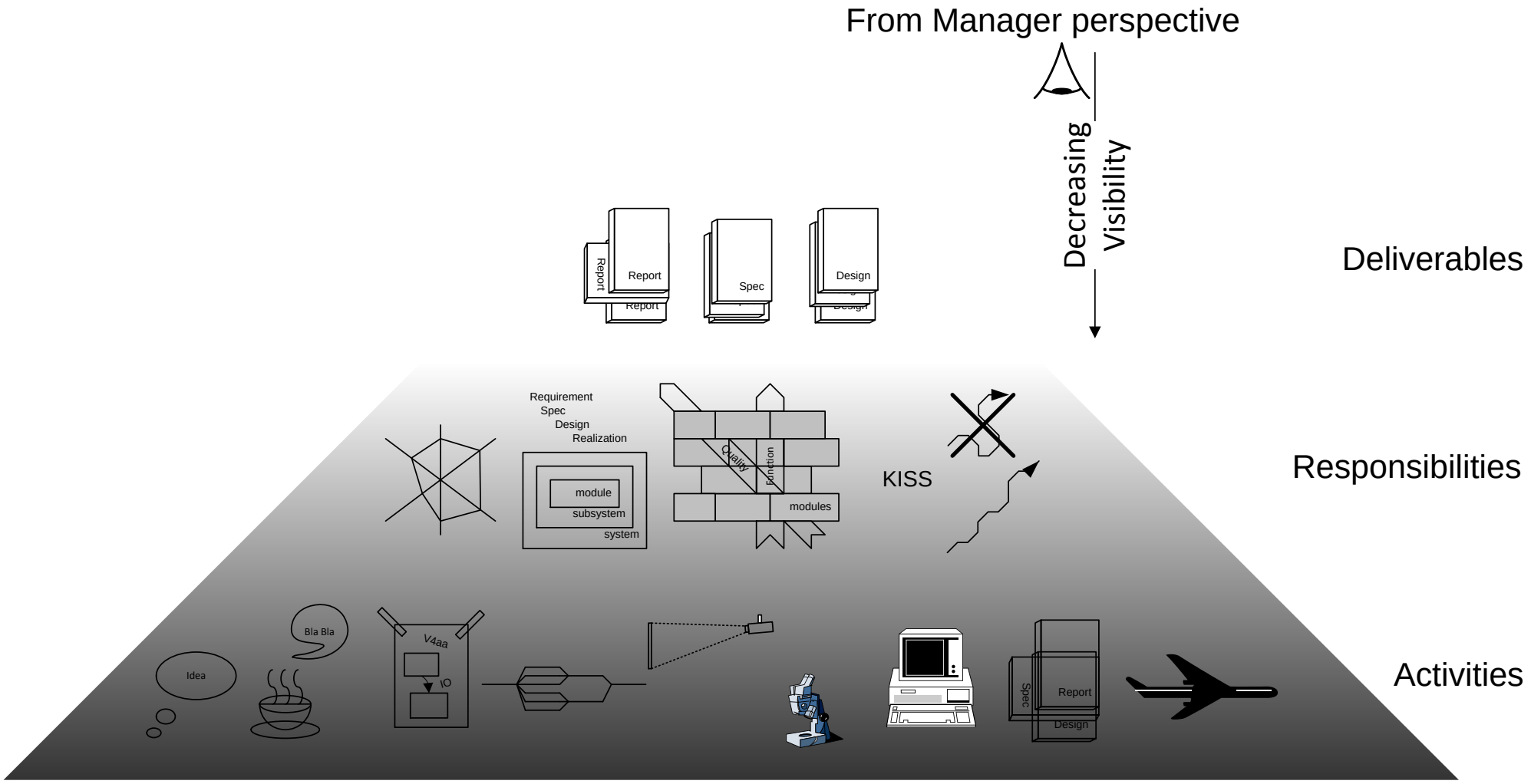
From Detail to Overview



Reality or Virtuality?

Abstractions only exist for concrete facts.

Visible Output versus Invisible Work



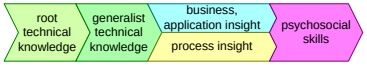
The Awakening of a System Architect

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

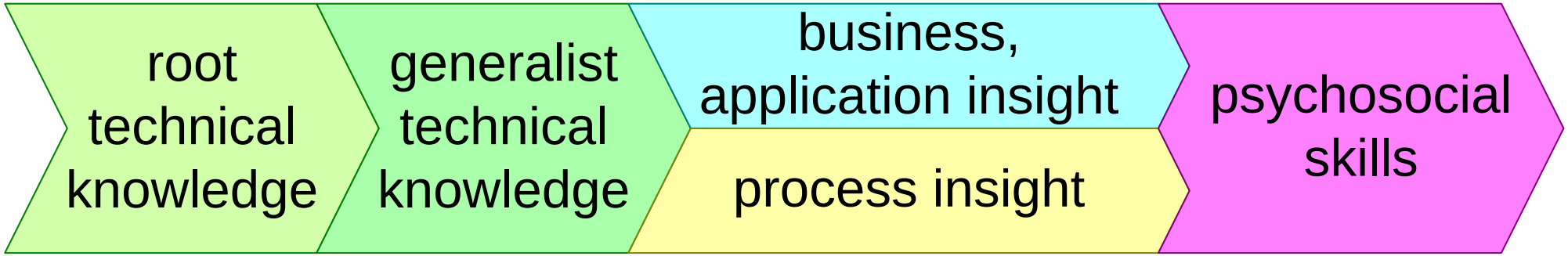
Abstract

The typical phases of a system architect development are described, beginning at the fundamental technology knowledge, with a later broadening in technology and in business aspects. Finally the subtlety of individual human beings is taken into account.

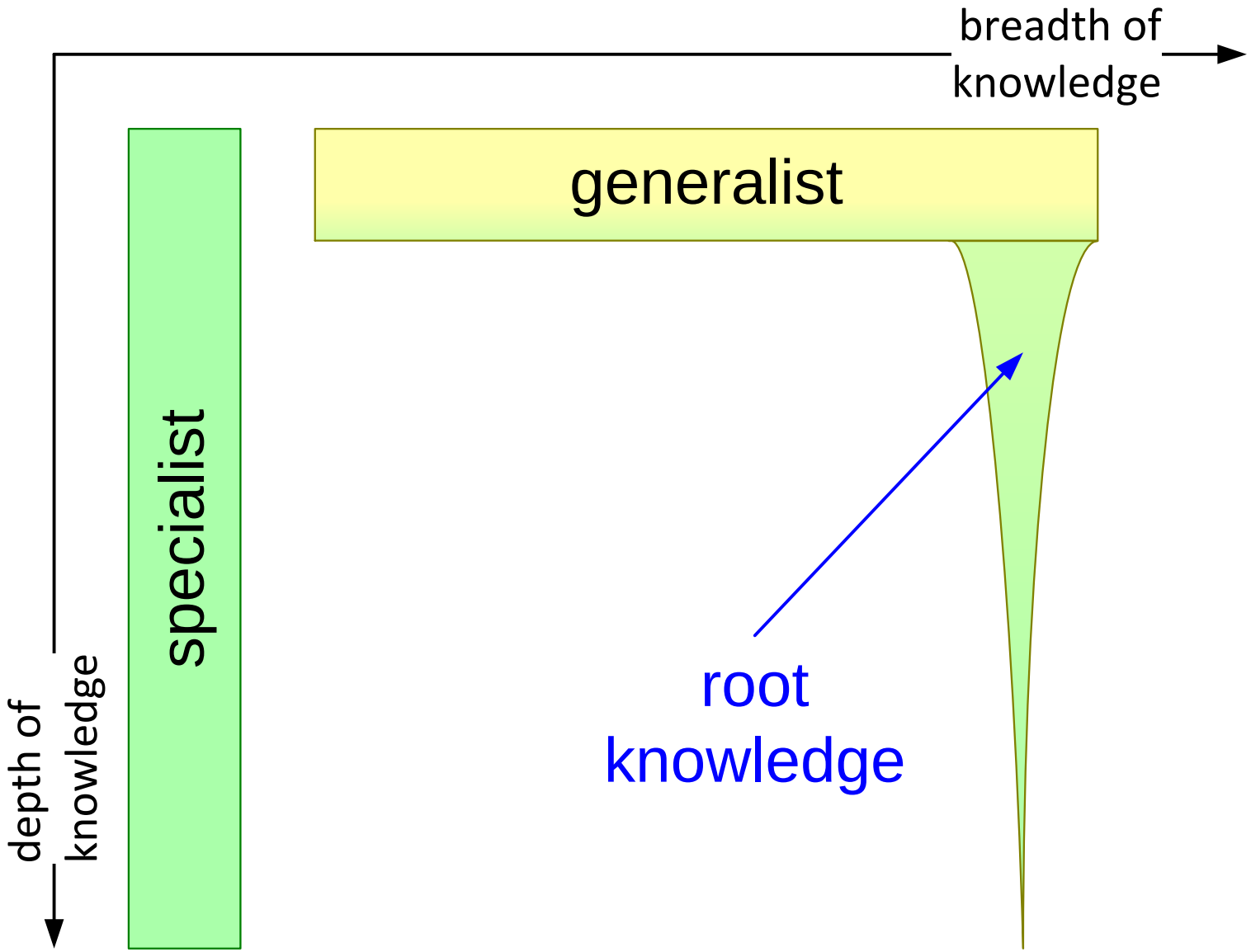
January 22, 2023
status: concept
version: 1.1



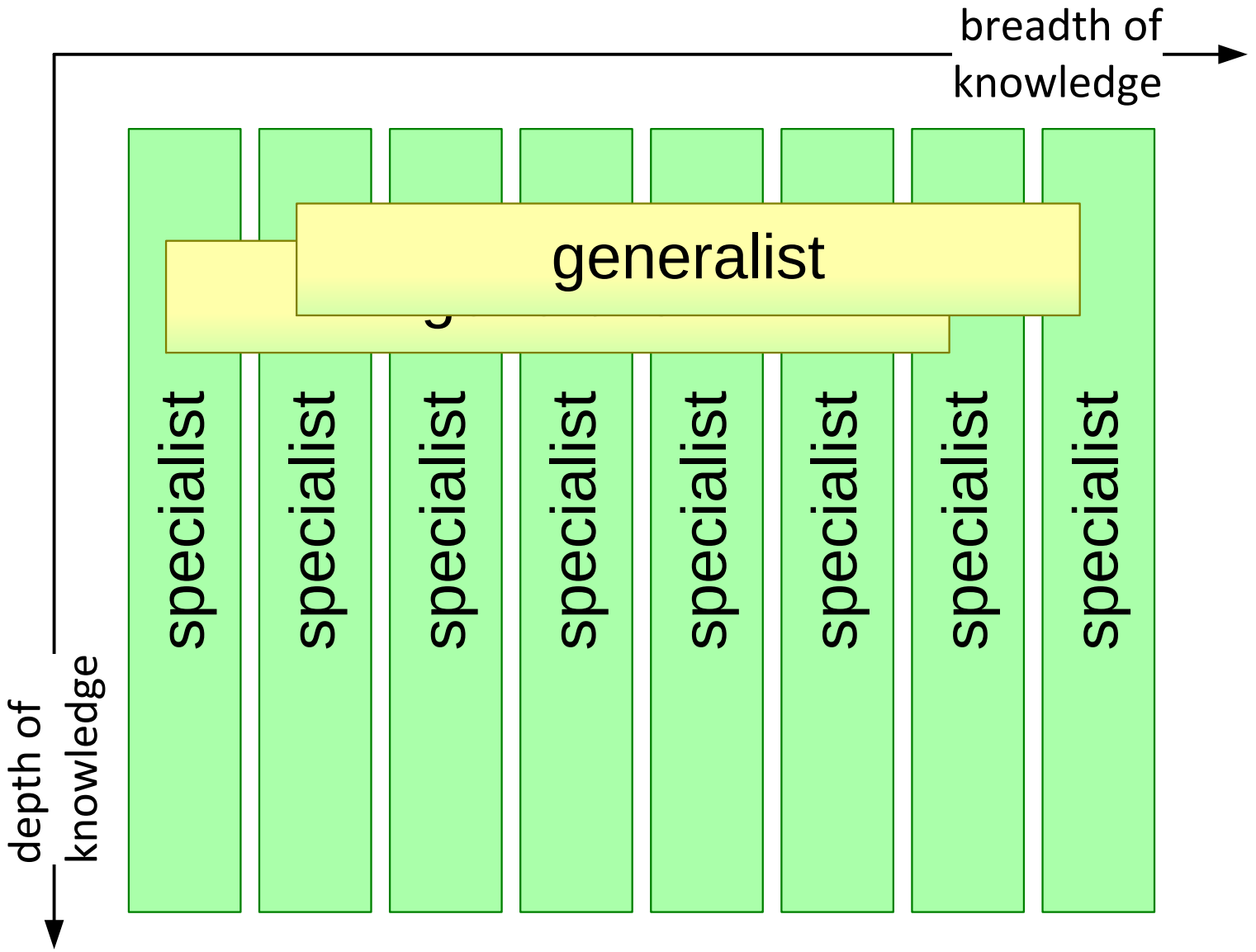
Typical Growth of a System Architect



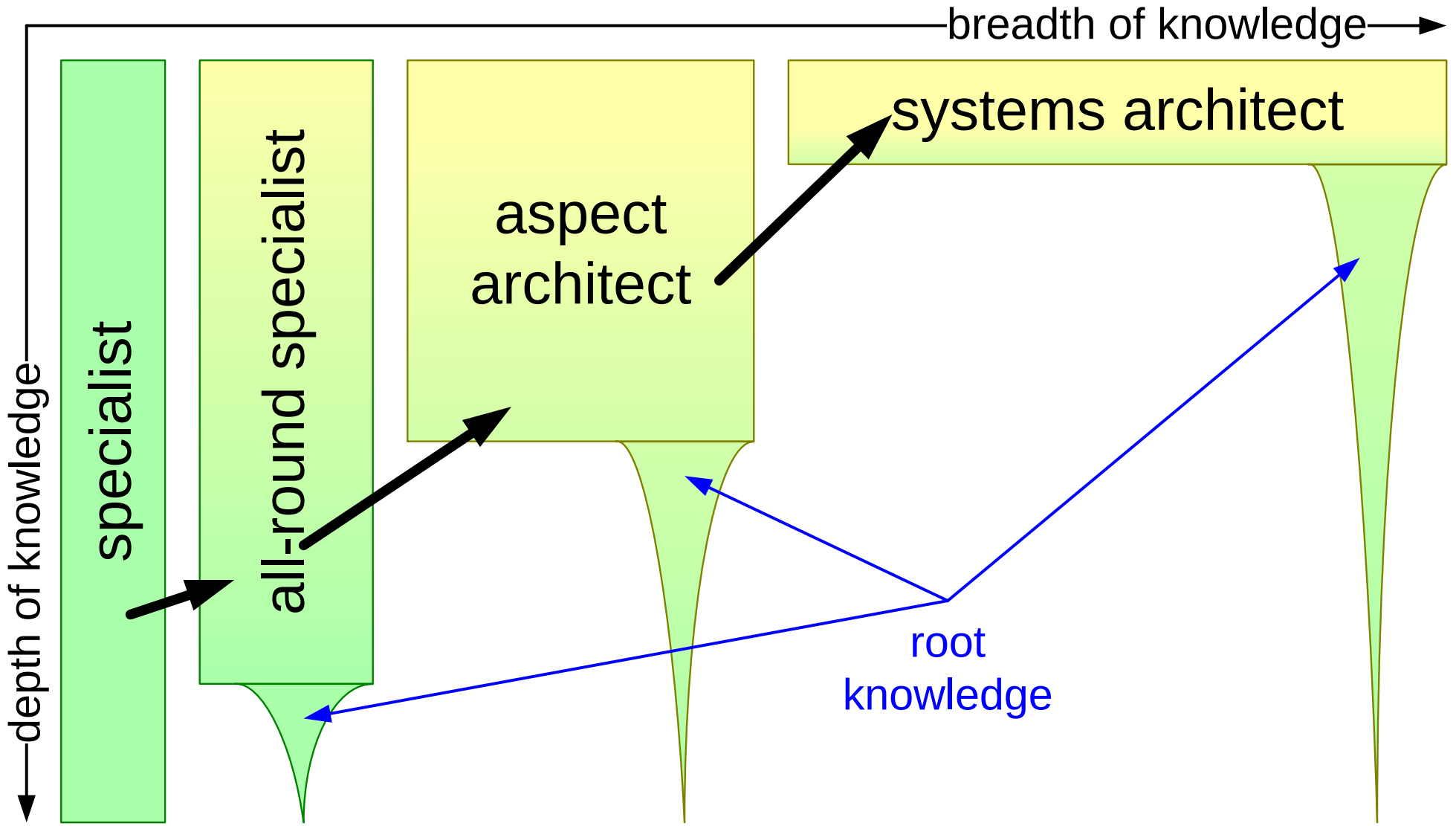
Generalist versus Specialist



Generalists and Specialists are Complementary



Spectrum from Specialist to System Architect



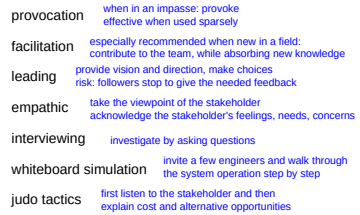
Architecting Interaction Styles

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

A system architects needs skills to apply different interactions styles, depending on the circumstances. This document discusses the following interaction styles: provocation, facilitation, leading, empathic, interviewing, white board simulation, and judo tactics.

January 22, 2023
status: draft
version: 0.2



Architecting Styles

provocation	when in an impasse: provoke effective when used sparsely
facilitation	especially recommended when new in a field: contribute to the team, while absorbing new knowledge
leading	provide vision and direction, make choices risk: followers stop to give the needed feedback
empathic	take the viewpoint of the stakeholder acknowledge the stakeholder's feelings, needs, concerns
interviewing	investigate by asking questions
whiteboard simulation	invite a few engineers and walk through the system operation step by step
judo tactics	first listen to the stakeholder and then explain cost and alternative opportunities

Exercise Role and Task of the System Architect

Role play with 3 roles and optional observer:

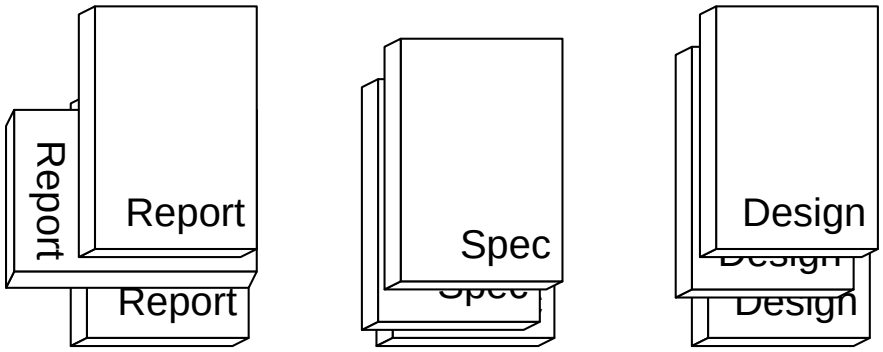
- 1 operational leader (project leader)
- 1 system architect
- 1 marketing manager
- 1 observer (optional)

Discuss the definition (business relevance, specification, and planning) of a travel e-mail mate.

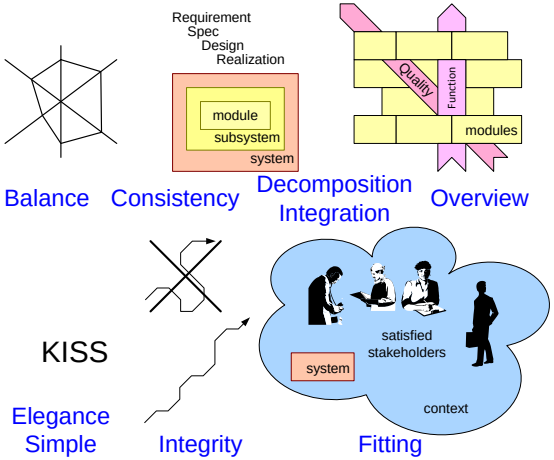
Present (max. 2 flips) the result and the process (the relation and interaction of the three roles).

Role and Task of a System Architect

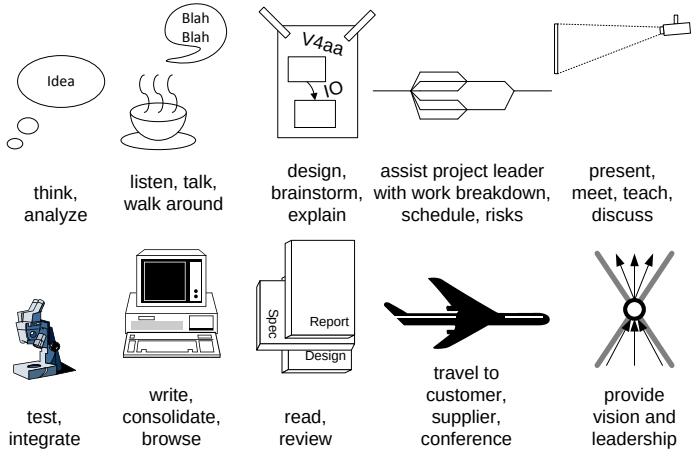
Deliverables



Responsibilities



Daily Activities

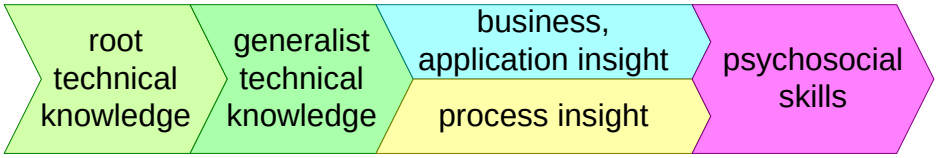


From detail to overview

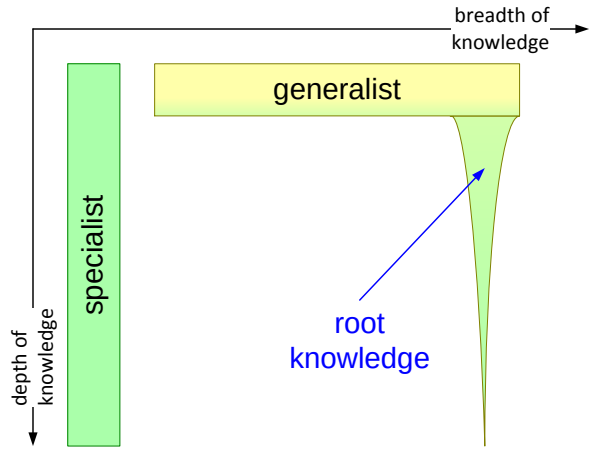
		Quantity per year (order-of-magnitude)	architect time per item
consolidation in deliverables	driving views	10	100 h
meetings	shared issues	10^2	1 h
informal contacts	touched details	10^4	0.5 – 10 min
sampling scanning	seen details	$10^5 - 10^6$	0.1 – 1 sec
	product details	$10^7 - 10^{10}$	
	real-world facts	infinite	

Personal characteristics of a System Architect

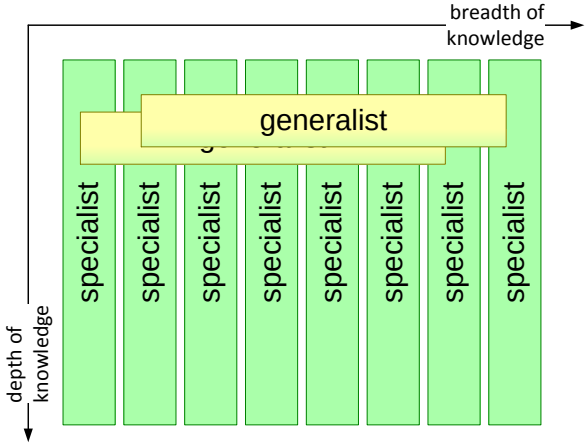
Typical growth of a Architect



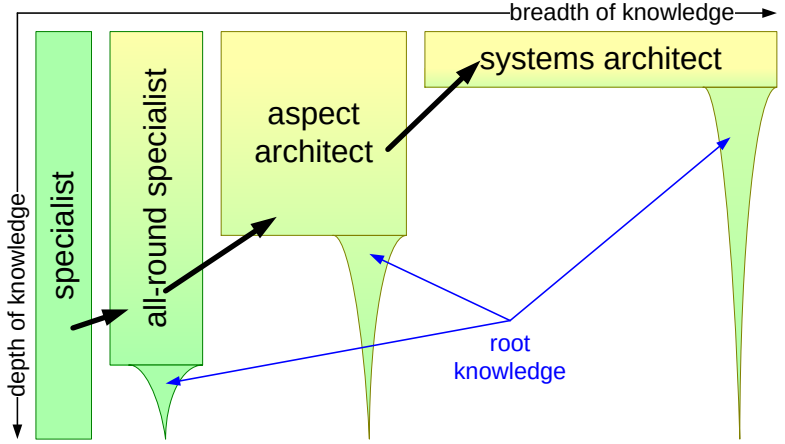
Generalist vs Specialist



Complementary Roles



Role Spectrum



Module Supporting Processes

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses supporting processes, for instance documentation, templates, and reviewing.

January 22, 2023
status: draft
version: 1.4

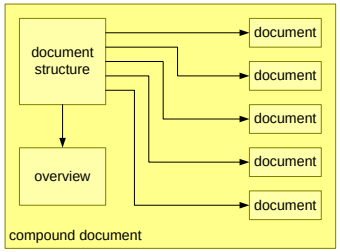
Granularity of Documentation

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

The design of documentation is discussed, with emphasis on the requirements, the need for decomposition, the measures needed to maintain overview and criteria for granularity.

January 22, 2023
status: concept
version: 1.2



Requirements for the Entire Documentation Structure

Accessibility for the readers

Low threshold for the readers

Low threshold for the authors

Completeness

Consistency

Maintainability

Scalability

Evolvability

Process to ensure the quality of the information

Requirements from Reader Point of View

Convenient

easy

fast

viewing

printing

searching

Requirements per Document

High cohesion (within the unit)

Low coupling (outside of the unit)

Accessibility for the readers

Low threshold for the reader

Low threshold for the author

Manageable steps to create, review, and change

Clear responsibilities

Clear position and relation with the context

Well-defined status of the information

Timely availability

Accessibility Requirements

Ease of reading, “juiciness”

High signal-to-noise ratio: information should not be hidden in a sea of words.

Understandability

Reachability in different ways, e.g., by hierarchical or full search

Reachability in a limited number of steps

Responsibility Requirements

single author

limited amount of reviewers

Scalability Requirements

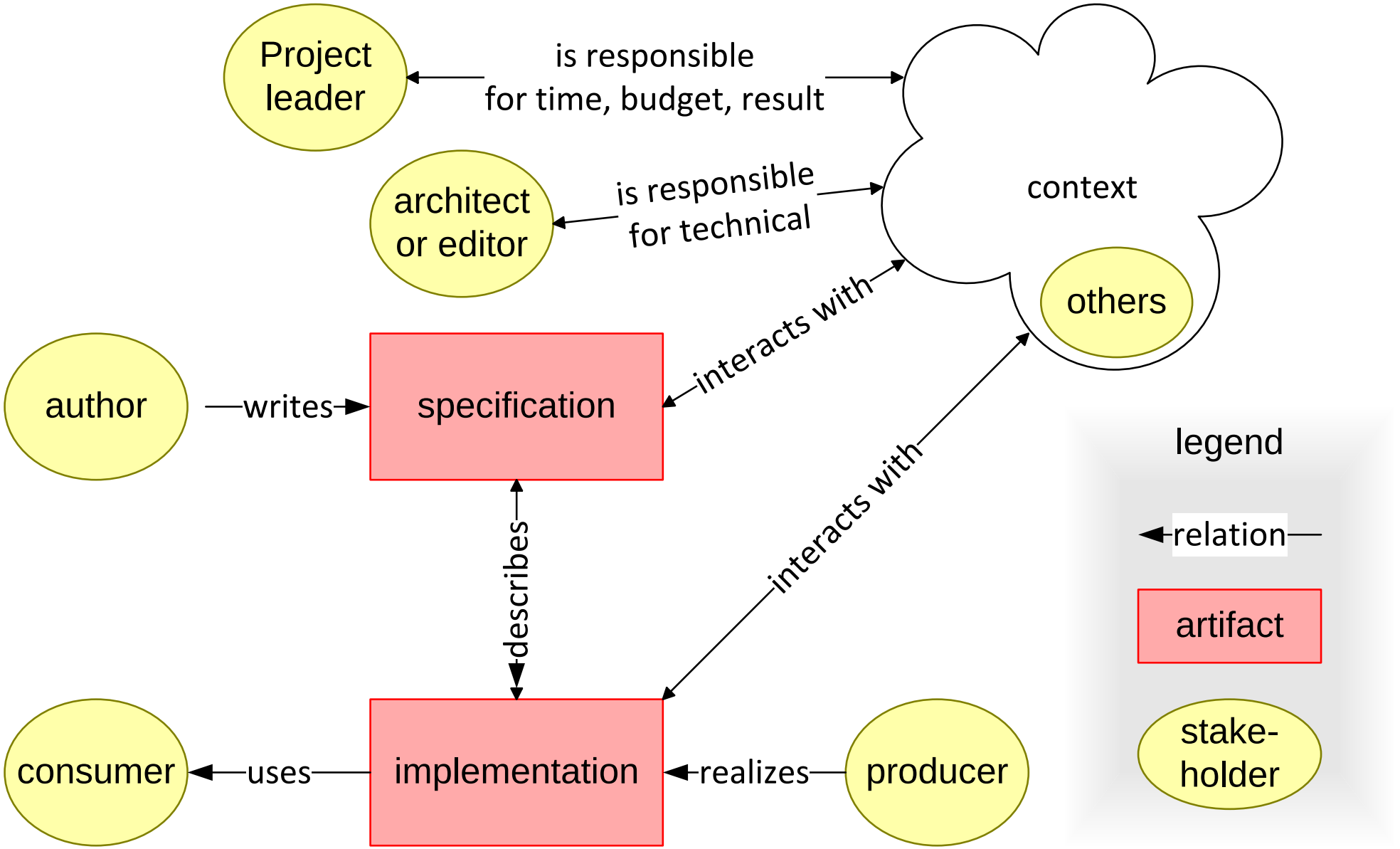
well defined documentation structure

overview specifications at higher aggregation levels

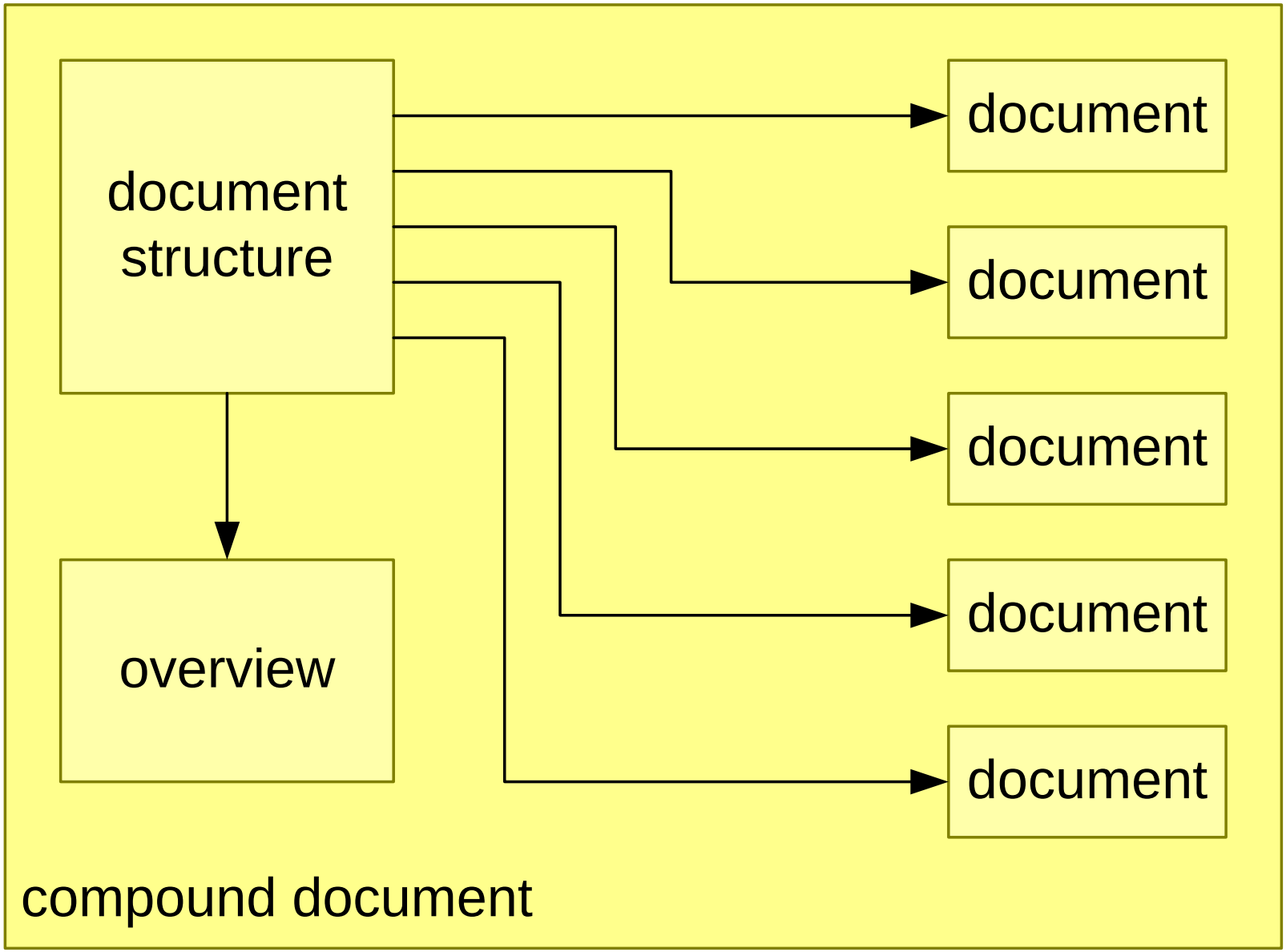
recursive application of structure and overview

delegation of review process

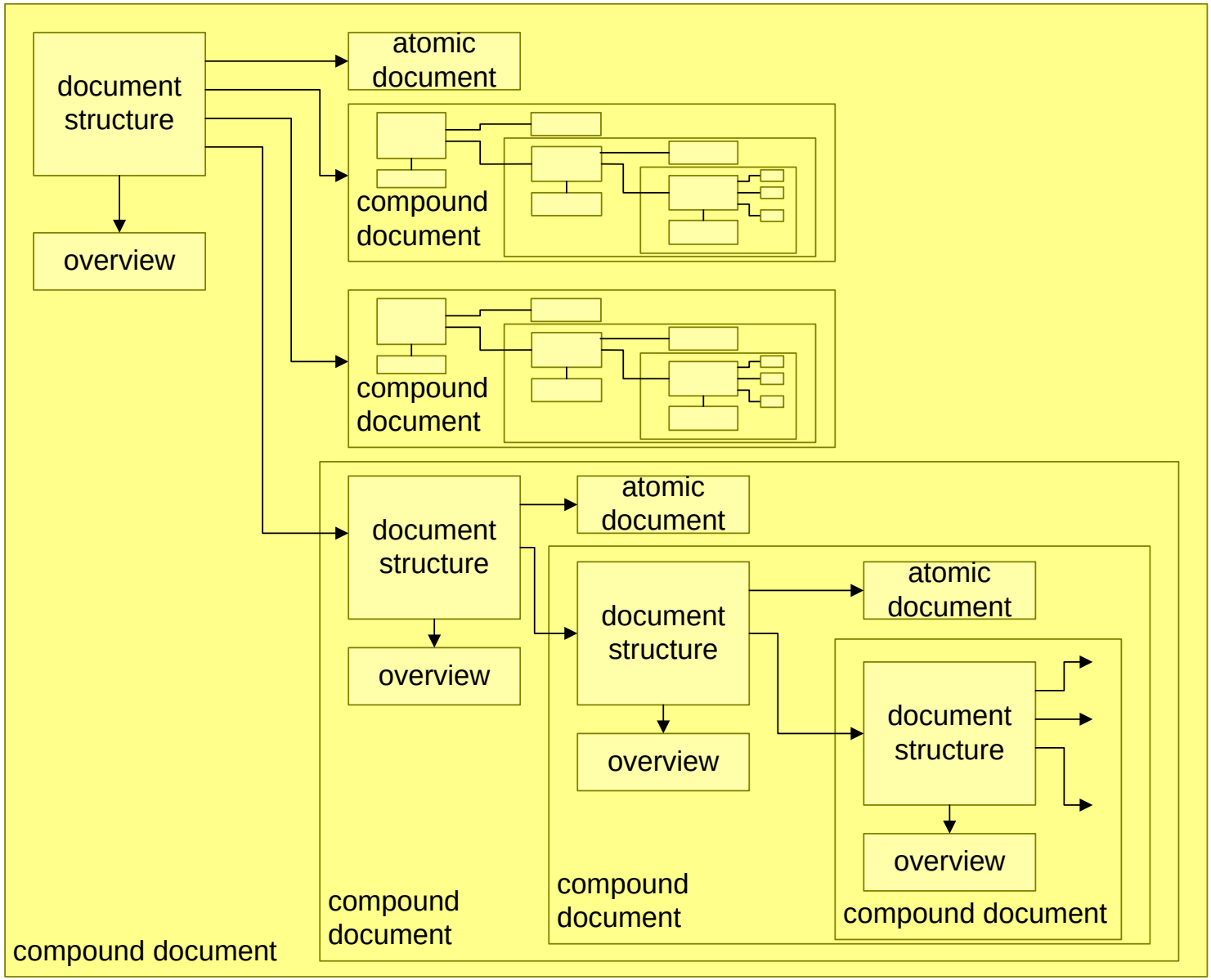
The Stakeholders of a Single Document



Decomposition of Large Documents

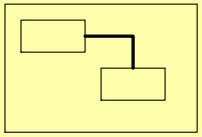
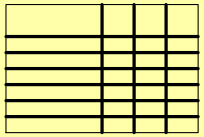


Documentation Tree by Recursive Decomposition



Payload: the Ratio between Content and Overhead

front page

title identification author distribution status review	history changes	 diagrams  tables	1. aap 2. noot 3. mies lists and ca 50% text

meta information
max 2 pages

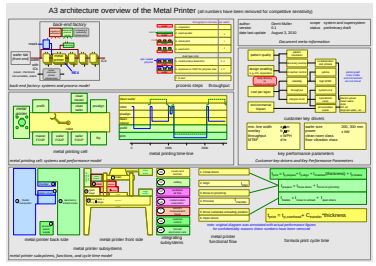
contents
2..18 pages

LEAN and A3 Approach to Supporting Processes

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

LEAN product development is in the process and means area pragmatic. Low tech tools, such as paper, pen and magnets, with very direct interaction are used. For communication the use of single A3-size documents is promoted, because this is a manageable amount of information.



January 22, 2023
status: draft
version: 0.1

Characteristics of LEAN

A holistic, systems approach to product development including people, processes, and technology.

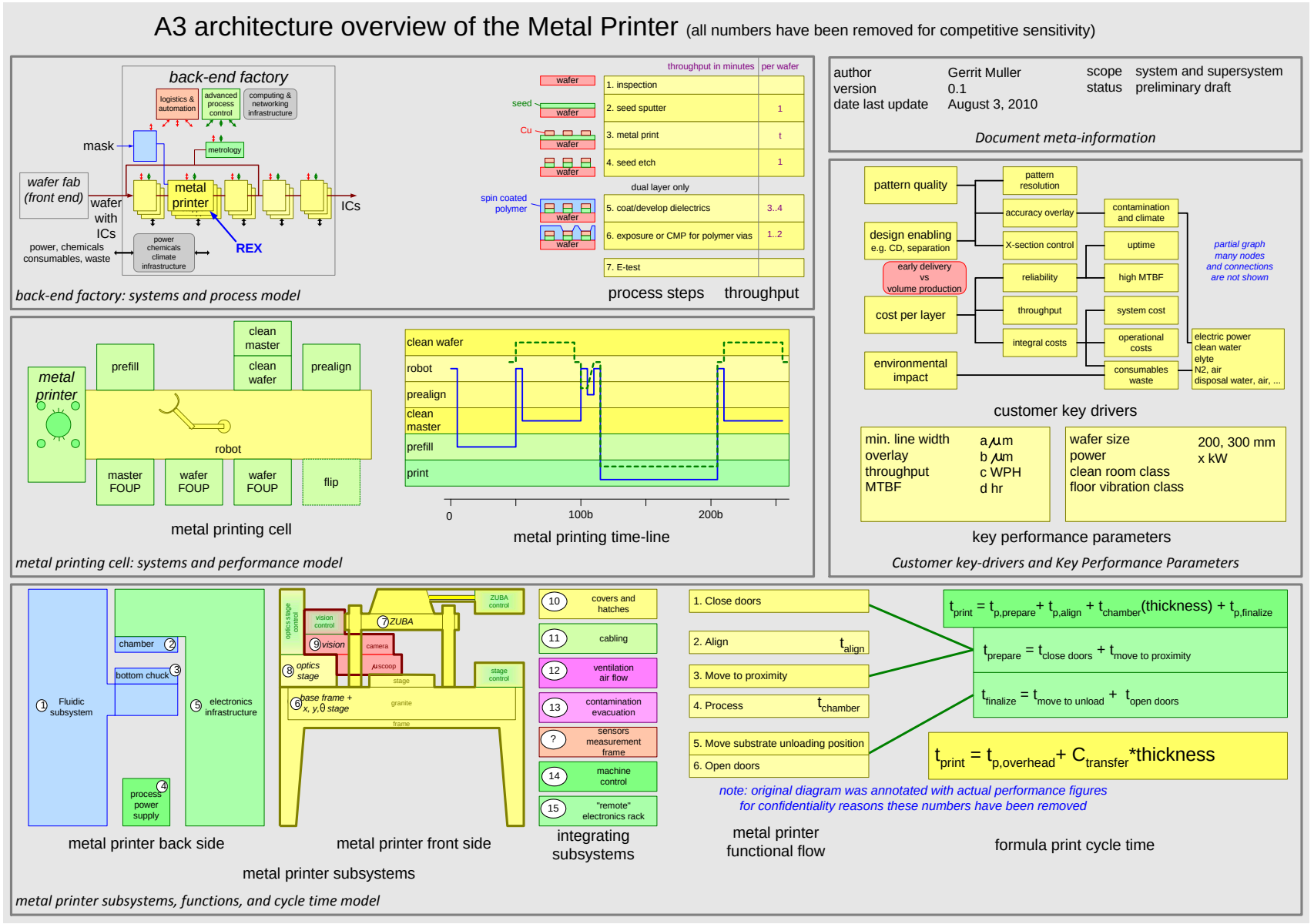
Multi-disciplinary from the early start, with a drive to be fact based.

Customer understanding as the the starting point.

Continuous improvement and learning as cultural value.

Small distance between engineers and real systems, including manufacturing, sales and service and the system of interest.

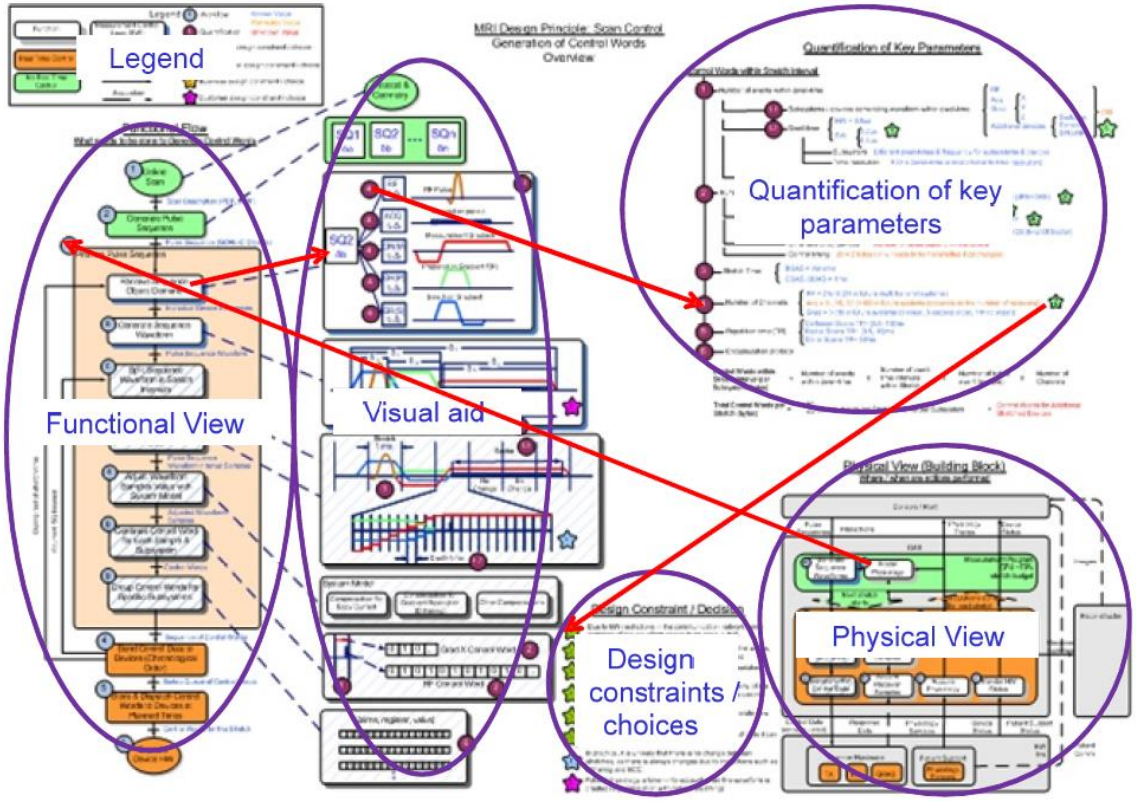
Example of A3 Architecture Overview



multiple related views

quantifications

one topic
per A3



source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/

capture
"hot" topics

digestable
(size limitation)

practical
close to stakeholder experience

Light Weight Review Process

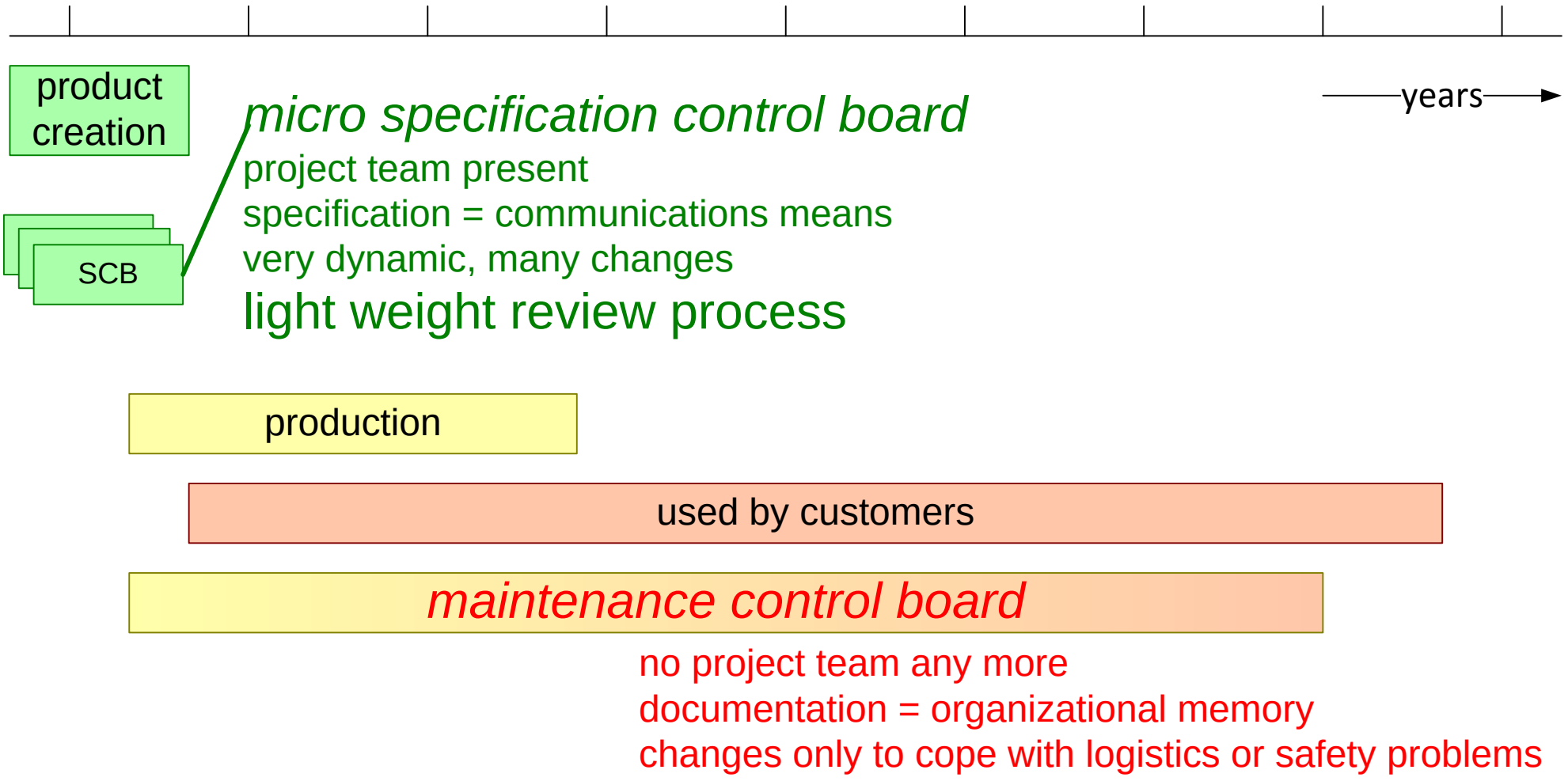
by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

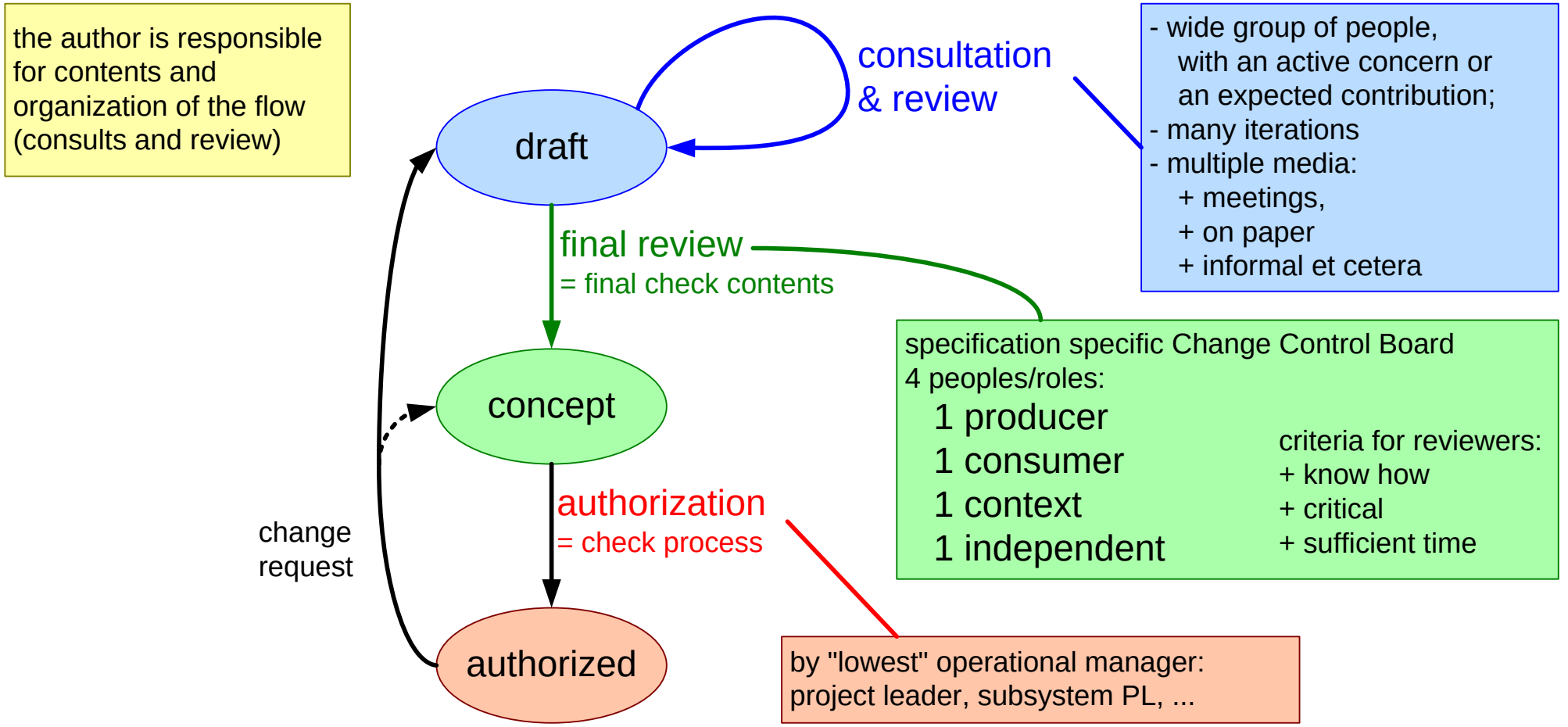
A light weight review process is described that can be used for documents made during product creation. This review process is focused on improving the contents of specifications as early as possible. The process is light weight to increase the likelihood that it is performed *de facto* instead of *pro forma*.

January 22, 2023
status: preliminary draft
version: 0

Product Life Cycle and Change Management



Light Weight Specification Review Process

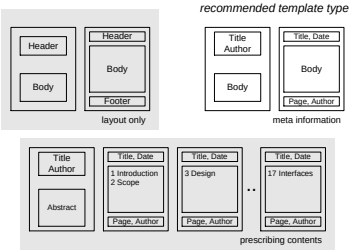


Template How To

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

The introduction of a new process (way of working) is quite often implemented by supplying ready-to-go tools and templates. This implementation mainly serves the purpose of a smooth introduction of the new process.
Unfortunately the benefits of templates are often cancelled by unforeseen side-effects, such as unintended application, inflexibility, and so on. This intermezzo gives hints to avoid the **Template Trap**, so that templates can be used more effectively to support introduction of new processes.



January 22, 2023
status: draft
version: 1.6

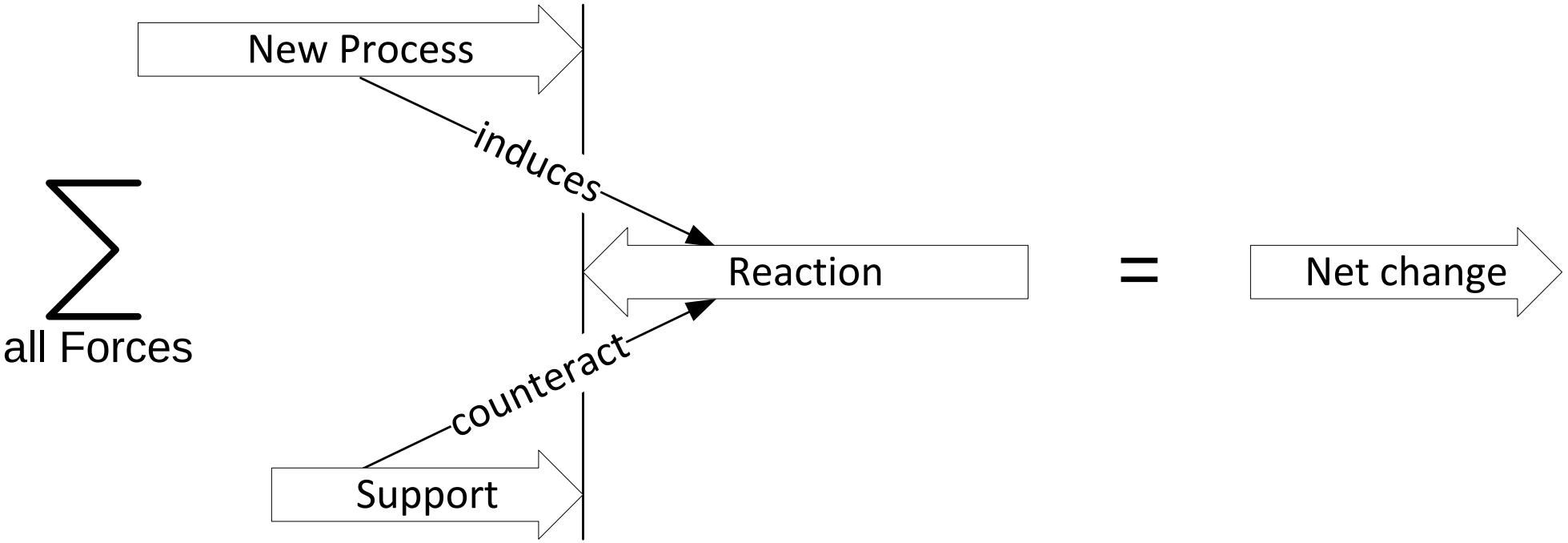
Rationale for Templates

- Low threshold to apply a (new) process (1)
- Low effort to apply a (new) process (2)
- No need to know low level implementation details (3)
- Means to consolidate and reuse experiences (4)

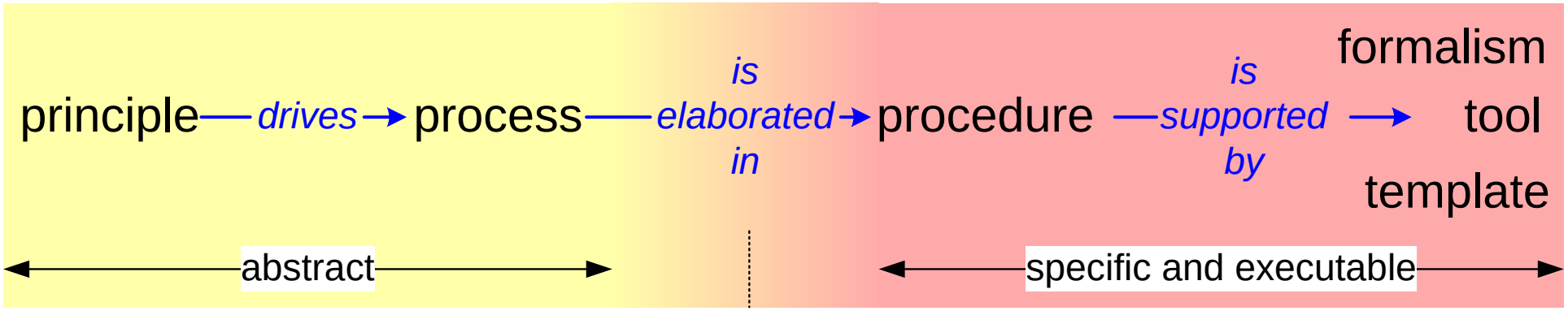
Bogus Arguments for Templates

- Obtain a uniform look (5)
- Force the application of a (new) process (6)
- Control the way a new process is applied (7)

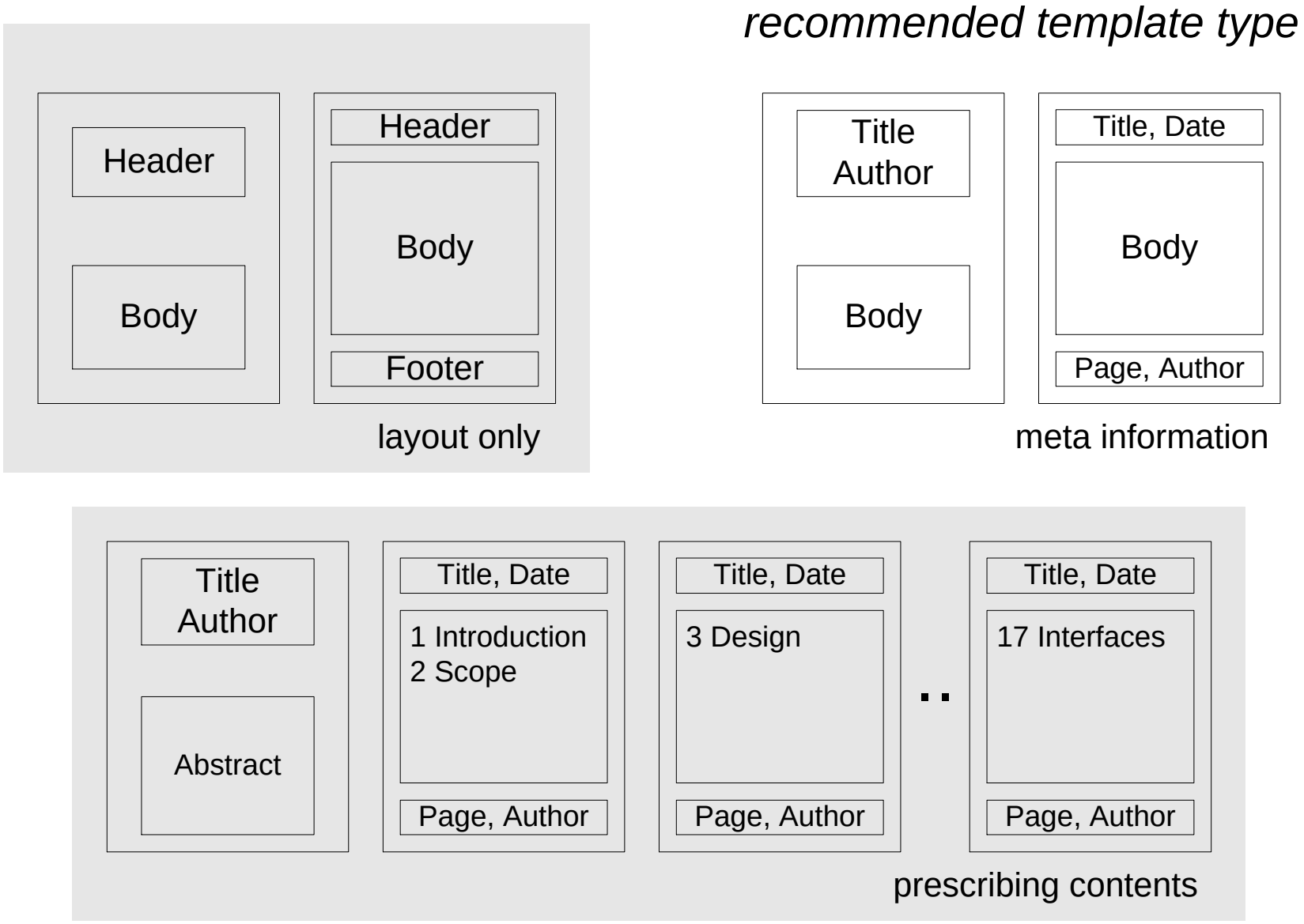
Forces of Change: Action = - Reaction



Template as Support for Process



Types of Templates



Recommendation

template type	context knowhow	value
layout only	no	low
meta information	process	high
prescribing content	process and domain	constraining

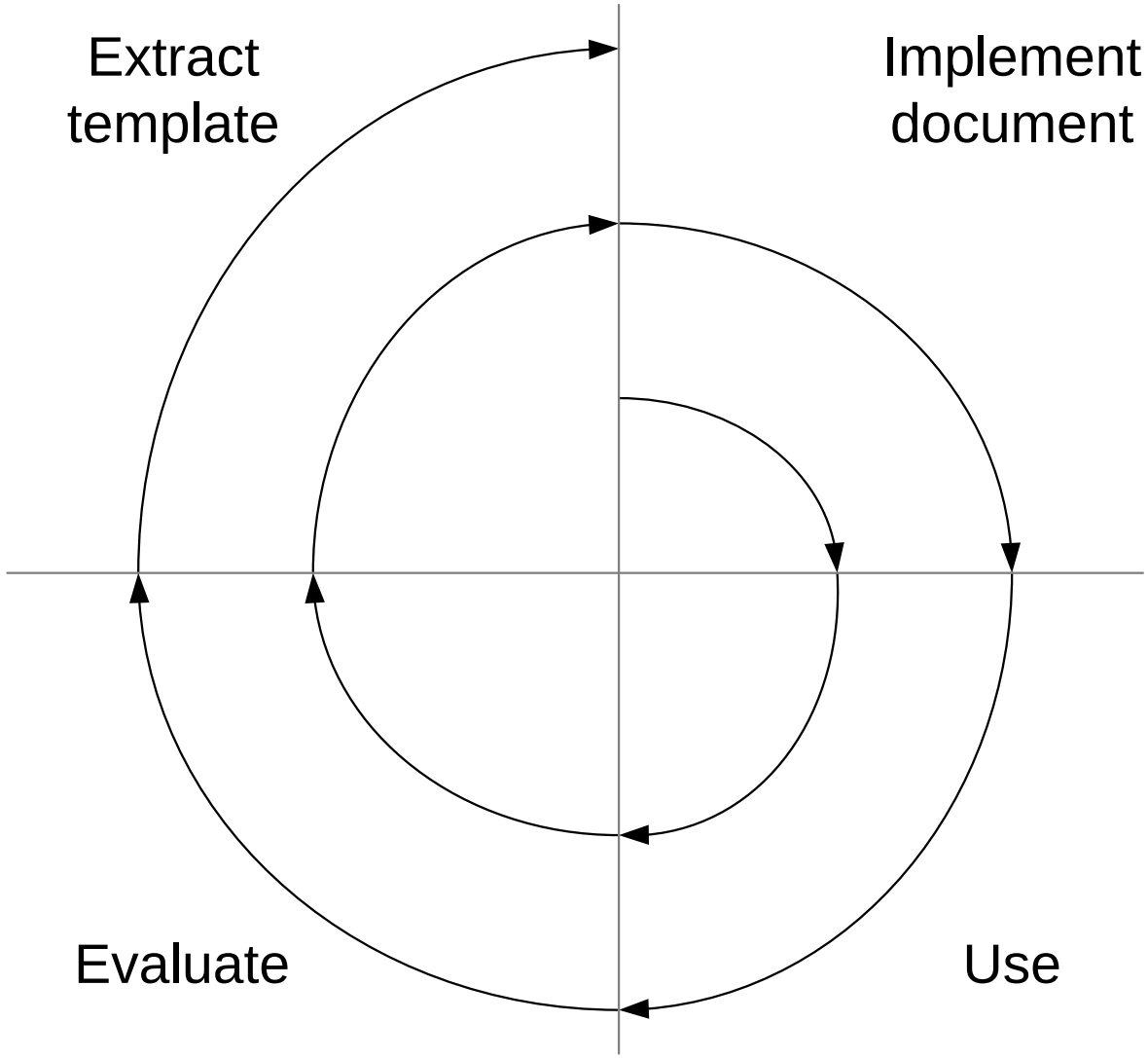
- Use templates for meta-information.
- Use checklists for structure and contents.

Template Development

Templates are an optimization of the Copy Paste Modify pattern:

- Look for a similar problem
- Copy its implementation
- Modify the copy to fulfil the new requirements

Spiral model: Use before Re-use



Example Guidelines Meta Information(1)

Mandatory per page:

- Author
- Title
- Status
- Version
- Date of last update
- Unique Identification
- Business Unit
- Page number

Example Guidelines Meta Information(2)

Mandatory per document:

- Distribution (Notification) list
- Reviewers and commentators
- Document scope (Product family, Product, Subsystem, Module as far as applicable)
- Change history

Example Guidelines Meta Information(3)

Recommended Practice:

- Short statement on frontpage stating what is expected from the addressed recipients, for example:
  - Please send comments before february 29, this document will be reviewed on that date
  - This document is authorized, changes are only applied via a change request
- See Granularity of Documentation [**?**] for guidelines for modularization and contents

Template Pitfalls

- Author follows template instead of considering the purpose of the document.
- Template is too complex.
- There is an unmanageable number of variants.
- Mandatory use of templates results in:
  - no innovation of templates (= no learning)
  - no common sense in deployment
  - strong dependency on templates

Recommendation:

- Enforce the procedure *(what)*
- Provide the template *(how)* as supporting means.

Summary

- Templates support (new) processes
- Use templates for layout and meta information support
- Do not use templates for documents structure or contents
- Stimulate evolution of templates, keep them alive
- Keep templates simple
- Standardize on **what** (process or procedure), not on **how** (tool and template)
- Provide (mandatory) guidelines and recommended practices
- Provide templates as a supportive choice, don't force people to use templates

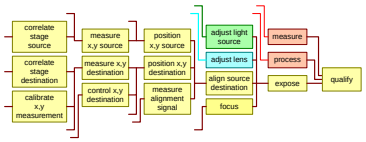
System Integration How-To

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

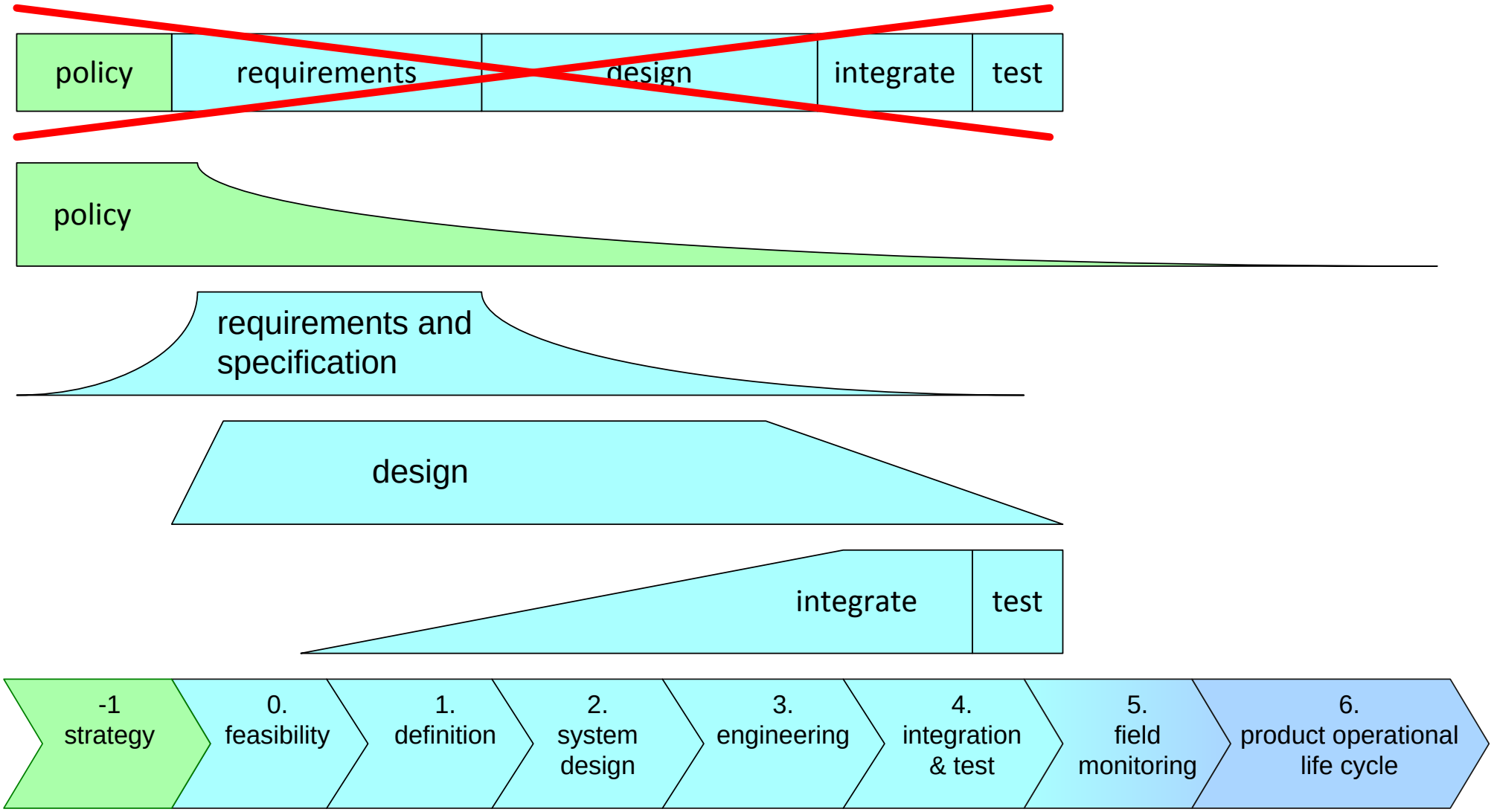
Abstract

In this document we will discuss the full integration flow. We will discuss the goal of integration, the relation between integration and testing, what is integration and how to integrate, an approach to integration, scheduling and dealing with disruptive events, roles and responsibilities, configuration management aspects, and typical order of integration problems occurring in real life.

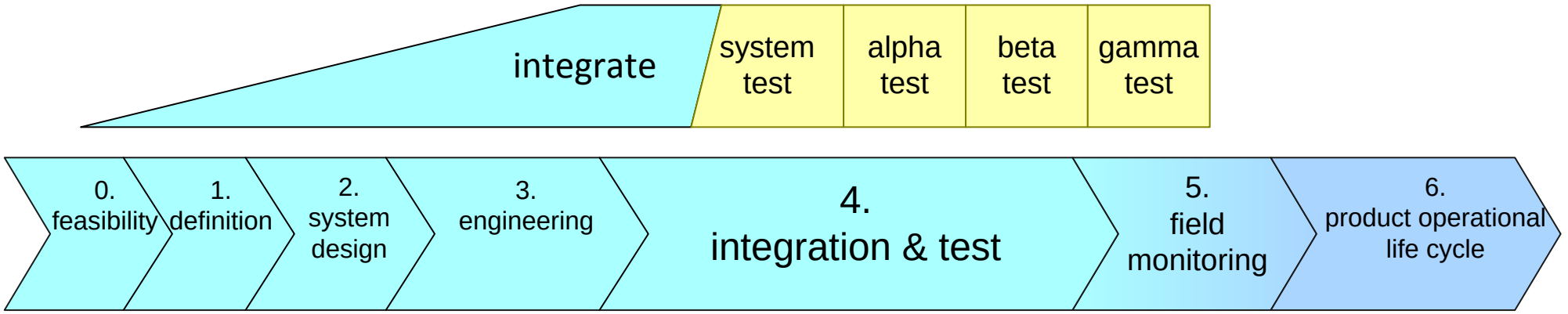
January 22, 2023
status: concept
version: 0.2



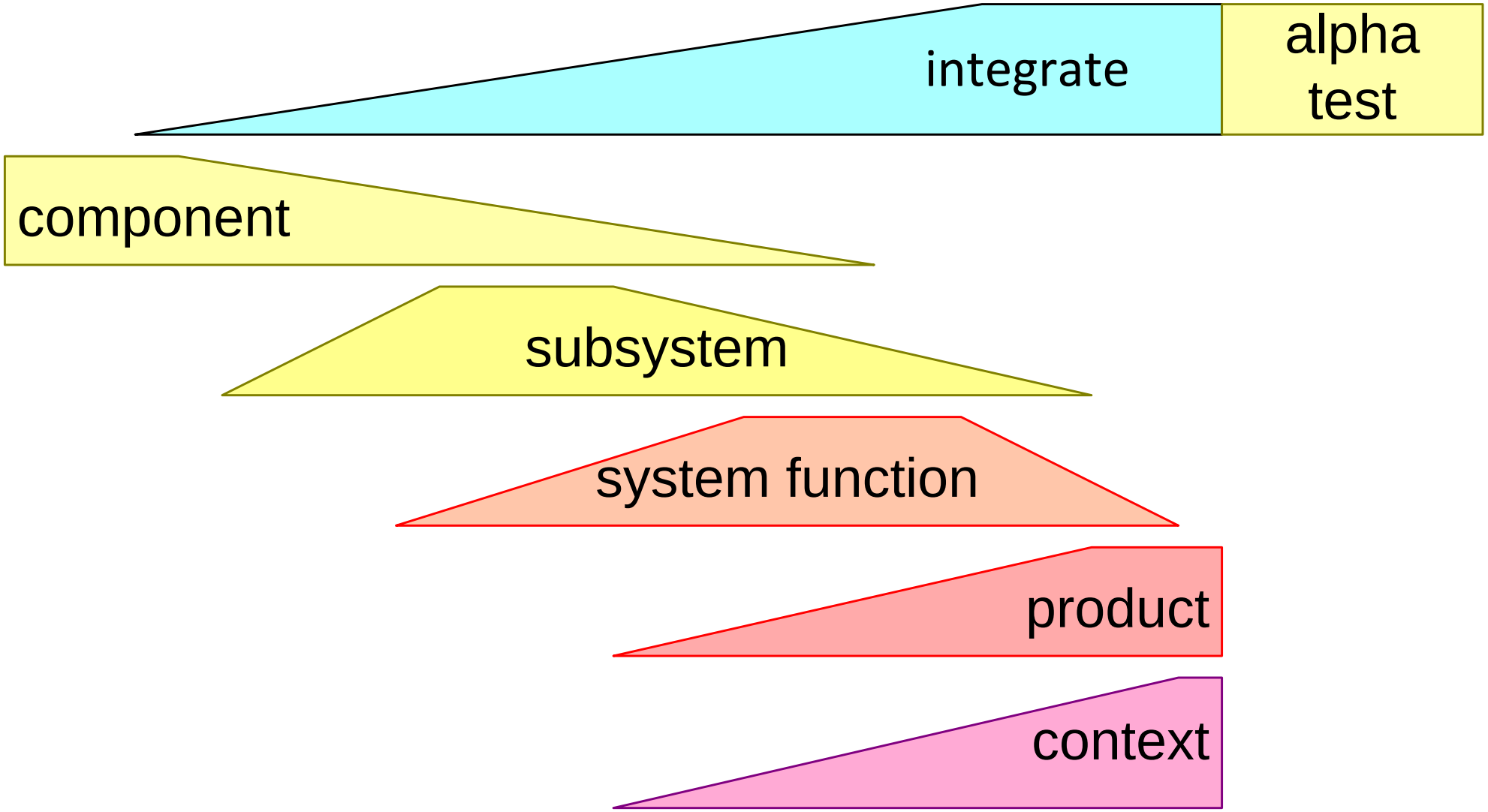
Typical Concurrent Product Creation Process



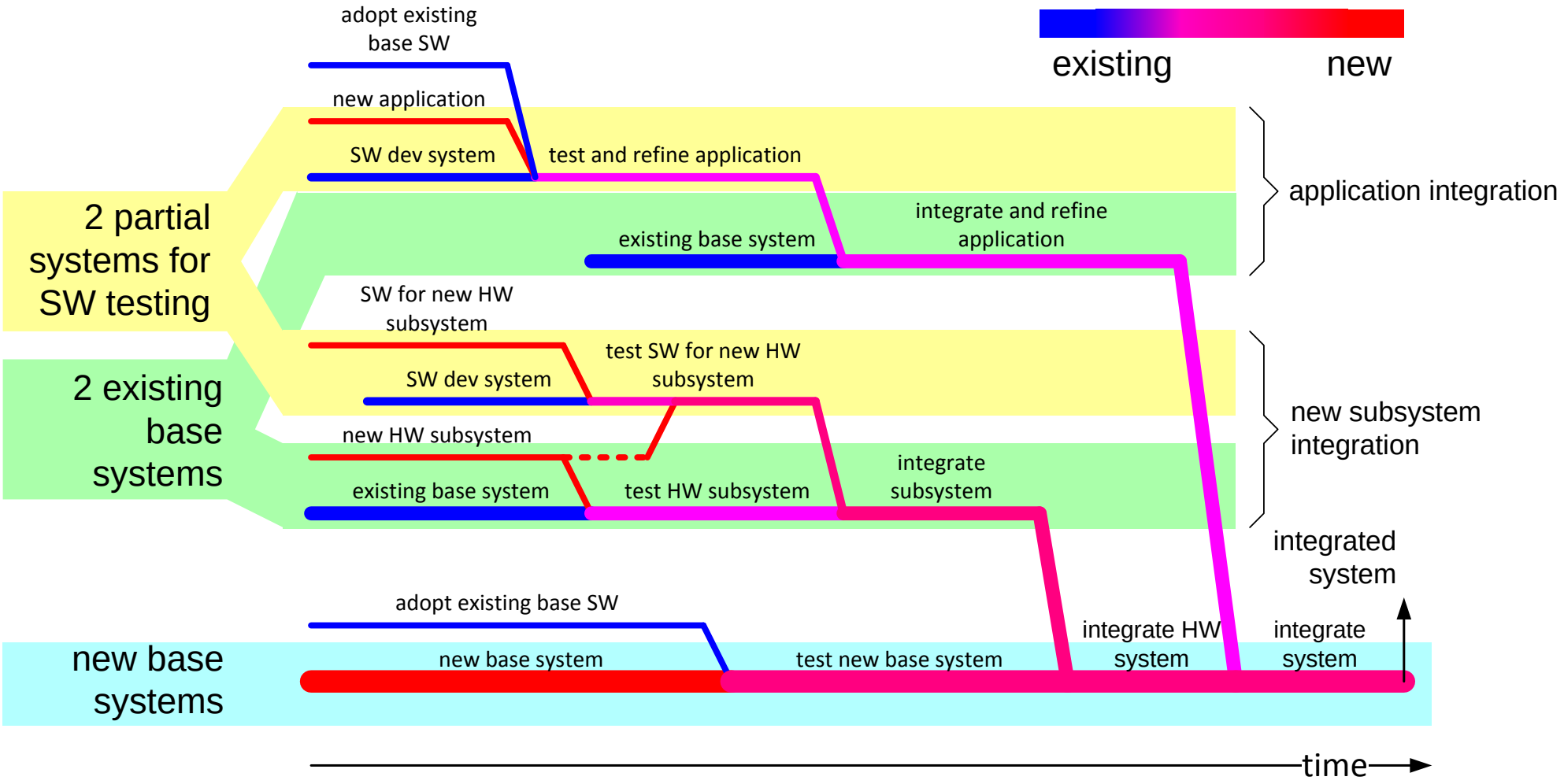
Zooming in on Integration and Tests



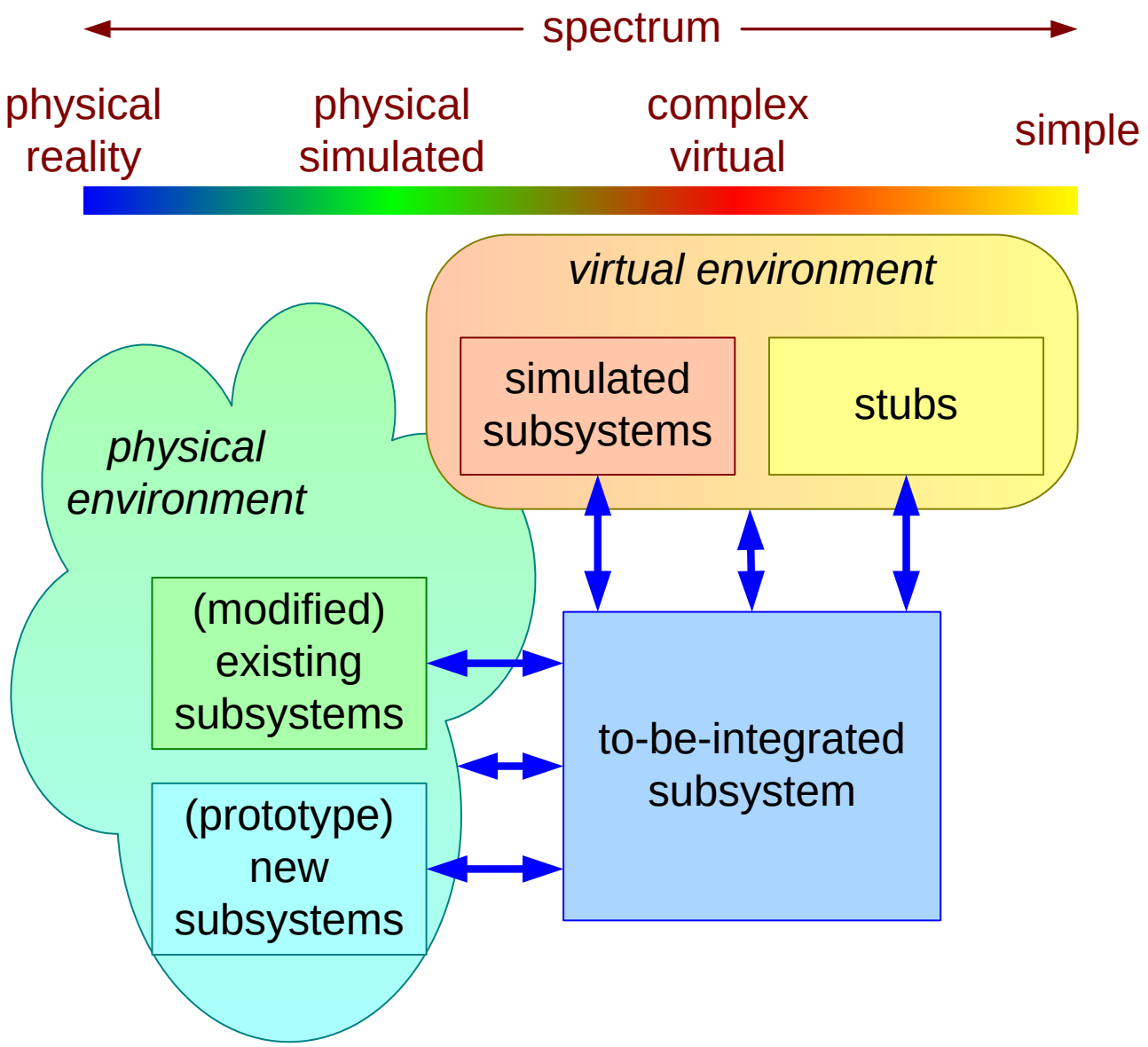
Integration Takes Place in a Bottom-up Fashion



Transition from Previous System to New System



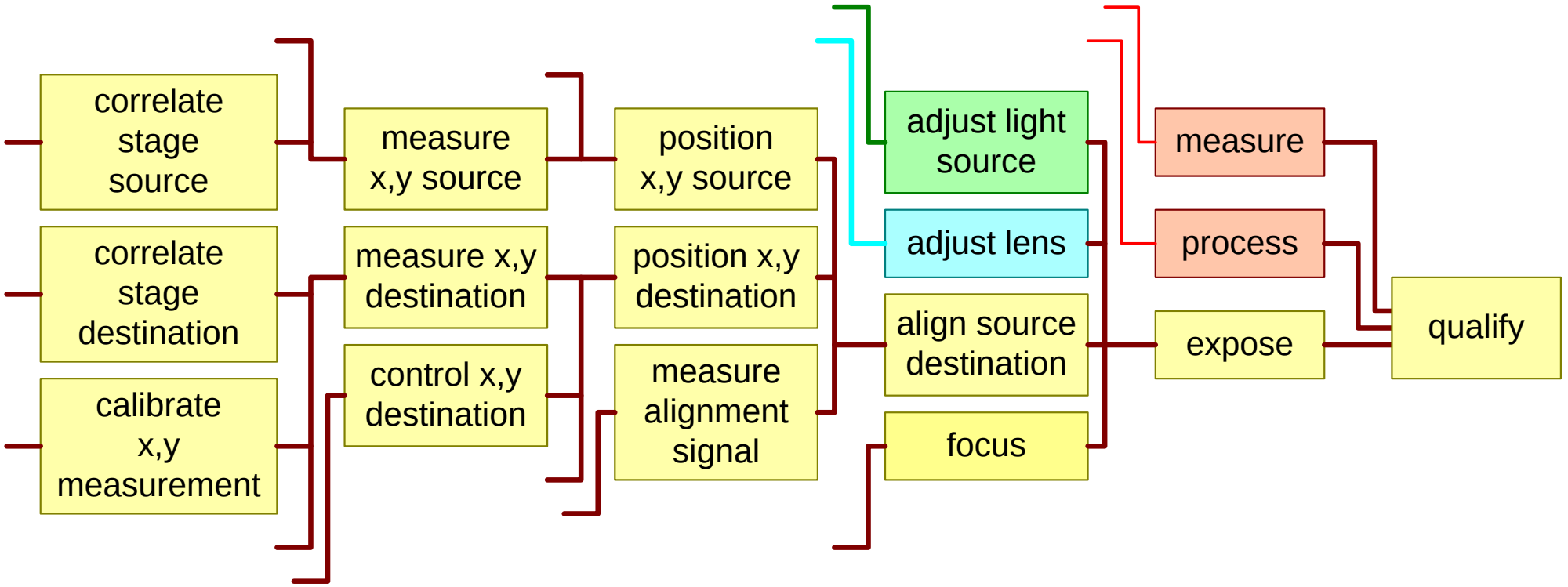
Alternatives to Integrate a Subsystem Early in the Project



Stepwise Integration Approach

1	Determine most critical system performance parameters.
2	Identify subsystems and functions involved in these parameters.
3	Work towards integration configurations along these chains of subsystems and functions.
4	Show system performance parameter as early as possible; start with showing "typical" system performance.
5	Show "worst-case" and "boundary" system performance.
6	Rework manual integration tests in steps into automated regression tests.
7	Monitor regression results with human-driven analysis.
8	Integrate the chains: show system performance of different parameters simultaneously on the same system.

Order of Functions Required for the IQ of a Waferstepper



Roles and Responsibilities During the Integration Process

project leader

organization
resources
schedule
budget

systems architect/ engineer/integrator

system requirements
design inputs
test specification
schedule rationale
troubleshooting
participate in test

system tester

test
troubleshooting
report

logistics and administrative support

configuration
orders
administration

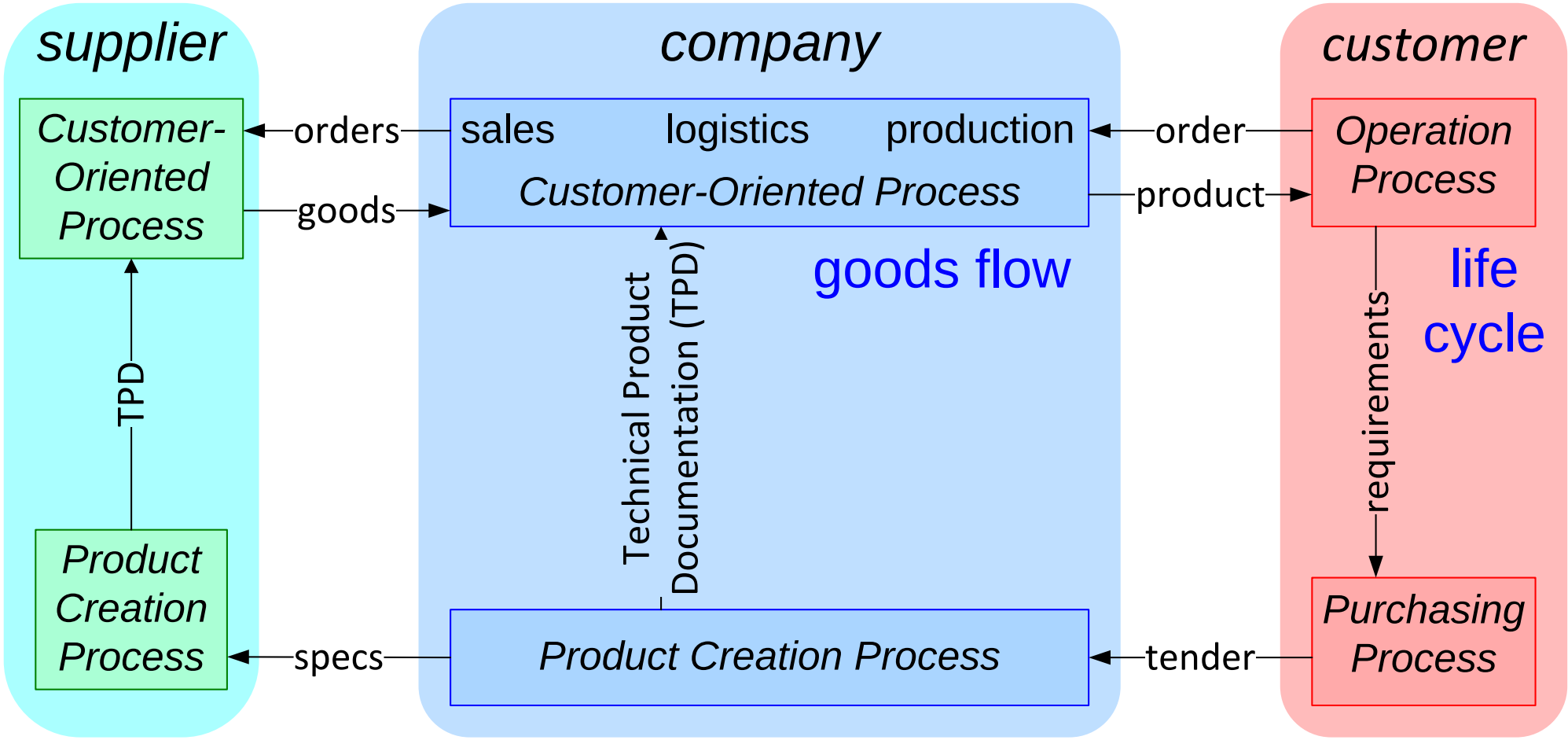
engineers

design
component test
troubleshooting
participate in test

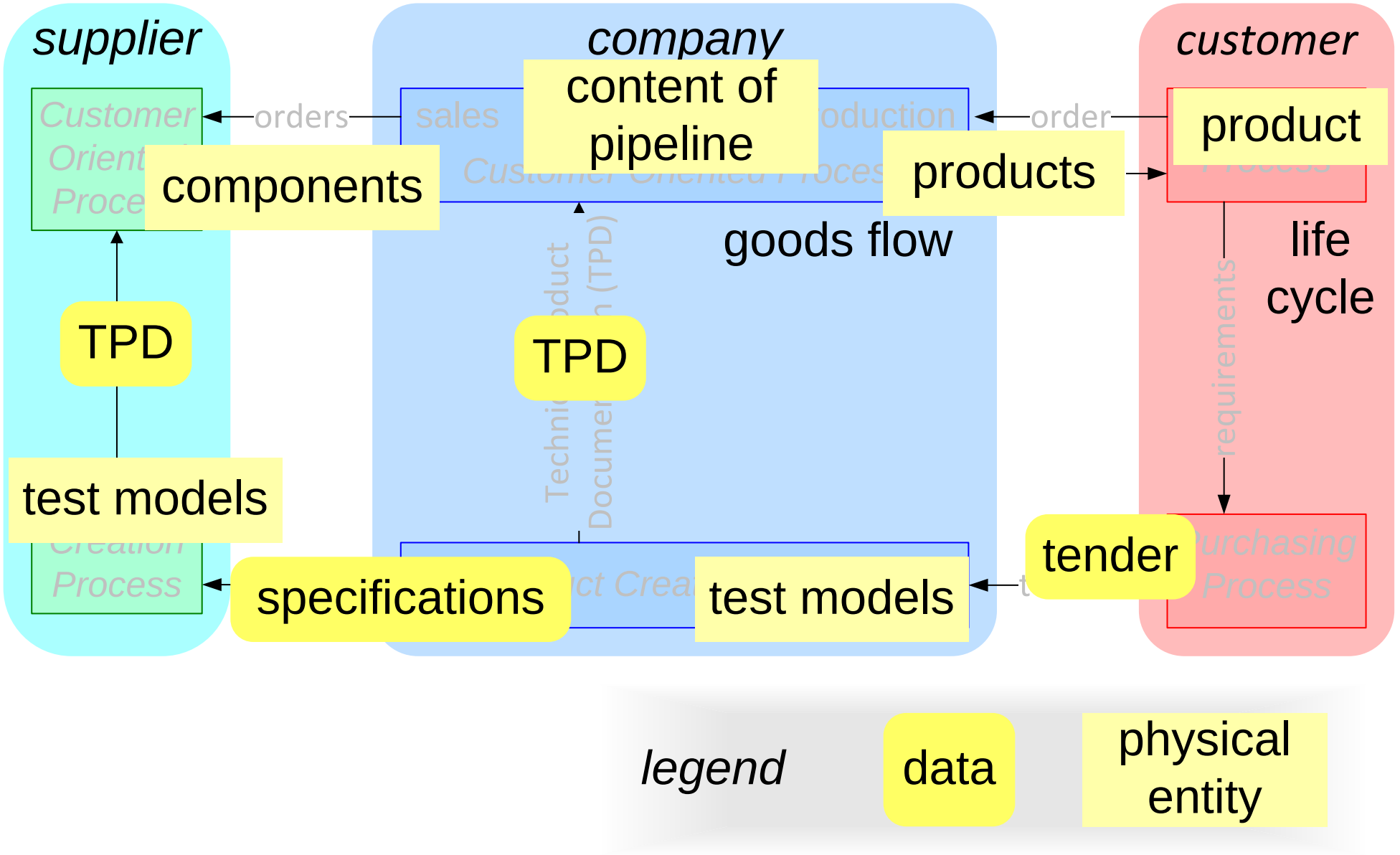
machine owner

maintain test model
support test

Simplified Process Diagram



Configuration Management Entities



Typical Order of Integration Problems

1. The (sub)system does not build.
2. The (sub)system does not function.
3. Interface errors.
4. The (sub)system is too slow.
5. Problems with the main performance parameter, such as image quality.
6. The (sub)system is not reliable.

Exercise Documentation

Make a design for the documentation structure of the case, take into account a.o.:

- target audience per documentation module
- lifecycle
- author
- size (budget)

Present (max 1 flip) the proposed documentation structure and the rationale.

Documentation

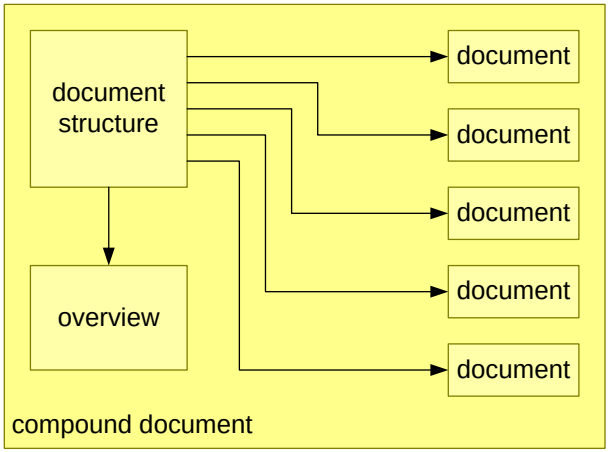
Requirements Entire Documentation

- Accessibility for the readers
- Low threshold for the readers
- Low threshold for the authors
- Completeness
- Consistency
- Maintainability
- Scalability
- Evolvability
- Process to ensure the quality of the information

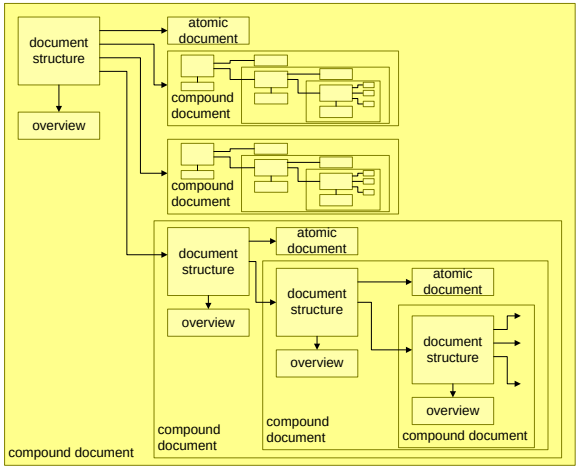
Requirements per Document

- High cohesion (within the unit)
- Low coupling (outside of the unit)
- Accessibility for the readers
- Low threshold for the reader
- Low threshold for the author
- Manageable steps to create, review, and change
- Clear responsibilities
- Clear position and relation with the context
- Well-defined status of the information
- Timely availability

Decompose Large Documents



Recursive Decomposition



Documentation

Maximize Payload

front page

title identification author distribution status review	history changes	diagrams tables	1. aap 2. noot 3. mies lists and ca 50% text

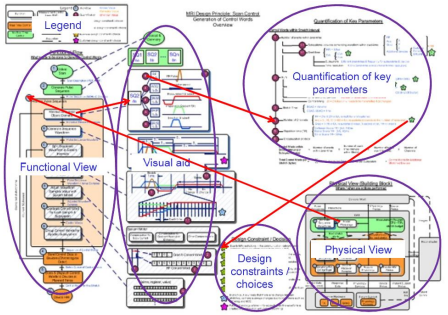
meta information max 2 pages ← → contents 2..18 pages

A3s

multiple related views

quantifications

one topic per A3



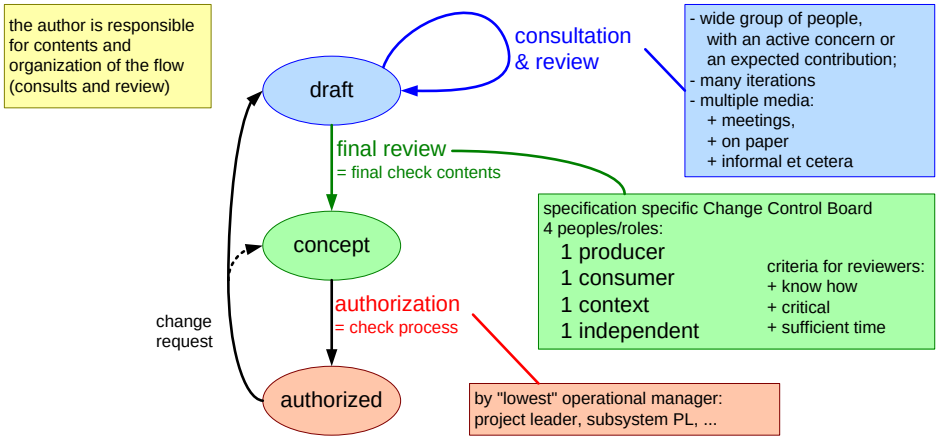
source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/

capture "hot" topics

digestable (size limitation)

practical close to stakeholder experience

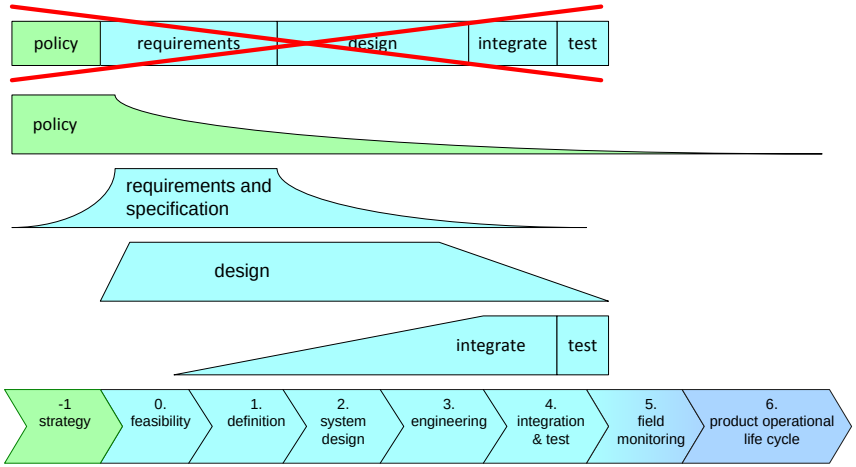
Light Weight Review



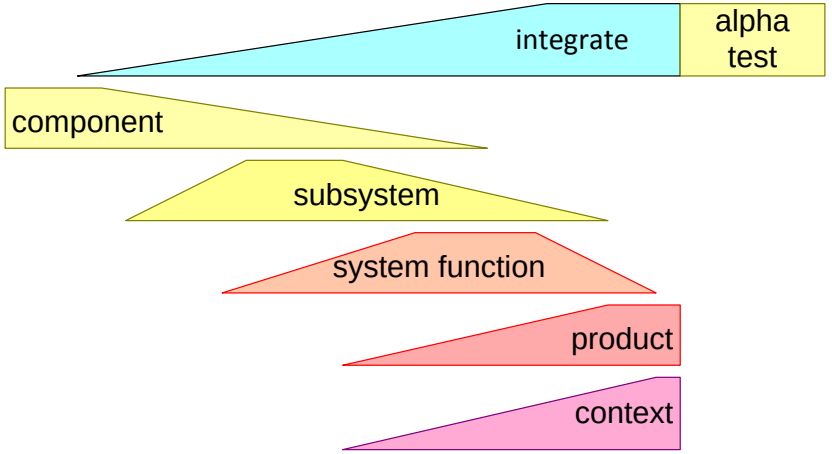
intentionally left blank

Systems Integration

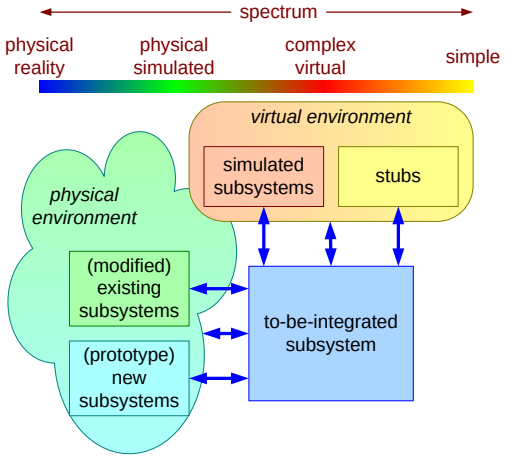
Integration Starts at Feasibility



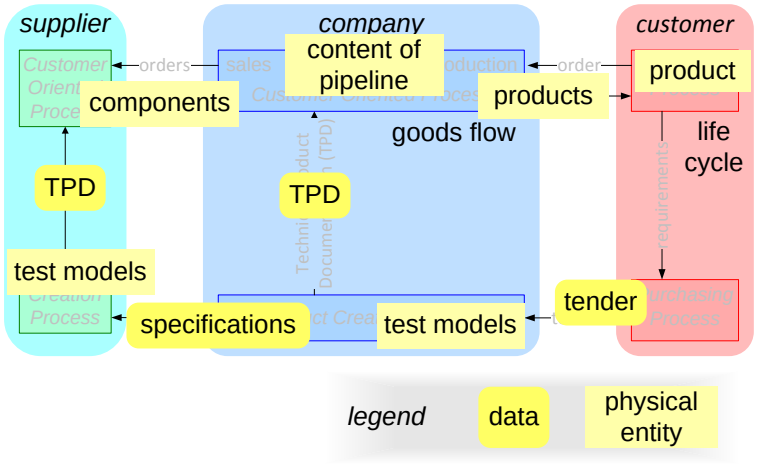
Bottom-up



Alternatives for Early Integration



Propagation of Configuration Issues



Module Management Presentation

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses the presentation of architectural issues to higher management teams.

January 22, 2023
status: draft
version: 1.1

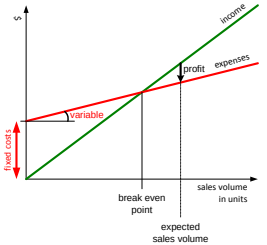
Simplistic Financial Computations for System Architects.

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

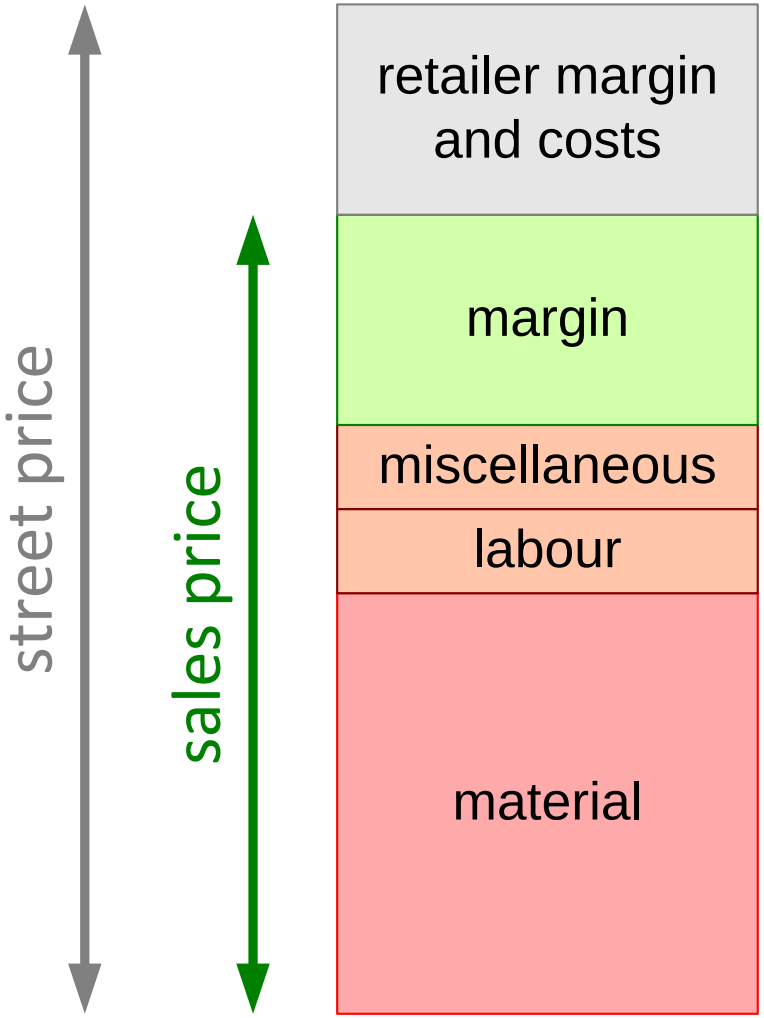
Abstract

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.

January 22, 2023
status: preliminary draft
version: 1.3



Product Margin = Sales Price - Cost



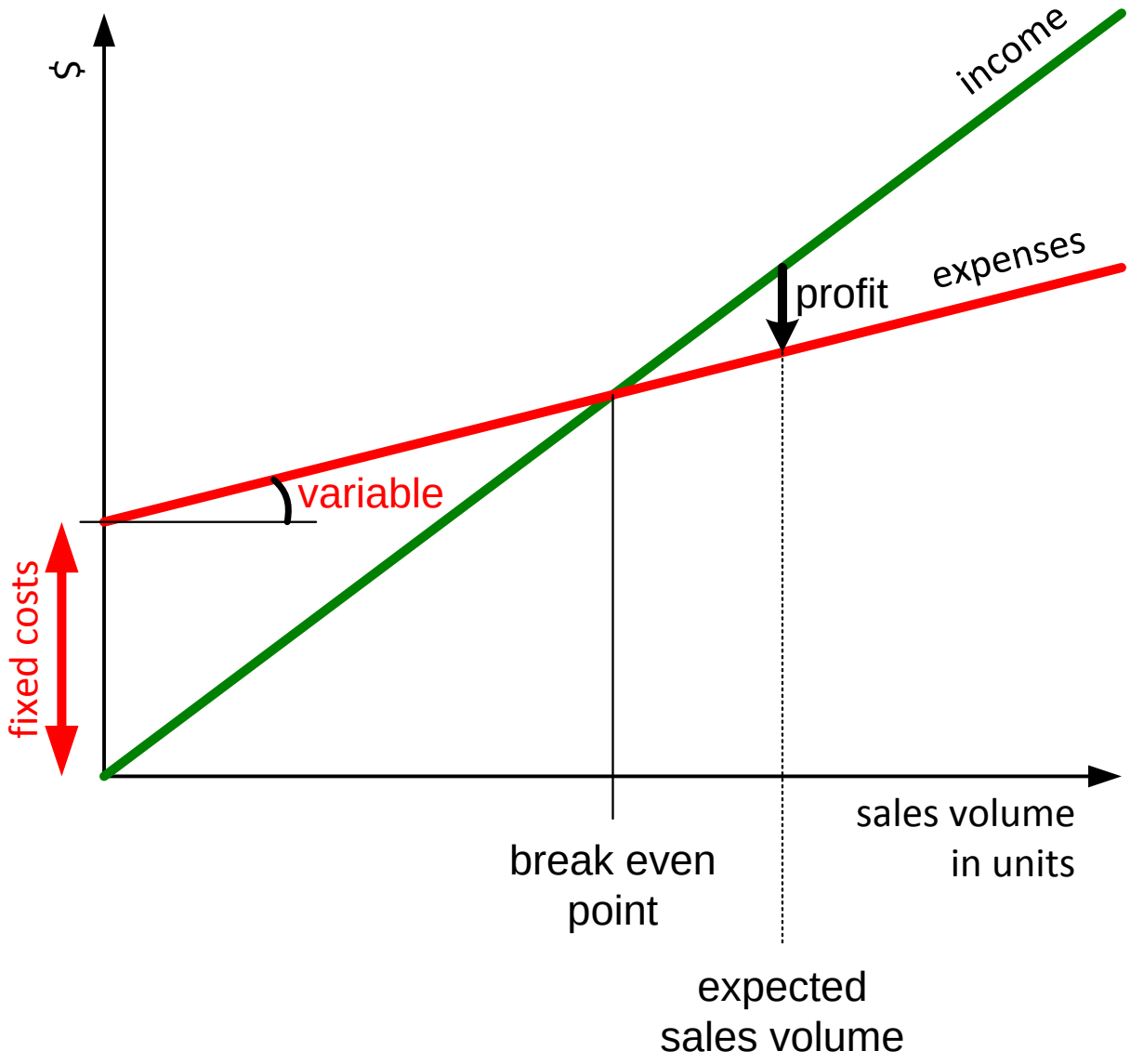
Margin per product.
The margin over the sales volume,
must cover the fixed costs, and generate profit

transportation, insurance,
royalties per product, ...

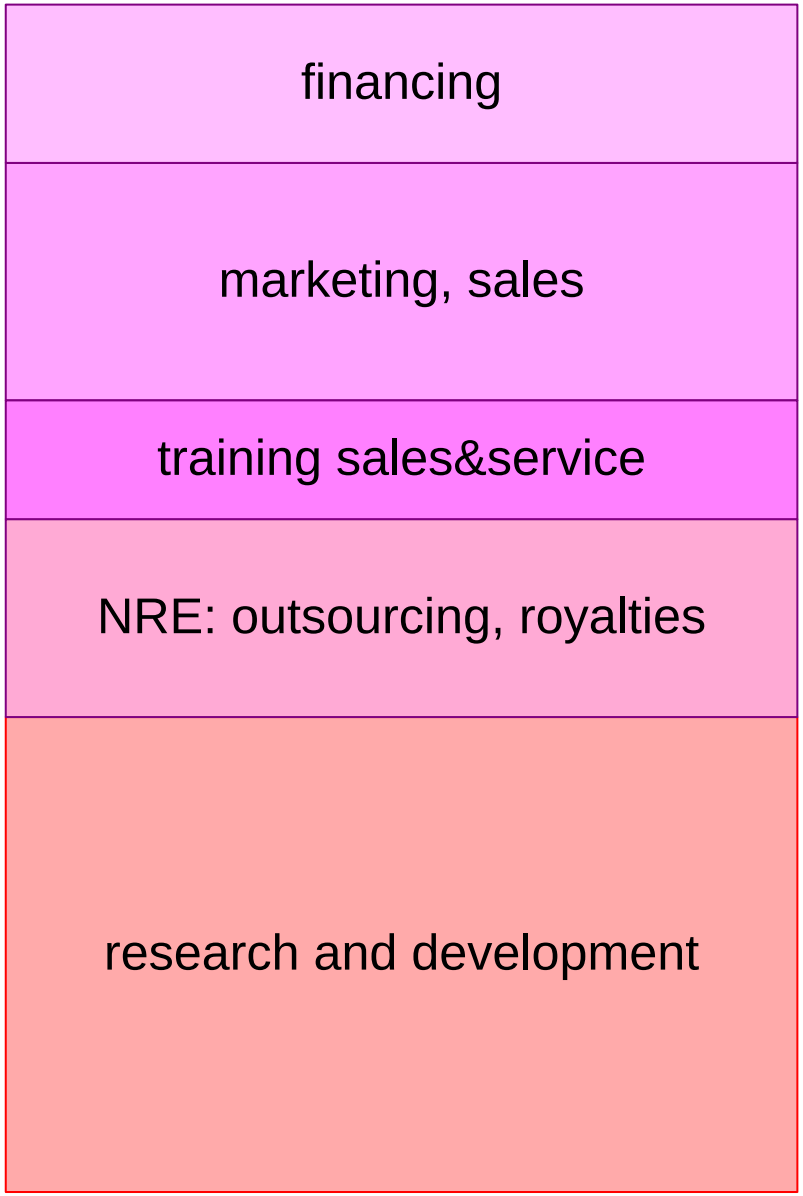
Cost per product,
excluding fixed costs

purchase price of components may cover
development cost of supplier

Profit as function of sales volume



Investments, more than R&D



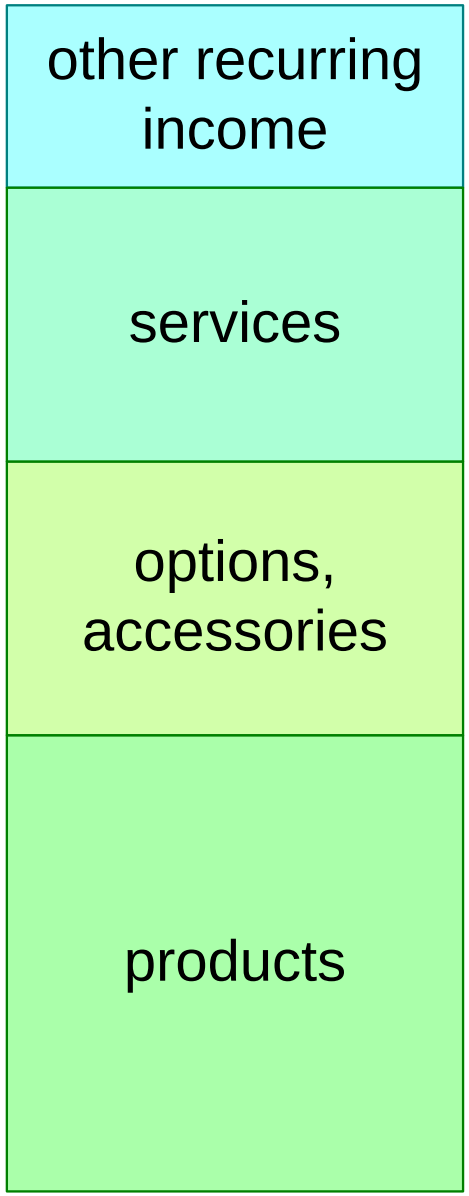
business dependent:
pharmaceutics industry
sales cost >> R&D cost

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

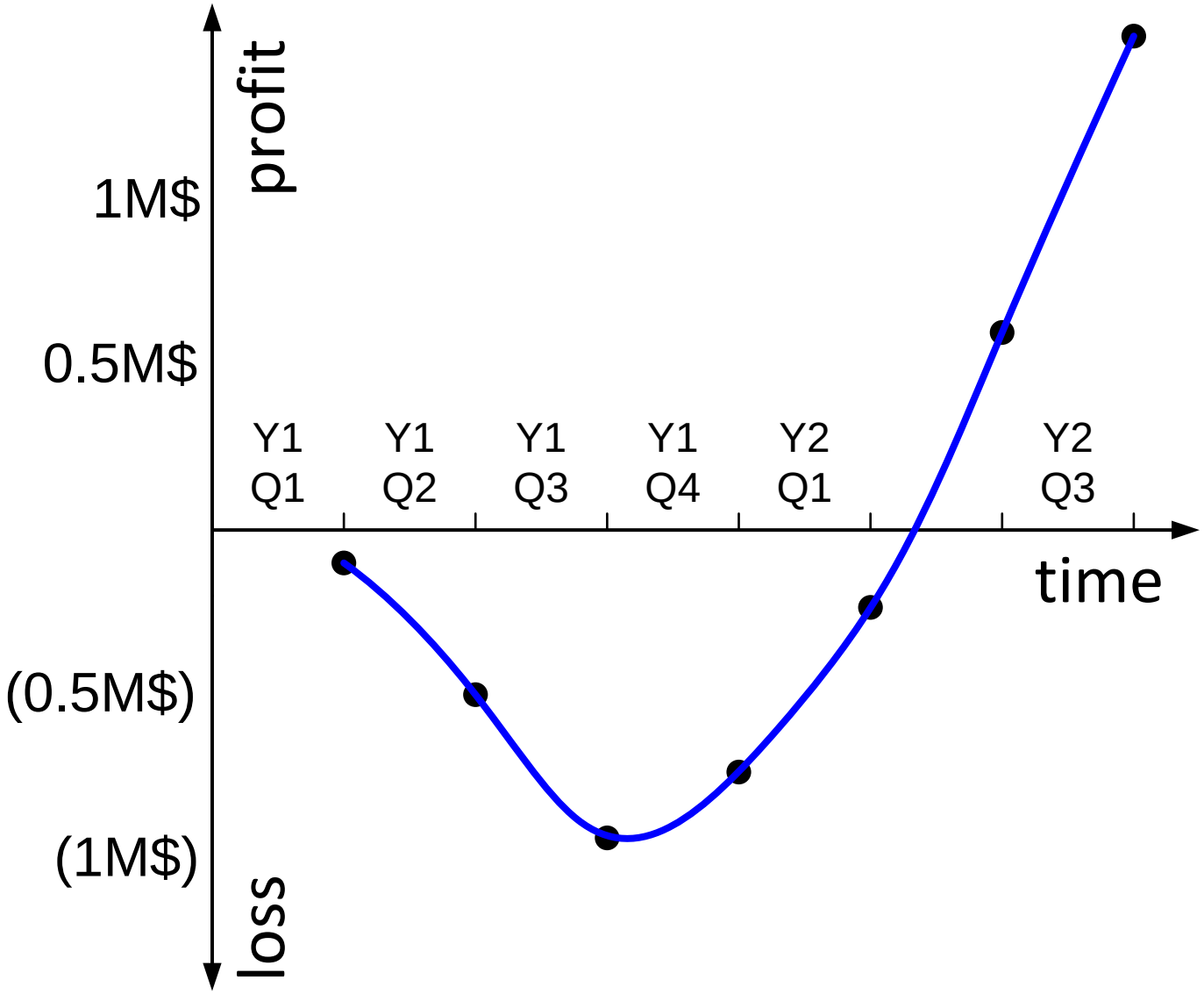
The Time Dimension

	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$

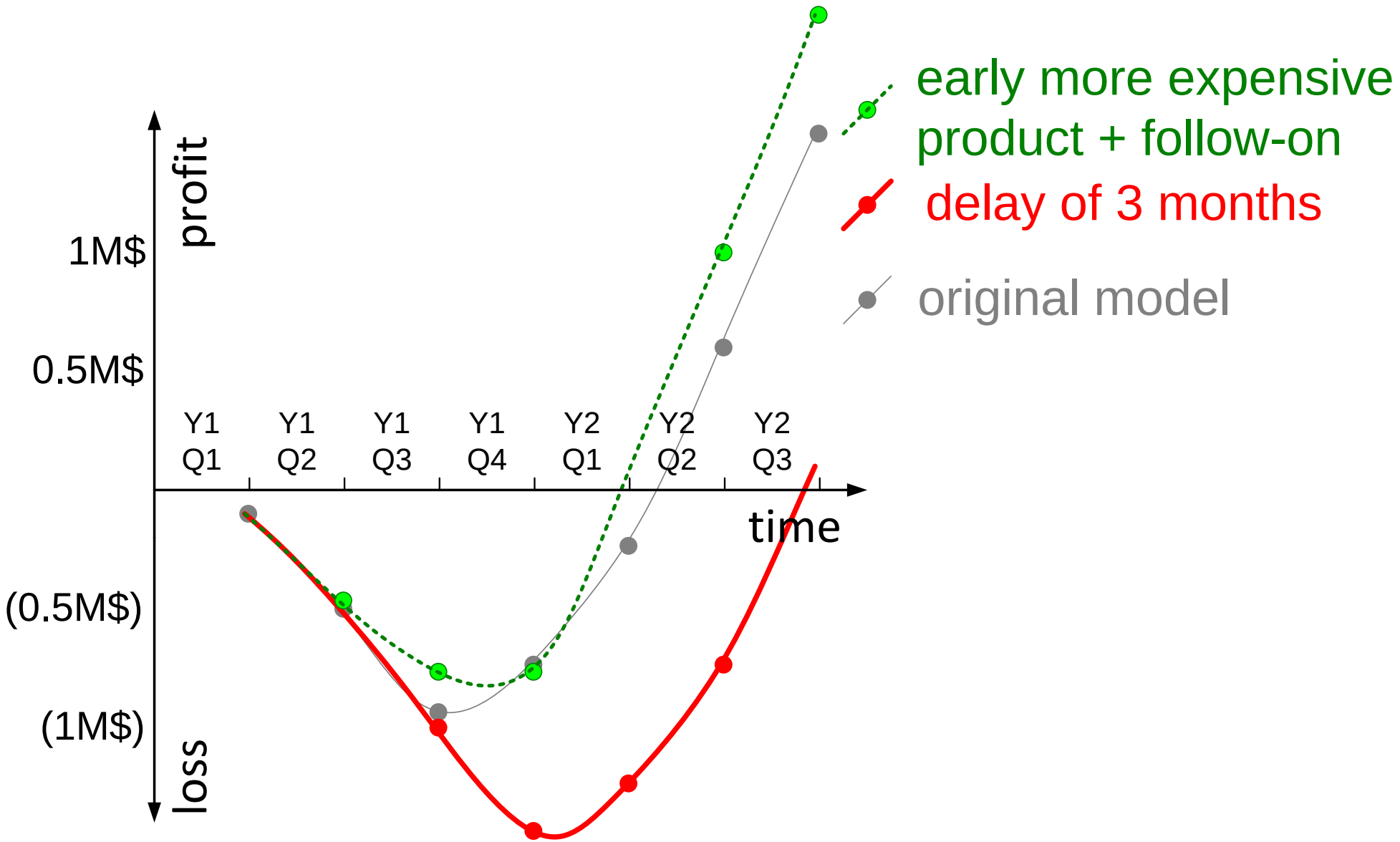
cost price / unit = 20k$
sales price / unit = 50k$

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

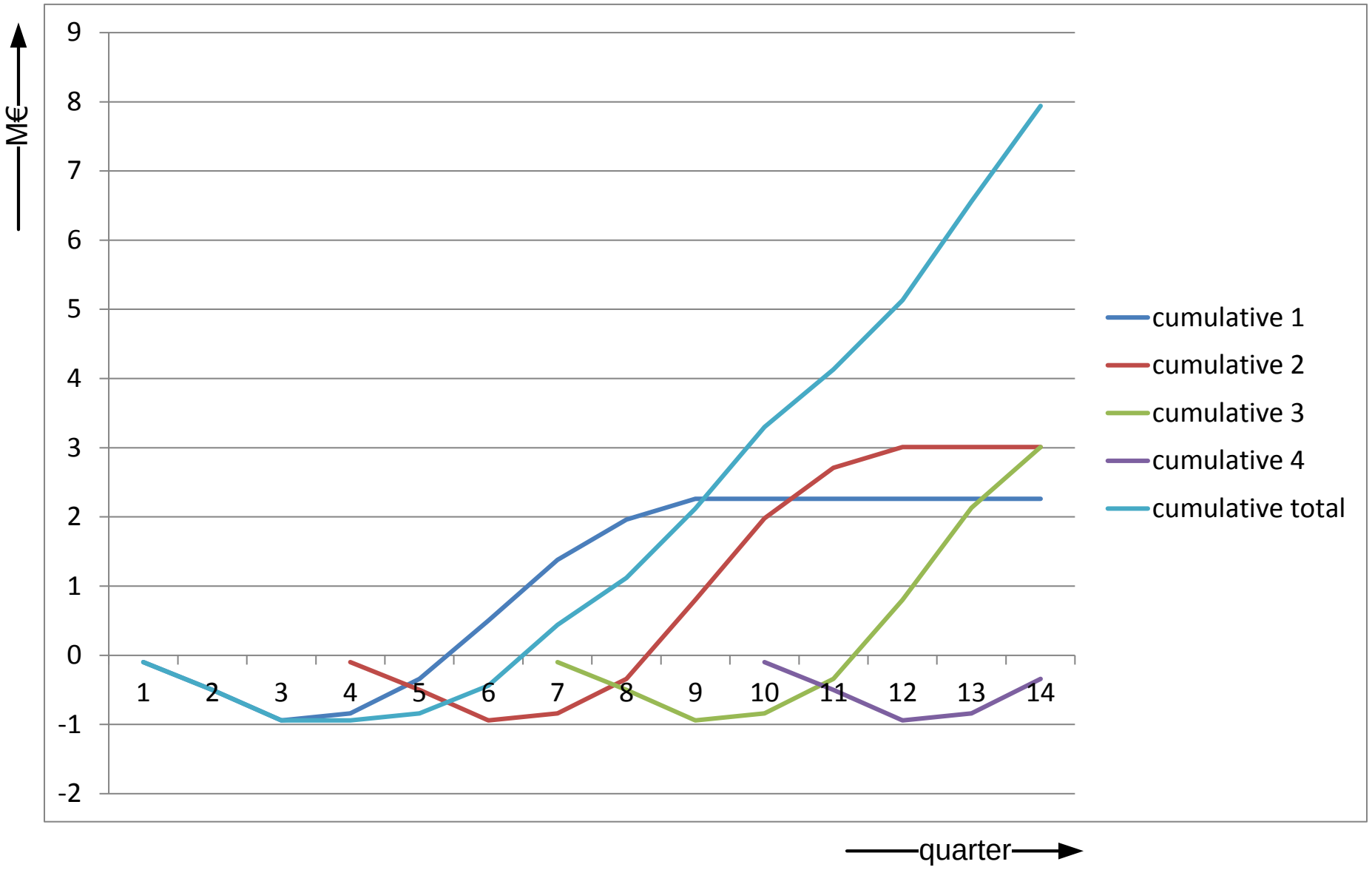
The “Hockey” Stick



What if ...?


profit
1M$
0.5M$
(0.5M$)
(1M$)
loss
Y1 Q1
Y1 Q2
Y1 Q3
Y1 Q4
Y2 Q1
Y2 Q2
Y2 Q3
time
early more expensive product + follow-on
delay of 3 months
original model

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

How to present architecture issues to higher management

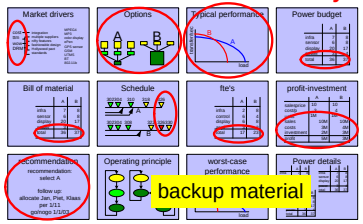
by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

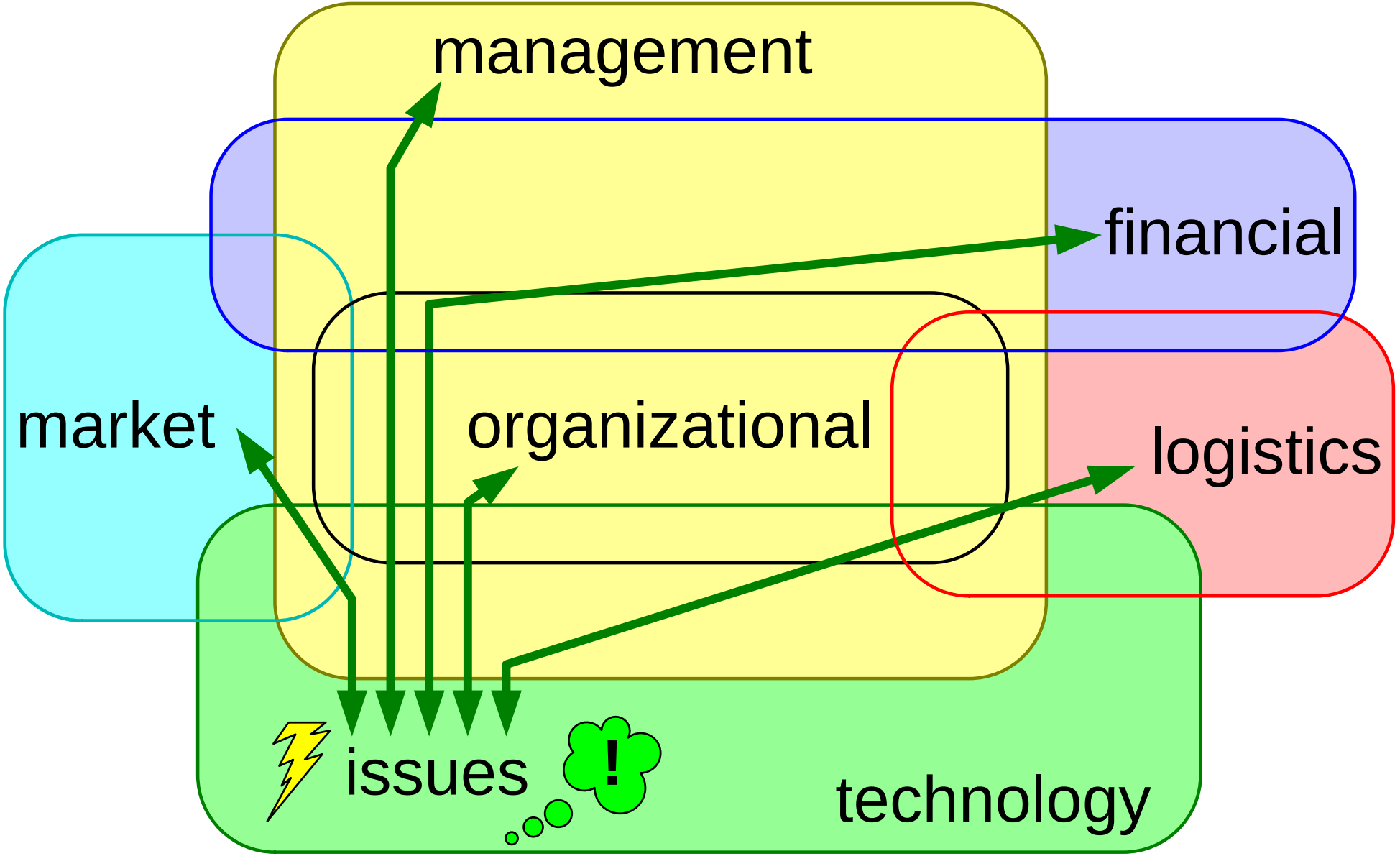
Architects struggle with their visibility at higher management echelons. The introvert nature of architects is a severe handicap. Participation of architects in management teams is important for balanced technical sound decisions and strategy. Improved managerial communication skills of architects are required.
This article describes how to give a more effective presentation to higher management teams. Subjects discussed are the preparation, content and form, do and don't advise.

January 22, 2023
status: concept
version: 0.1

mention the red information only



Architectural issues related to managerial viewpoints



Characteristics of managers in higher management teams

common characteristics

+ action-oriented
+ solution rather than problem
+ impatient, busy
+ want facts not beliefs
+ operate in a political context
+ bottom-line oriented:
  profit, return on investment,
  market share, etc.

highly variable characteristics

? technology knowledge
  from extensive to shallow

? style from power play to
  inspirational leadership

How to prepare

Always prepare with small team!

content ⟷ mutual interaction ⟷ understand audience

70% of effort

+ gather facts
+ perform analysis
+ identify goal and message
+ make presentation
+ polish presentation form

30% of effort

+ gather audience background
+ analysis audience interests
+ identify expected responses
+ simulate audience, exercise presentation

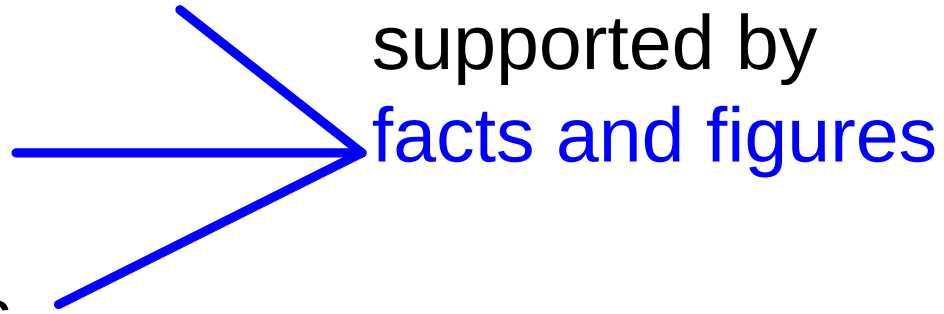
Recommended content

+ clear problem statement (what, why)

+ solution exploration (how)

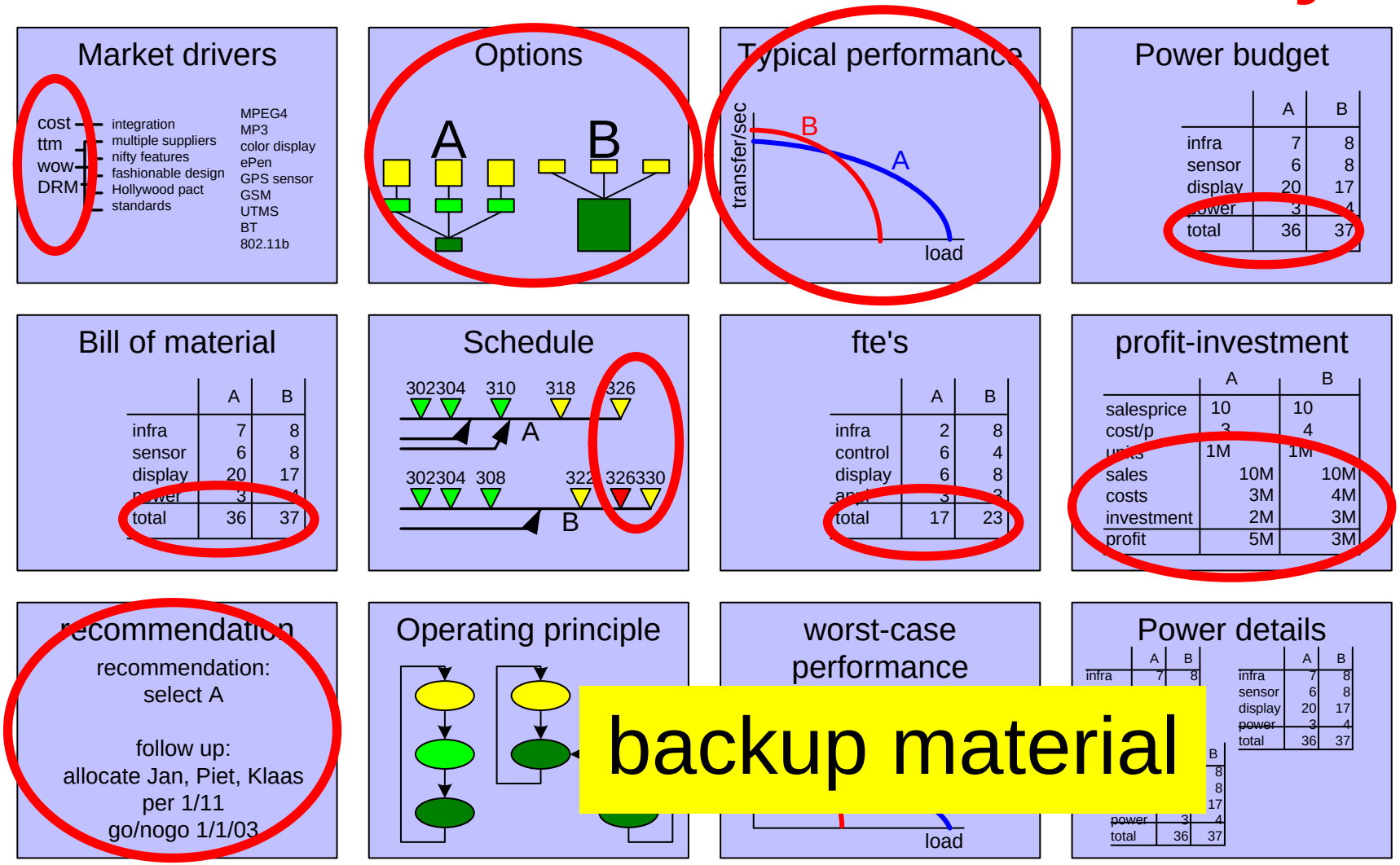
+ options, recommendations

+ expected actions or decisions



Mentioned info, shown info and backup info

mention the red information only



Form is important

poor form can easily distract from purpose and content

presentation material

+ professional
+ moderate use of color and animations
+ readable
+ use demos and show artifacts

presenter's appearance

+ well dressed
+ self confident but open

but stay yourself, stay authentic

Don’t force your opinion, understand the audience

do not	*do*
- preach beliefs	+ quantify, show figures and facts
- underestimate technology knowledge of managers	+ create faith in your knowledge
- tell them what they did wrong	+ focus on objectives
- oversell	+ manage expectations

How to cope with managerial dominance

do not	*do*
- let one of the managers hijack the meeting	+ maintain the lead
- build up tensions by withholding facts or solutions	+ be to the point and direct
- be lost or panic at unexpected inputs or alternatives	+ acknowledge input, indicate consequences (facts based)

Exercise presentation to higher management

+ Bring a clear architecture message to

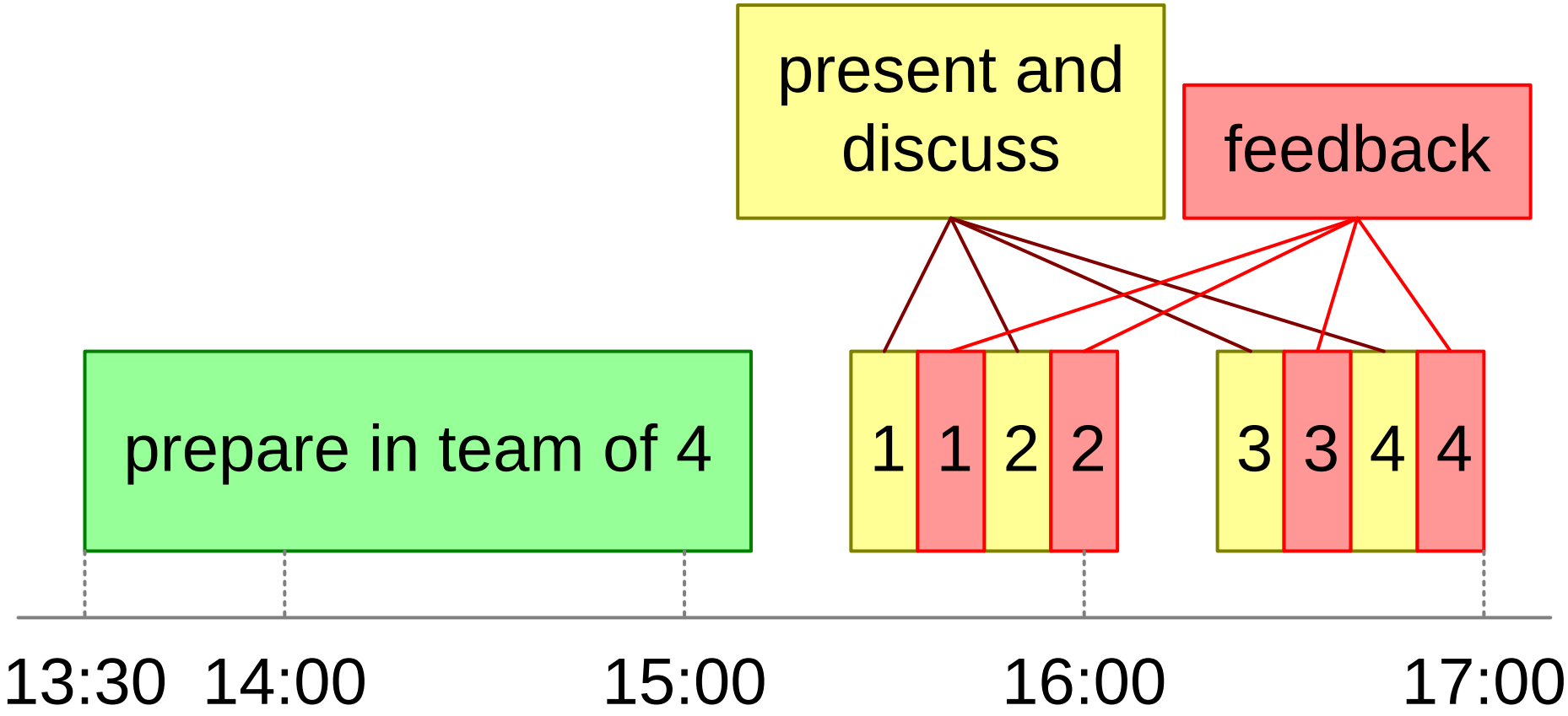
+ a Management team at least 2 hierarchical levels higher

+ with 10 minutes for presentation including discussion
(no limitation on number of slides)

* architecture message =
technology options in relation with market/product

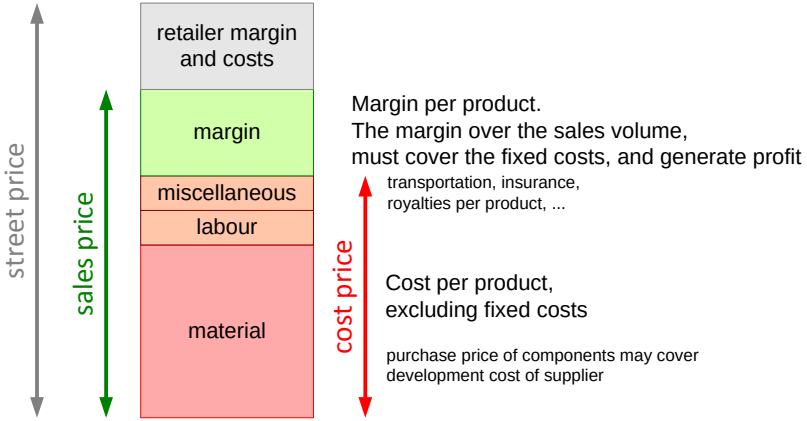
* address the concerns of the management stakeholders:
translation required from technology issues into
business consequences (months, fte's, turnover, profit, investments)

Exercise schedule



Simplistic Financial Computations

Product Margin = Sales Price - Cost



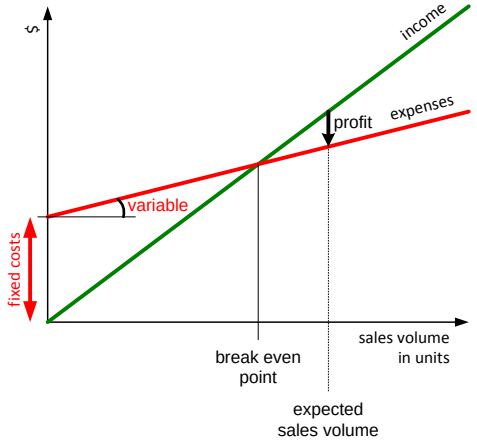
Margin per product.
The margin over the sales volume, must cover the fixed costs, and generate profit

transportation, insurance, royalties per product, ...

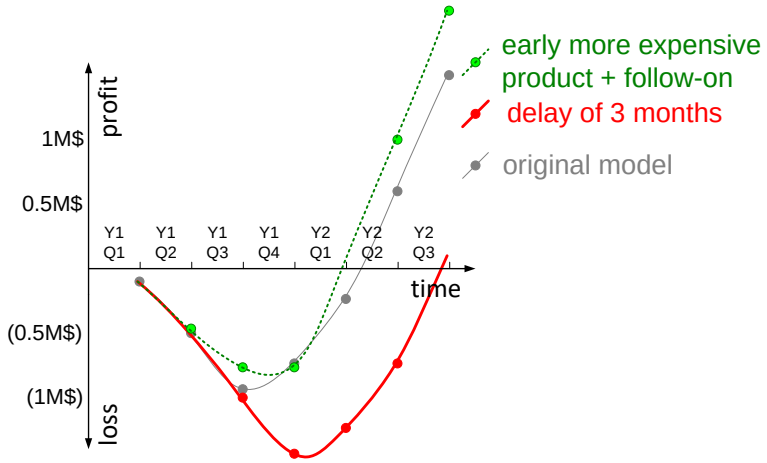
Cost per product, excluding fixed costs

purchase price of components may cover development cost of supplier

Profit as function of sales volume



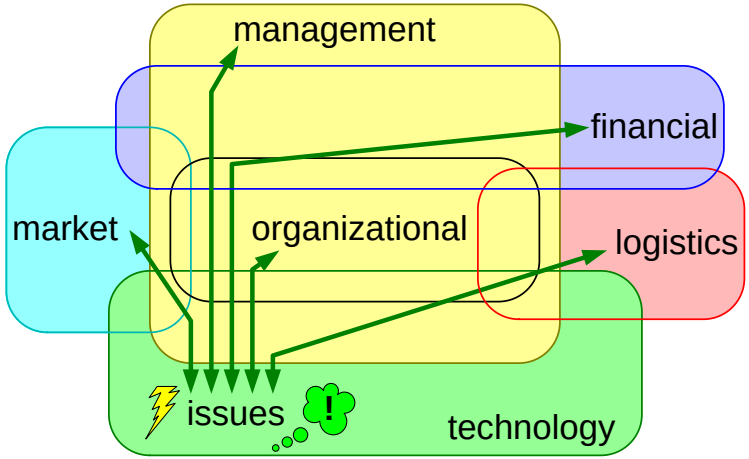
Hockey stick and scenarios



intentionally left blank

Presentation to Management

Managerial Viewpoints



Prepare Content, Understand Audience

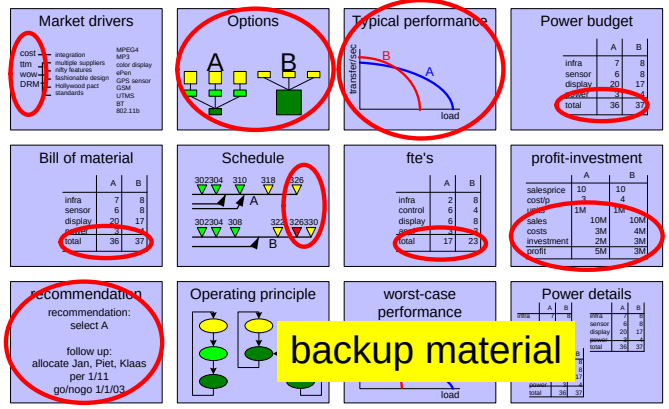
Always prepare with small team!

content (70% of effort)	understand audience (30% of effort)
+ gather facts	+ gather audience background
+ perform analysis	+ analysis audience interests
+ identify goal and message	+ identify expected responses
+ make presentation	+ simulate audience, exercise presentation
+ polish presentation form	

mutual interaction

Show underlying info

mention the red information only



Form, do and do not

poor form can easily distract from purpose and content

presentation material

+ professional
+ moderate use of color and animations
+ readable
+ use demos and show artifacts

presenter's appearance

+ well dressed
+ self confident but open

but stay yourself, stay authentic

Module Roadmapping

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses roadmapping.

January 22, 2023
status: draft
version: 1.2

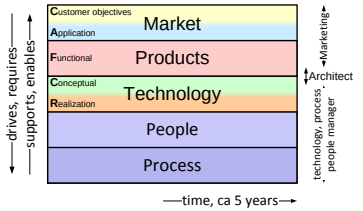
Roadmapping

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

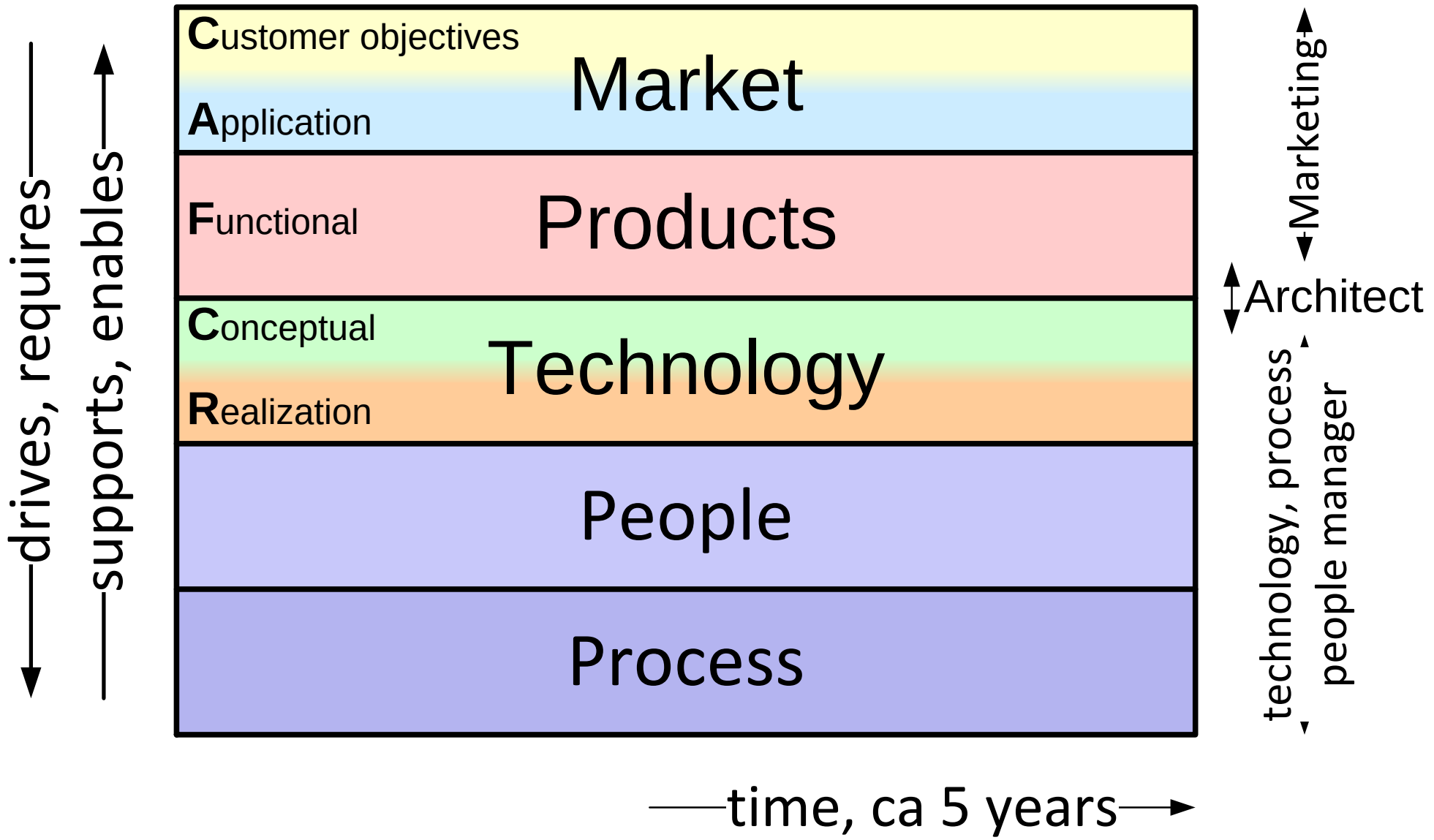
Abstract

This article describes what a roadmap is, how to create and maintain a roadmap, the involvement of the stakeholders, and criteria for the structure of a roadmap.

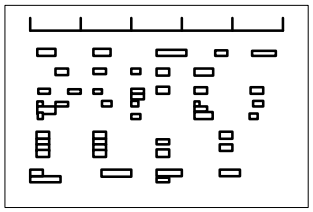
January 22, 2023
status: concept
version: 2.0



The Roadmap Integrates Five Views



Granularity of Roadmap Material

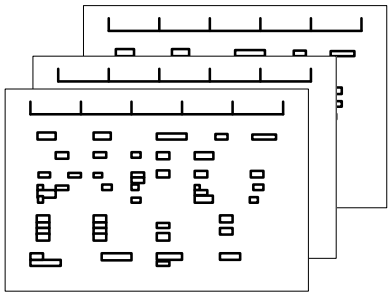


Top-level roadmap

Single page

Poster

part of many presentations

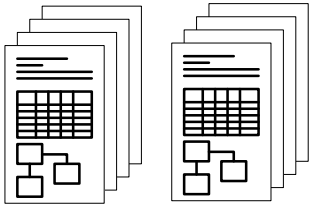


Supporting roadmaps

Single page per view or per driver

Poster

part of many presentations



Supporting reports

Document per relevant subject

Problems that Occur without Roadmapping

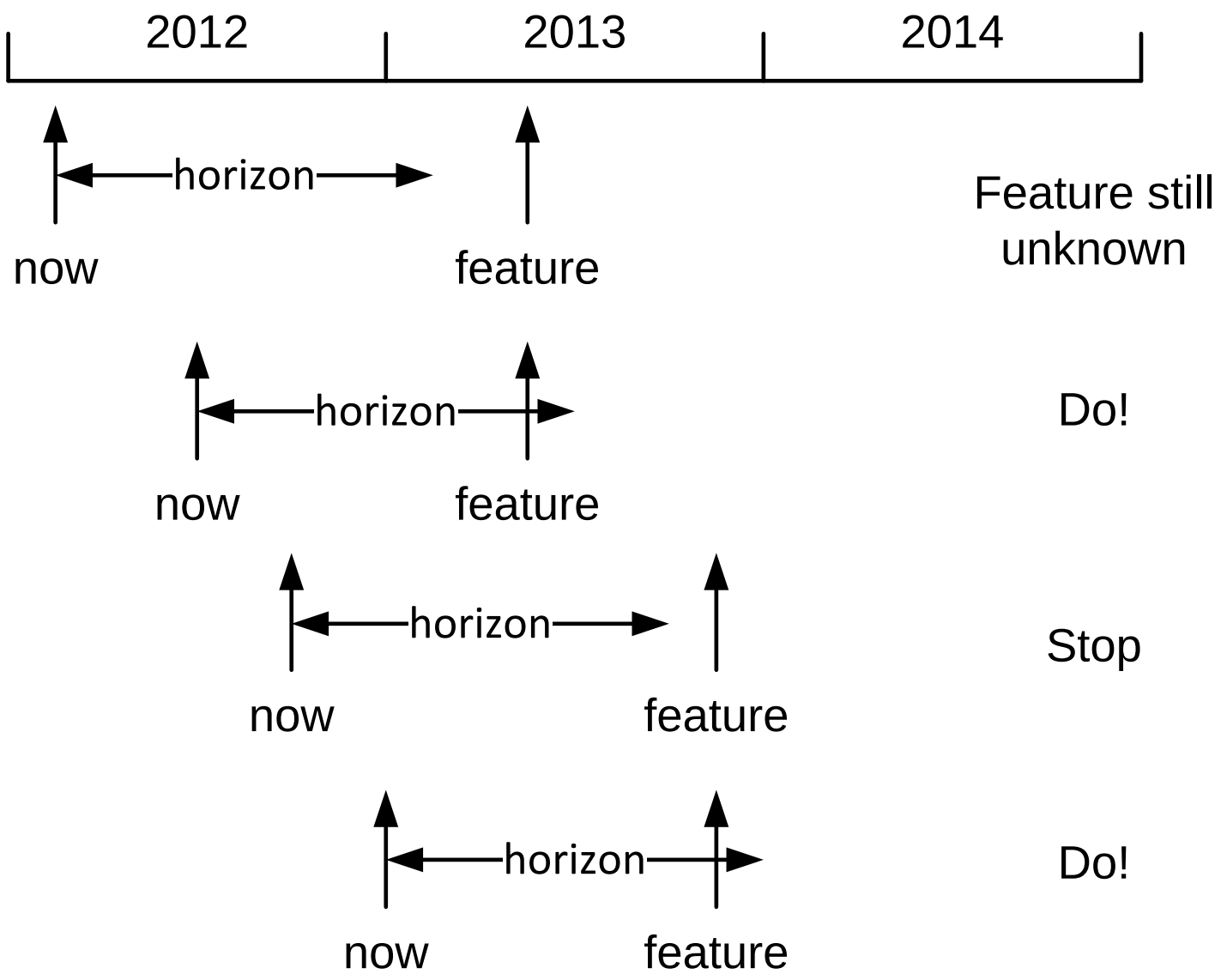
Frequent changes in product policy

Late start up of long lead activities, such as people recruitment and process change

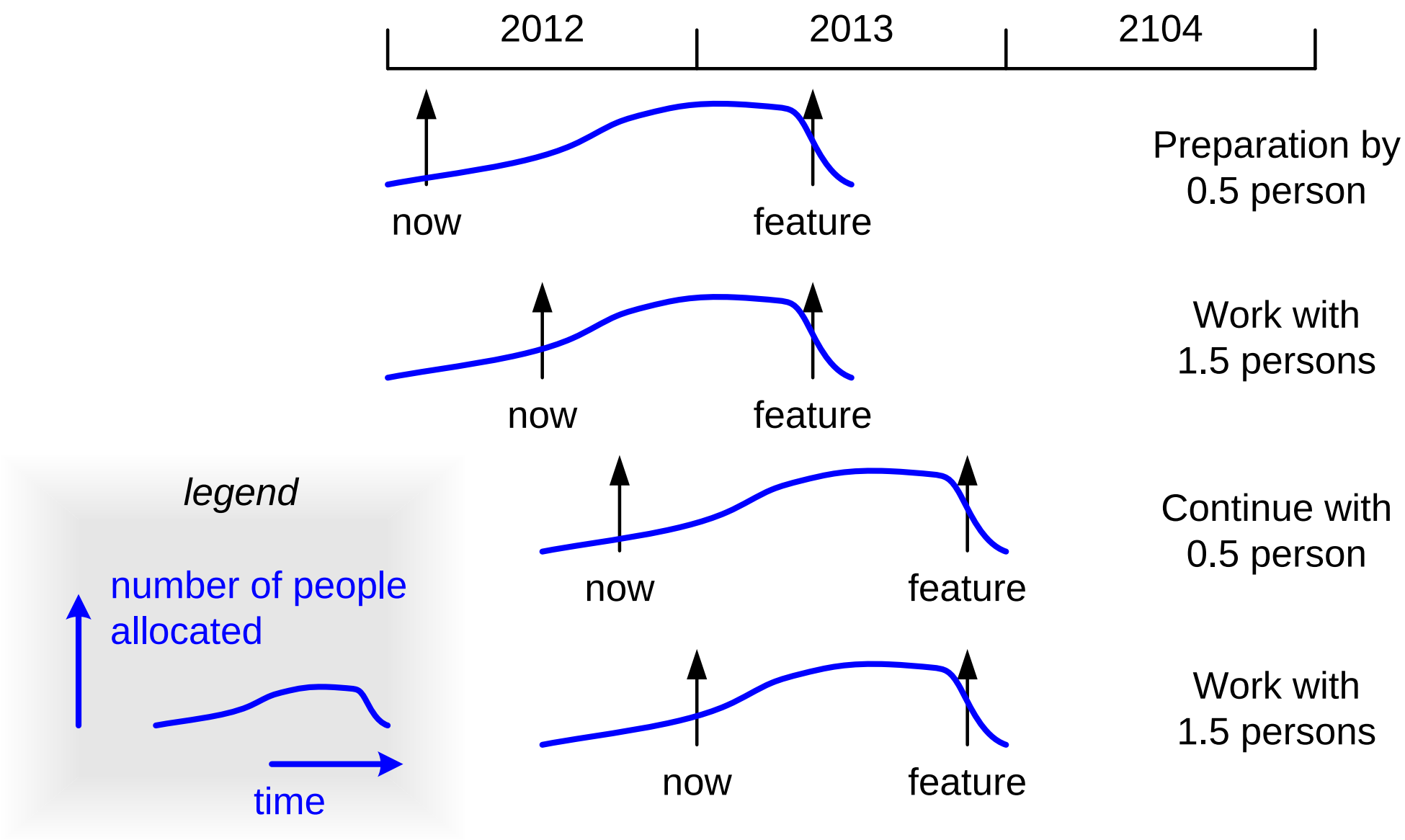
Diverging activities of teams

Missed market opportunities

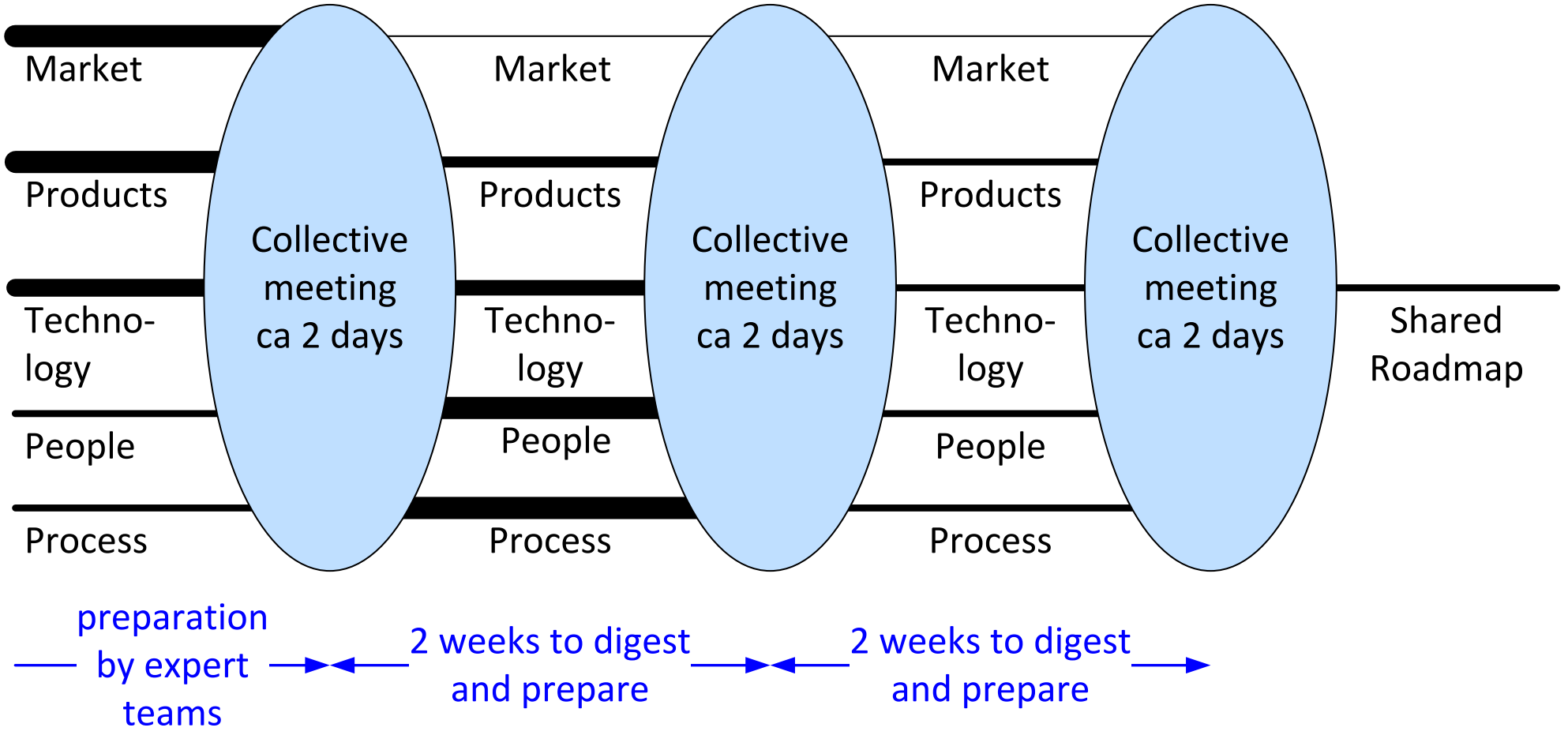
Management with a Limited Horizon



Management with a Broader Time Perspective



Creation or Update of Roadmap in Burst Mode



Typical Stakeholders of a Roadmap

business manager overall enterprise responsible

marketing manager(s)

discipline or line managers

people, process, and technology manager(s)

operational manager(s) project or program managers

architect(s)

Target of the First Session

Shared vision on market

First iteration of possible products as an answer to the market

Share technology status, as starting point for technology roadmap

Explore people and technology status, to identify main issues

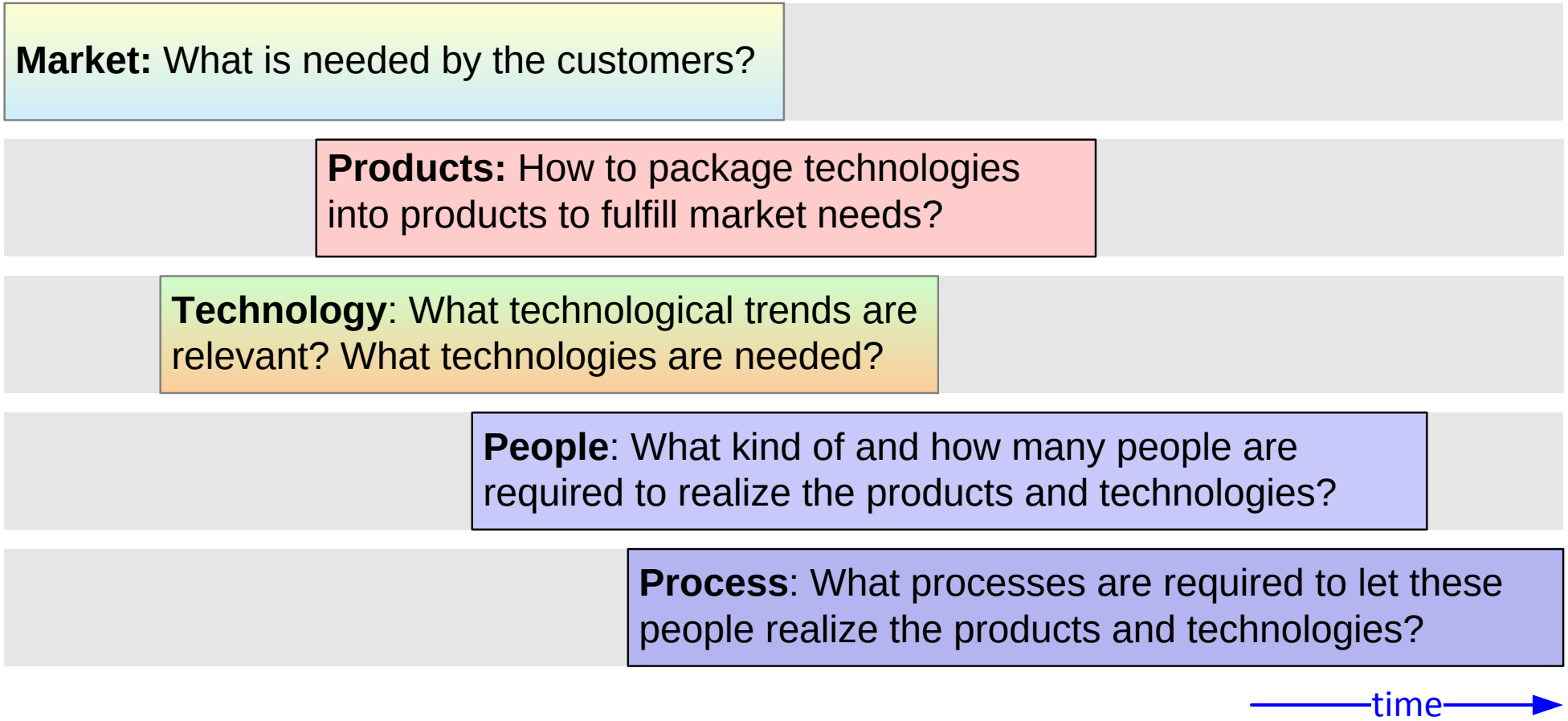
Target of the Second Session

Obtaining a shared vision on the desired technology roadmap

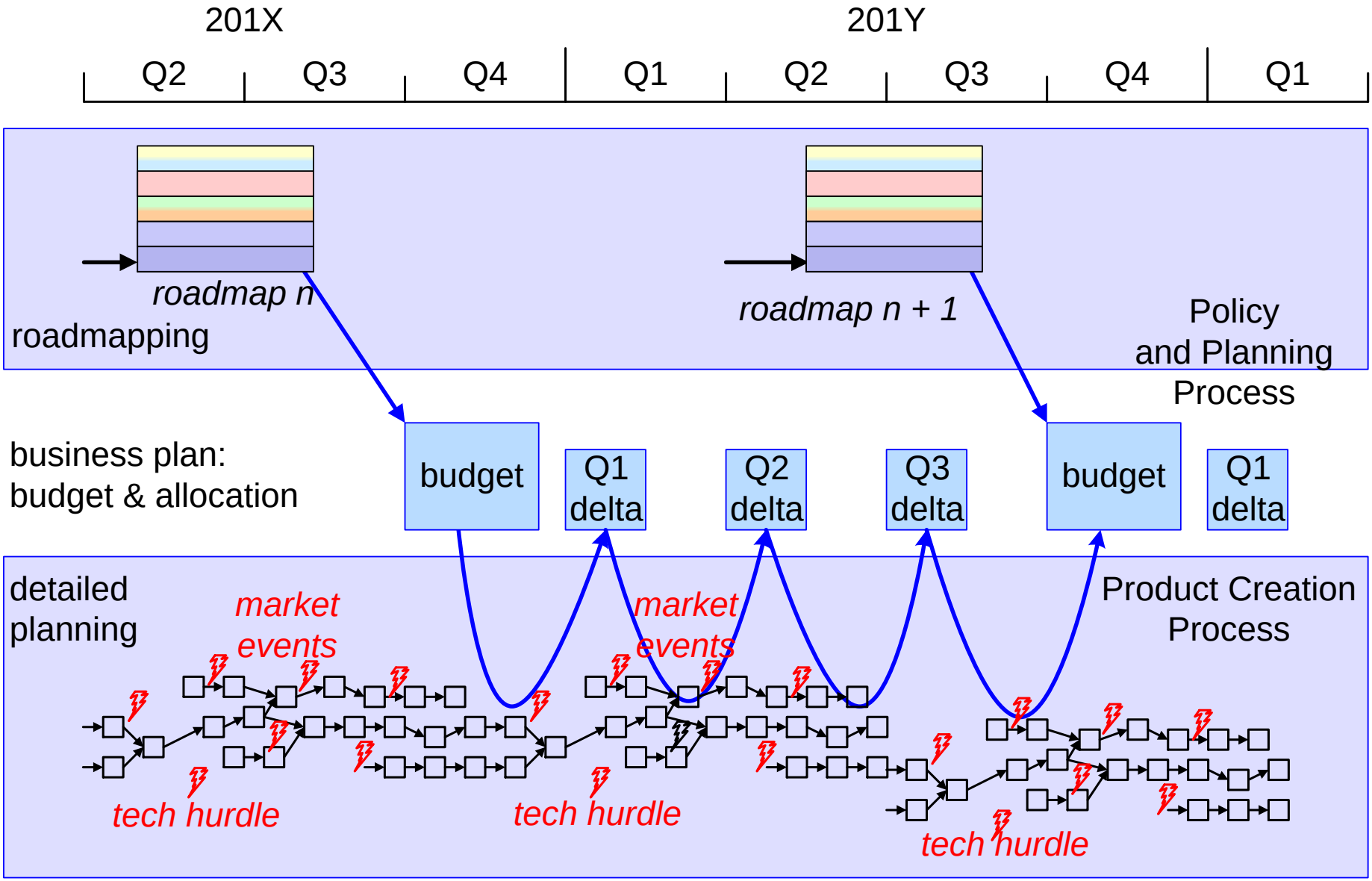
Sharing the people and process issues required for the products defined in the first iteration

Analyzing a few scenarios for products, technologies, people, and process

The Roadmap Update Visualized in Time



From Roadmap to Detailed Plans



3-Tier Approach

	horizon	*update*	*scope*	*type*
roadmap	5 years	1 year	portfolio	vision
budget	1 year	3 months	program	commitment
detailed plan	1 mnth-1yr	1 day-1 mnth	program or activity	control means

Roadmap Essentials

Selection of most important or relevant issues

Key drivers as a means to structure the roadmap

Nothing is certain; ambiguity is normal

Use facts whenever possible

Don’t panic in case of impossibilities

Requirements for a Good Roadmap

Recognizable issues for all stakeholders

Clear positioning in time; uncertainty can be visualized

The main events (enabling or constraining) must be present

Limited amount of information to maintain the overview

Sources of Facts

Market analysis reports

number of customers, market size, competition, trends

Installed base

change requests, problem reports, historical data

Manufacturing (statistical process control)

statistical process control

Suppliers (roadmaps, historical data)

roadmaps, historical data

Internal reports (technology studies, simulations)

technology studies, simulations

Causes for Overestimation

Quantization effects of small activities (the amount of time is rounded to manweeks/months/years)

Uncertainty is translated into margins at every level (module, subsystem, system)

Counting activities twice (e.g., in technology development and in product development)

Quantization effects of persons/roles (full time project leader, architect, product manager, et cetera per product)

Lack of pragmatism (technical ambition is not too bad during the roadmap process, as long as it does not pre-empt a healthy decision)

Too many bells and whistles without business or customer value

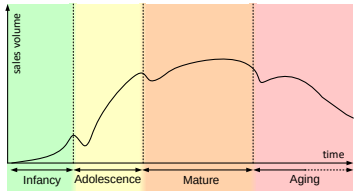
Market Product Life Cycle Consequences for Architecting

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

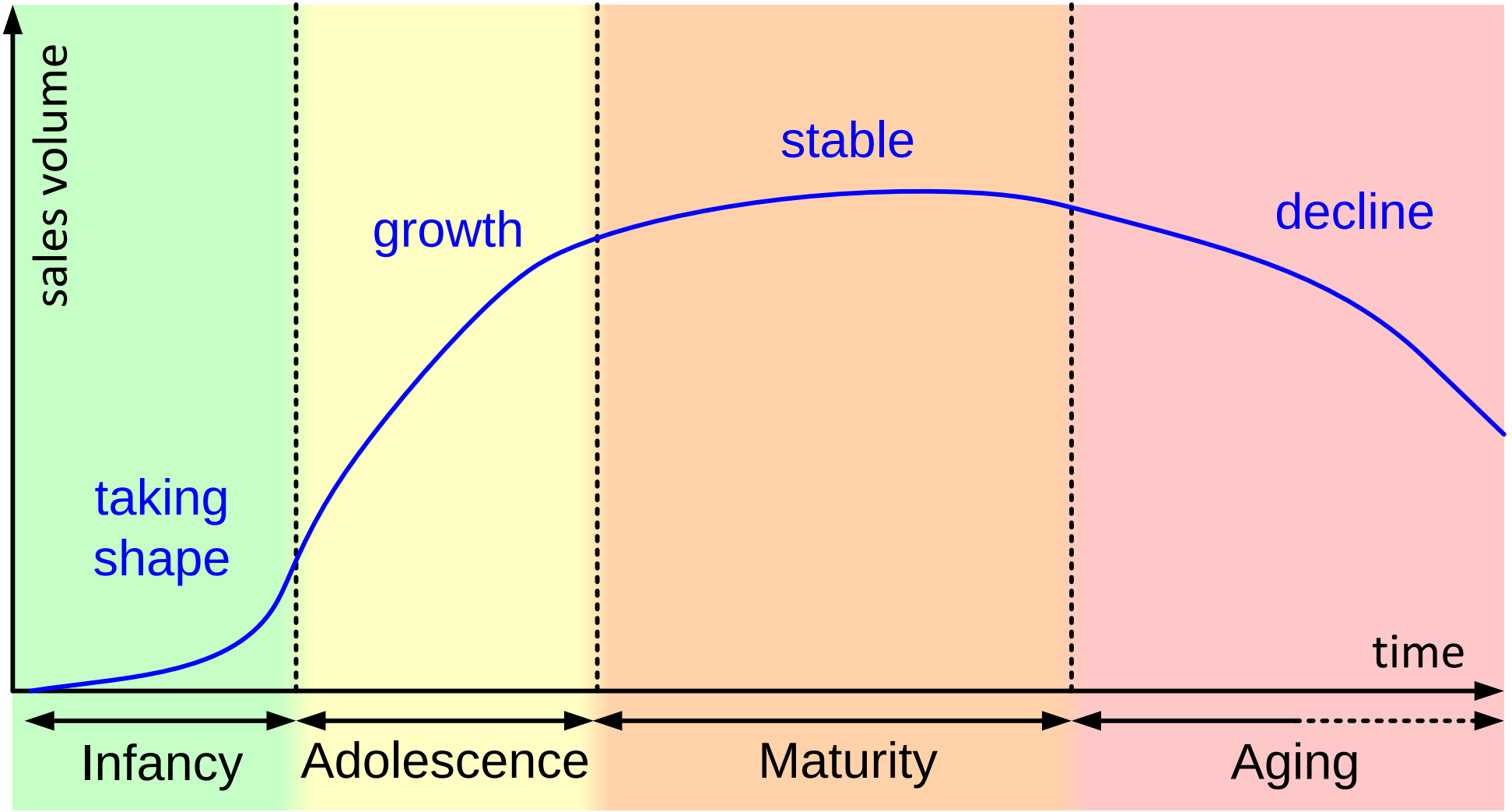
Abstract

The lifecycle of a product category in the market determines many aspects of the architecting approach. The lifecycle consists typical of 4 phases: infancy, adolesence, mature and aging.
A discontinuity in market success is seen in the transition from one phase to the next phase. The explanation given is that the phases differ in characterictics and require different approaches. The right approach for one phase is sub optimal for the next phase. A set of characteristics per phase is given and the consequences for architecting are discussed.

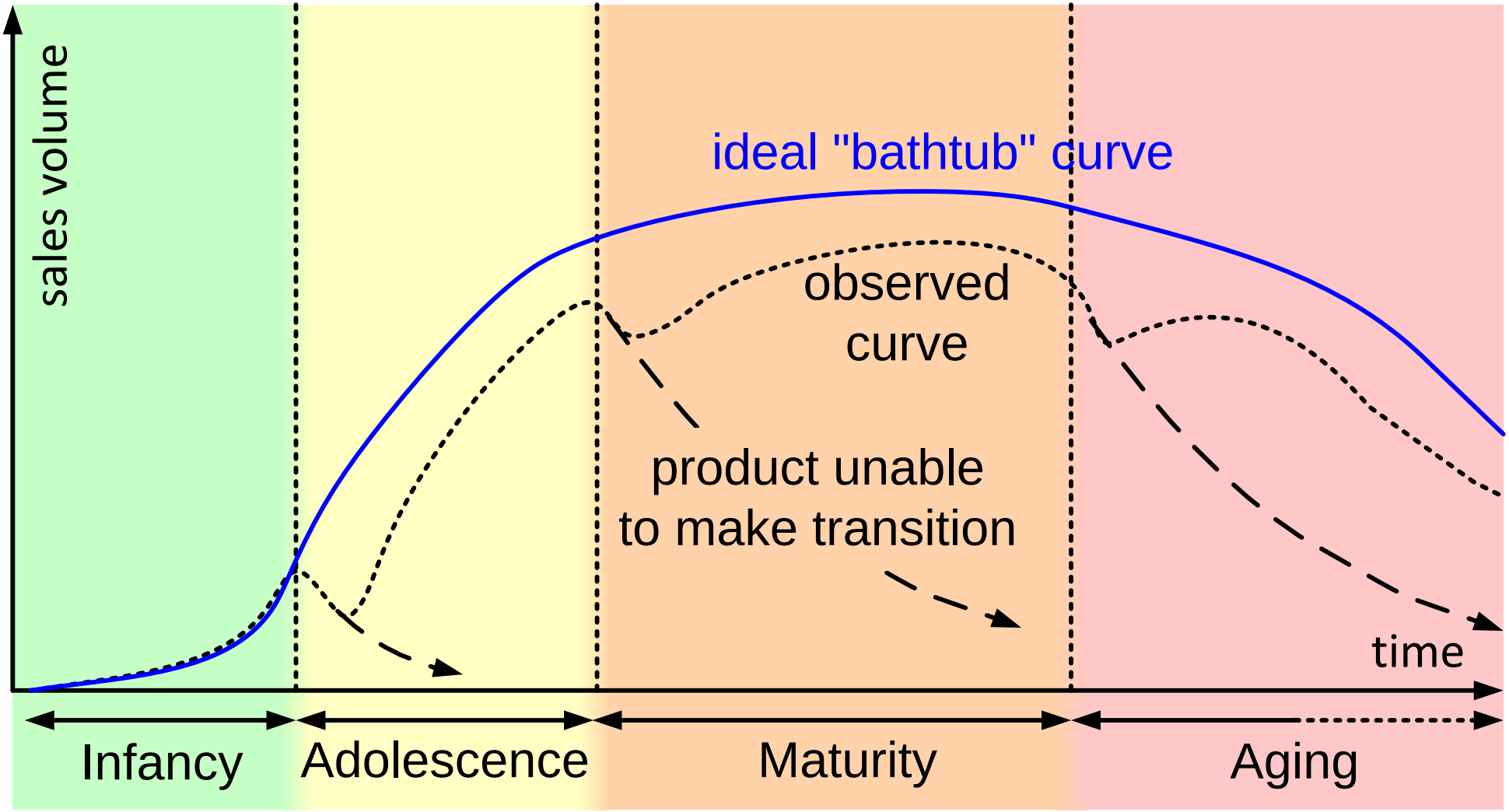
January 22, 2023
status: concept
version: 1.2



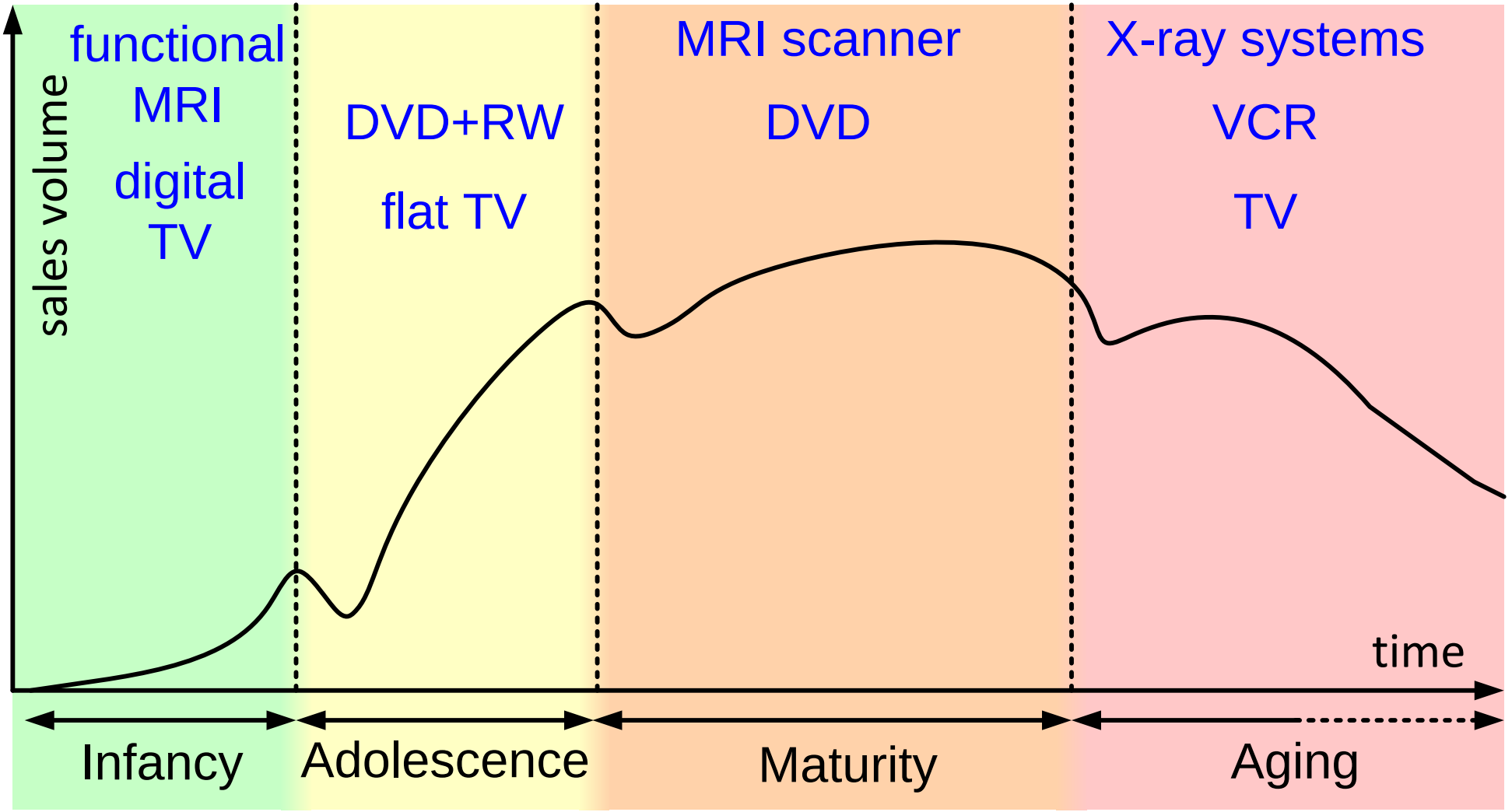
Ideal Bathtub Curve



Market Product Life Cycle Phases in Practice



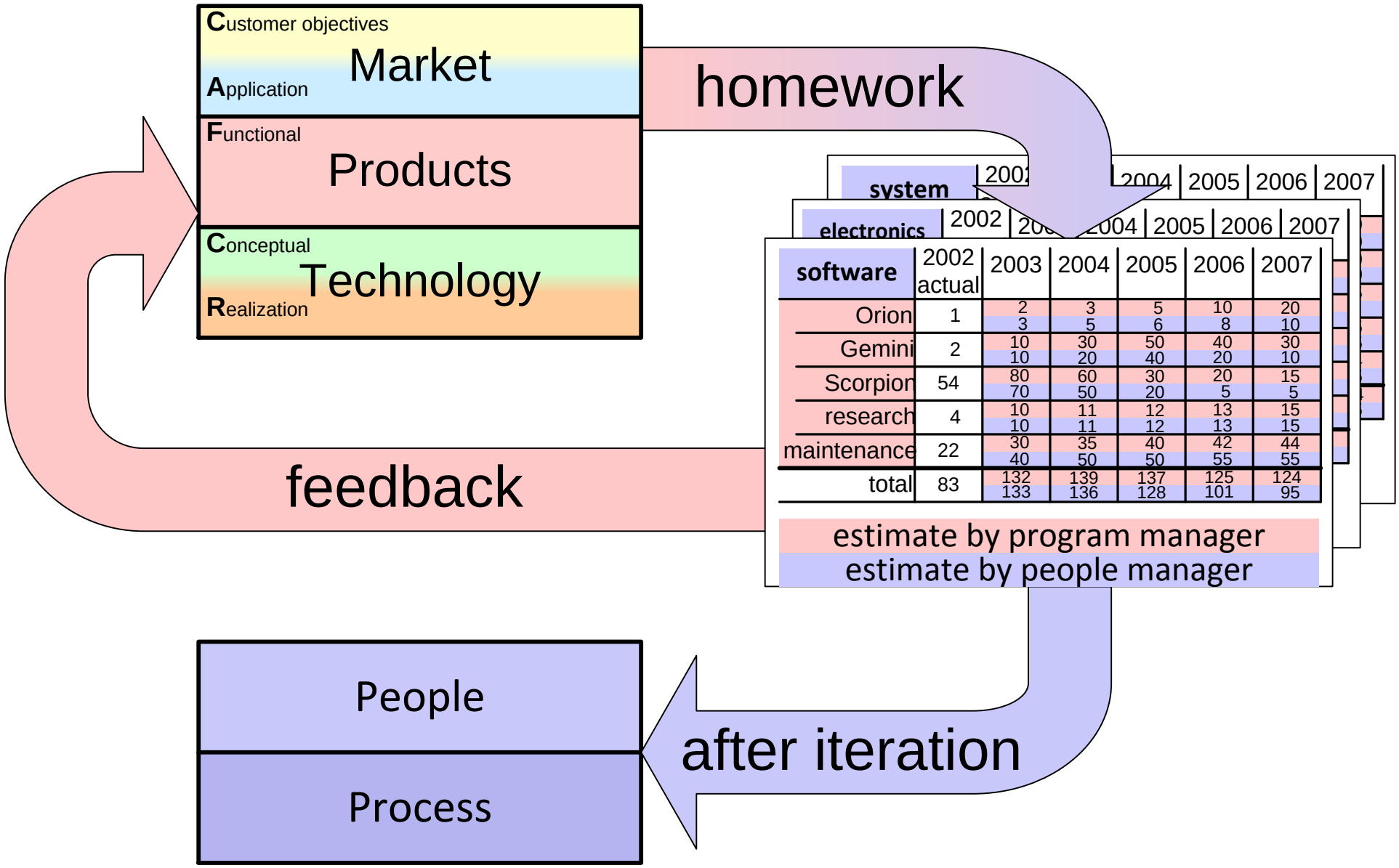
Examples of Product Classes on the Curve



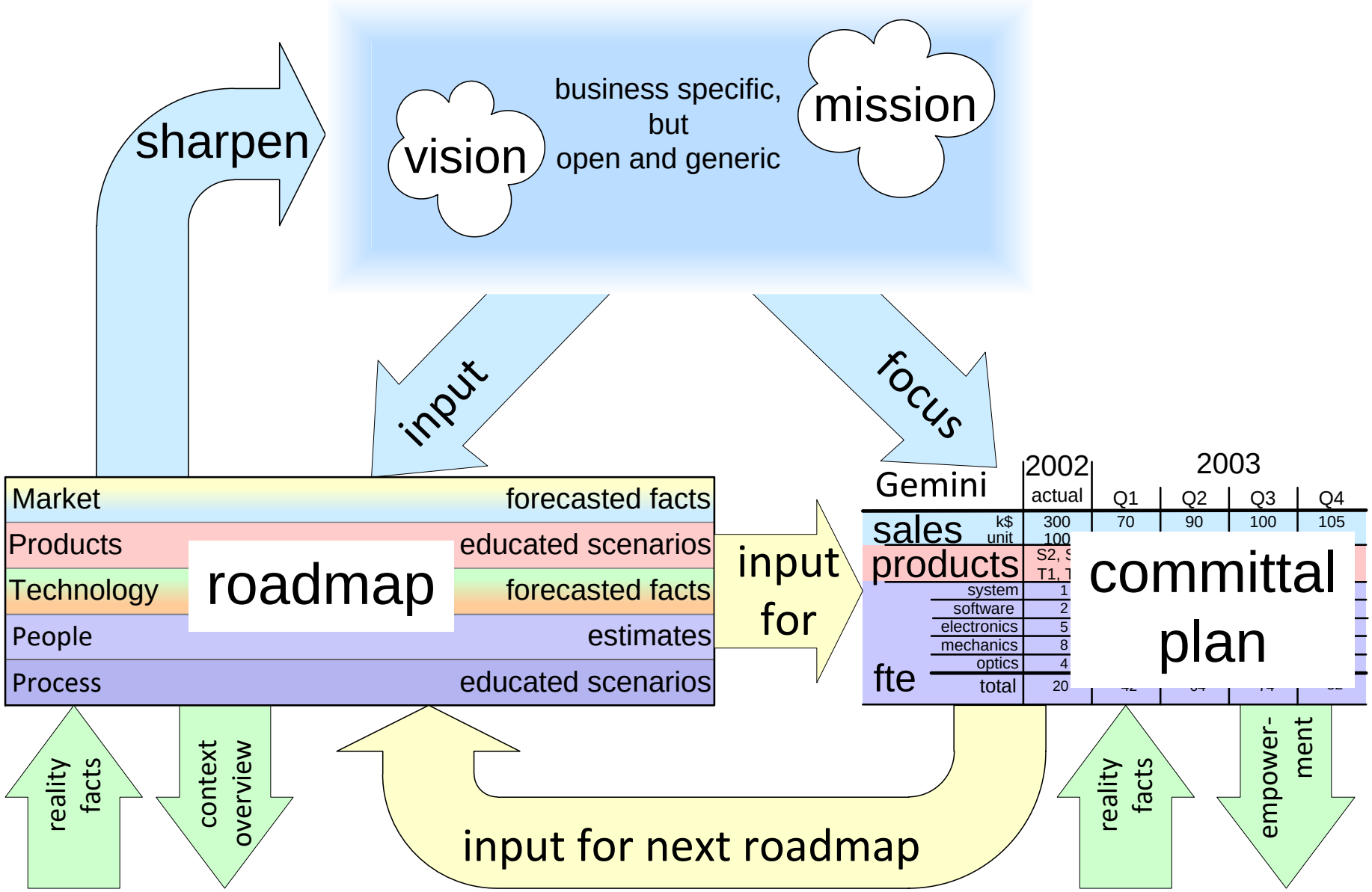
Attributes per Phase

	Infancy	Adolescence	Mature	Ageing
Driving factor	Business vision		Stable business model	Harvesting of assets
Value from	Responsiveness	Features	Refinements / service	Refining existing assets
Requirements	Discovery	Select strategic	Prioritize	Low effort high value only
Dominant technical concerns	Feasibility	Scaling	Legacy Obsolescence	Lack of product knowledge Low effort for obsolete technologies
Type of people	Inventors & pioneers	Few inventors & pioneers "designers"	"Engineers"	"Maintainers"
Process	Chaotic		Bureaucratic	Budget driven
Dominant pattern	Overdimensioning	Conservative expansion	Midlife refactoring	UI gadgets

From Market, Product, Technology to People, Process



Summary of strategy process

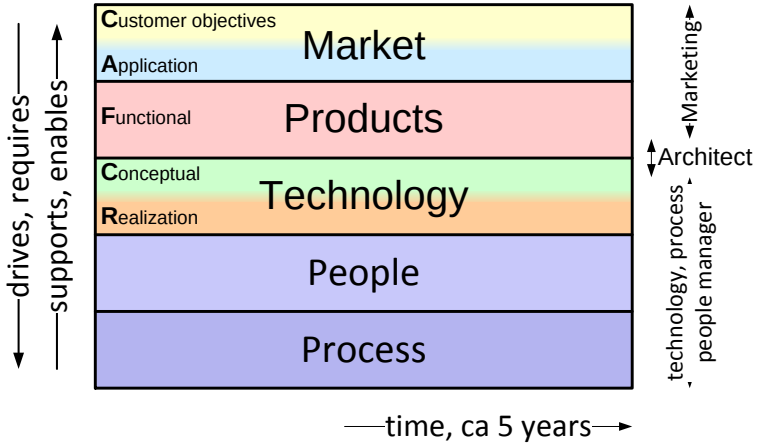


Exercise Roadmapping

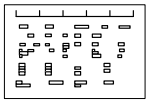
Make a roadmap on the basis of what you know at this moment, or what you perceive as the "shared expectation".
Try to fill in as many views (market, products, technology, people and process) as possible.
Present an overview by minimizing the contents to the most essential data.

Roadmap Creation

The Roadmap Integrates Five Views



Multiple Levels

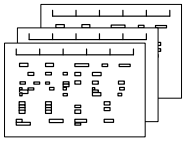


Top-level roadmap

Single page

Poster

part of many presentations



Supporting roadmaps

Single page per view or per driver

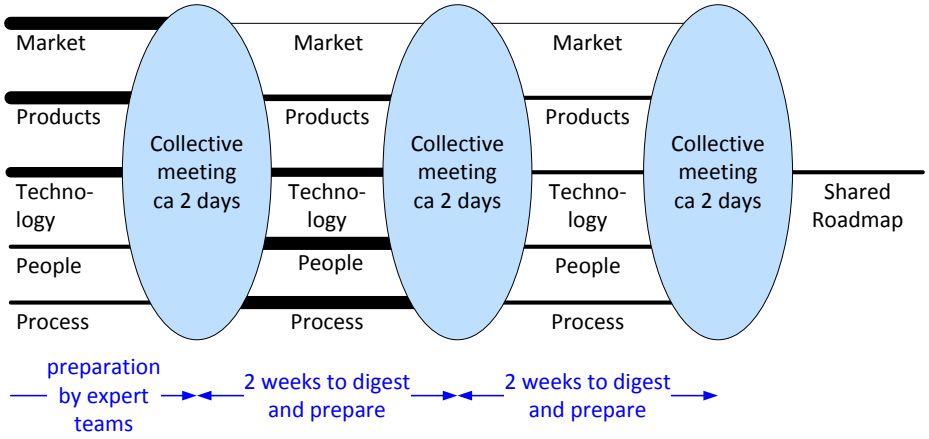
Poster

part of many presentations

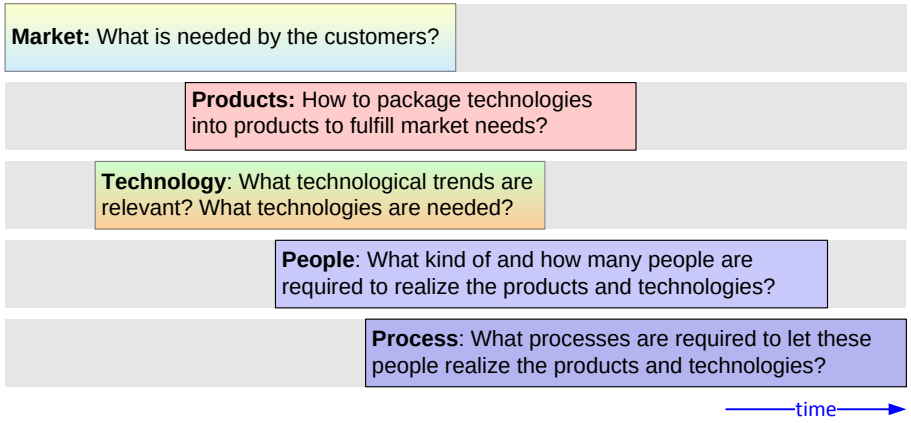
Supporting reports

Document per relevant subject

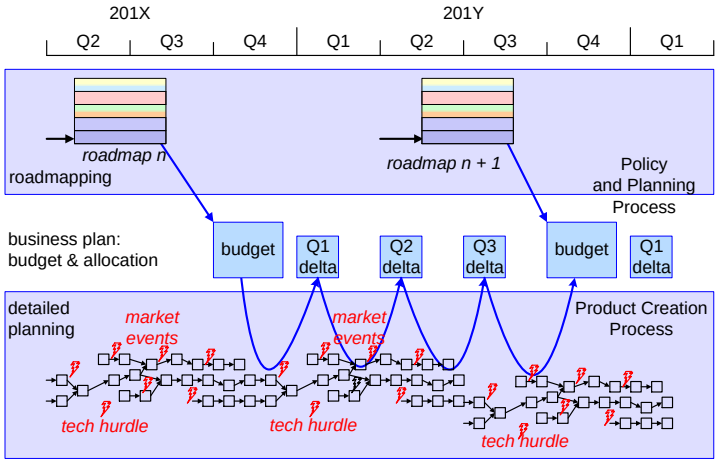
Creation in Teams



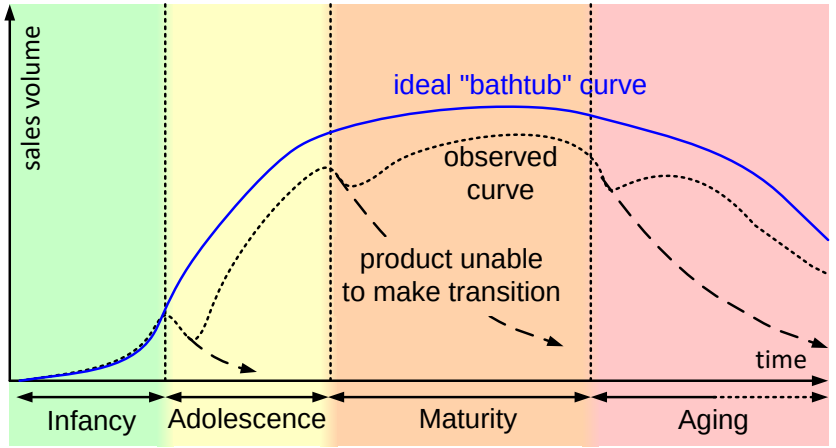
Order of Creation



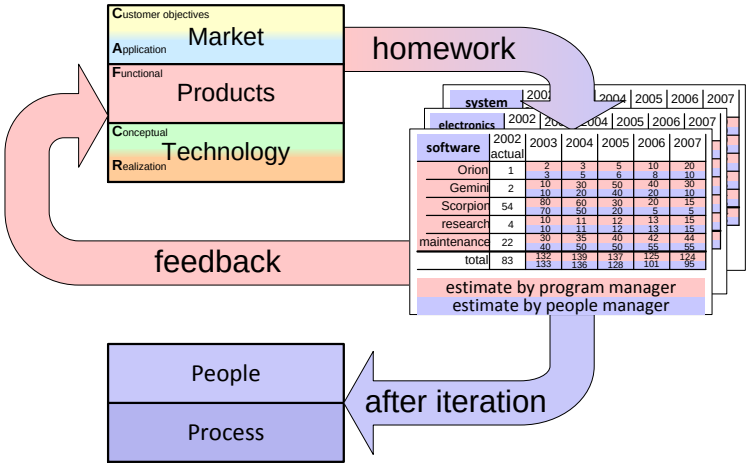
Time Horizons



Life Cycle Transitions



People and Process



intentionally left blank

Module Product Families and Generic Developments

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses product families and generic developments.

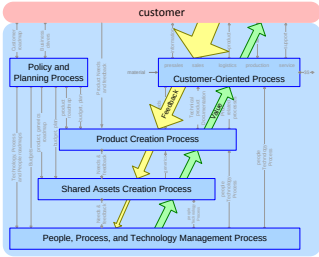
January 22, 2023
status: preliminary draft
version: 1.3

Product Families and Generic Aspects

by *Gerrit Muller* USN-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

Most products fit in a larger family of products. The members of such a product family share a lot of functionality and features. It is attractive to share implementations, designs et cetera between those members to increase the efficiency of the entire company.
In practice many difficulties pop up when product developments become coupled, due to the partial developments which are shared. This article discusses the advantages and disadvantages of a family approach based on shared developments and provides some methods to increase the chance on success.



January 22, 2023
status: concept
version: 2.3

Typical Examples of Generic Developments

Platform

Common components

Standard design

Framework

Family architecture

Generic aspects, functions, or features

Reuse

Products (in project environment)

Claimed Advantages of Generic Developments

Reduced time to market — building on shared components

Reduced cost per function — build every function only once

Improved quality
Improved reliability
Improved predictability
— maturing realization

Easier diversity management — modularity

Increases uniformity
Employees only have to understand one base system
— less learning

Larger purchasing power — economy of scale

Means to consolidate knowledge

Increase added value — not reinventing existing functionality

Enables parallel developments of multiple products

“Free” feature propagation — product-to-product or project-to-project

Experiences with reuse, from counterproductive to effective

bad	good
longer time to market	reduced time to market
high investments	reduced investment
lots of maintenance	reduced (shared) maintenance cost
poor quality	improved quality
poor reliability	improved reliability
diversity is opposed	easier diversity management
lot of know how required	understanding of one base system
predictable too late	improved predictability
dependability	larger purchasing power
knowledge dilution	means to consolidate knowledge
lack of market focus	increase added value
interference	enables parallel developments
but integration required	free feature propagation

Successful examples of reuse

homogeneous domain	cath lab MRI television waferstepper
hardware dominated	car airplane shaver television
limited scope	audio codec compression library streaming library

Limits of successful reuse

struggle with integration/convergence with other domains

TV: digital networks and media
cath lab: US imaging, MRI

poor/slow response on paradigm shifts

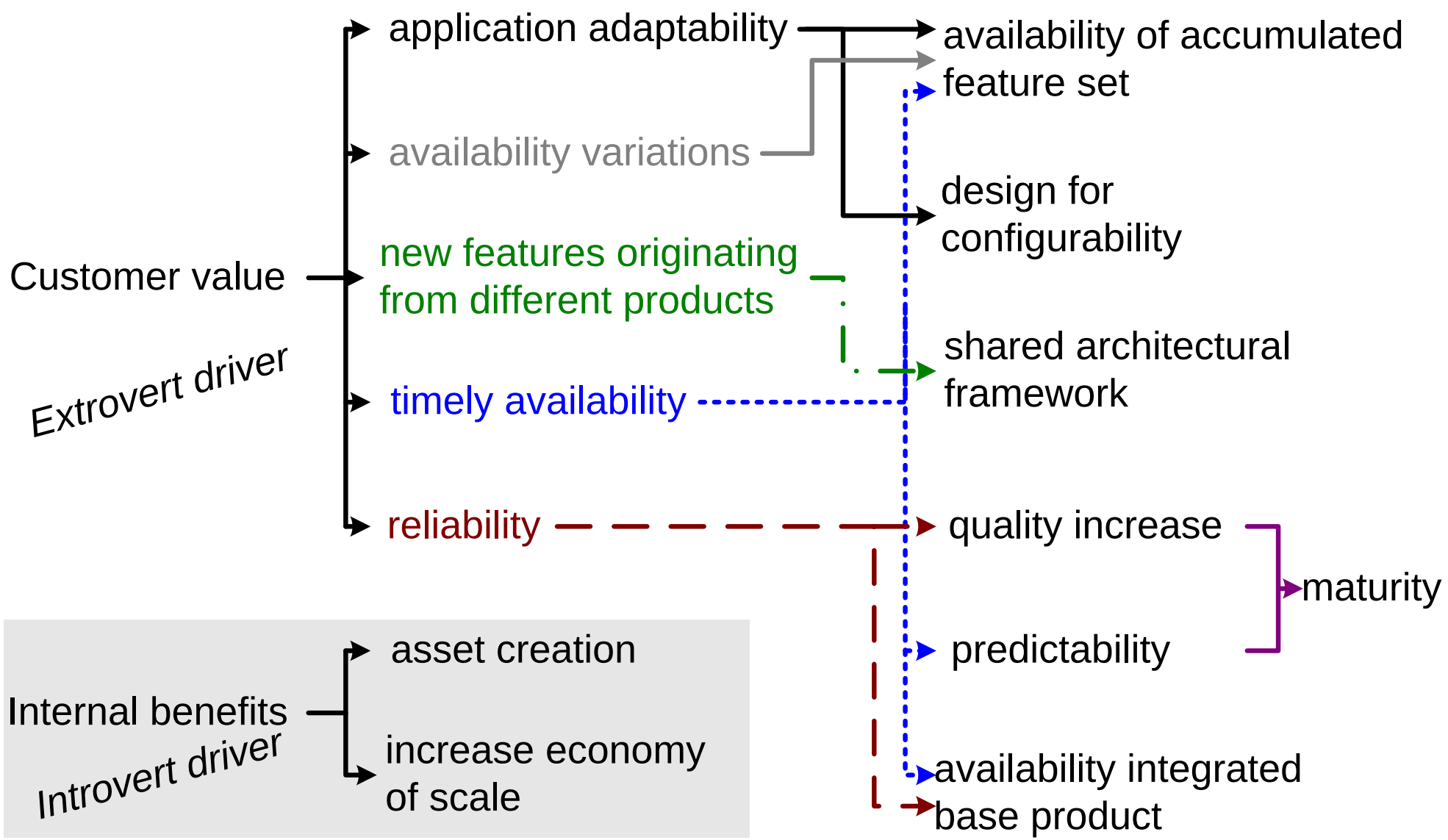
TV: LCD screens
cath lab: image based acquisition control

software maintenance, configurations, integration, release

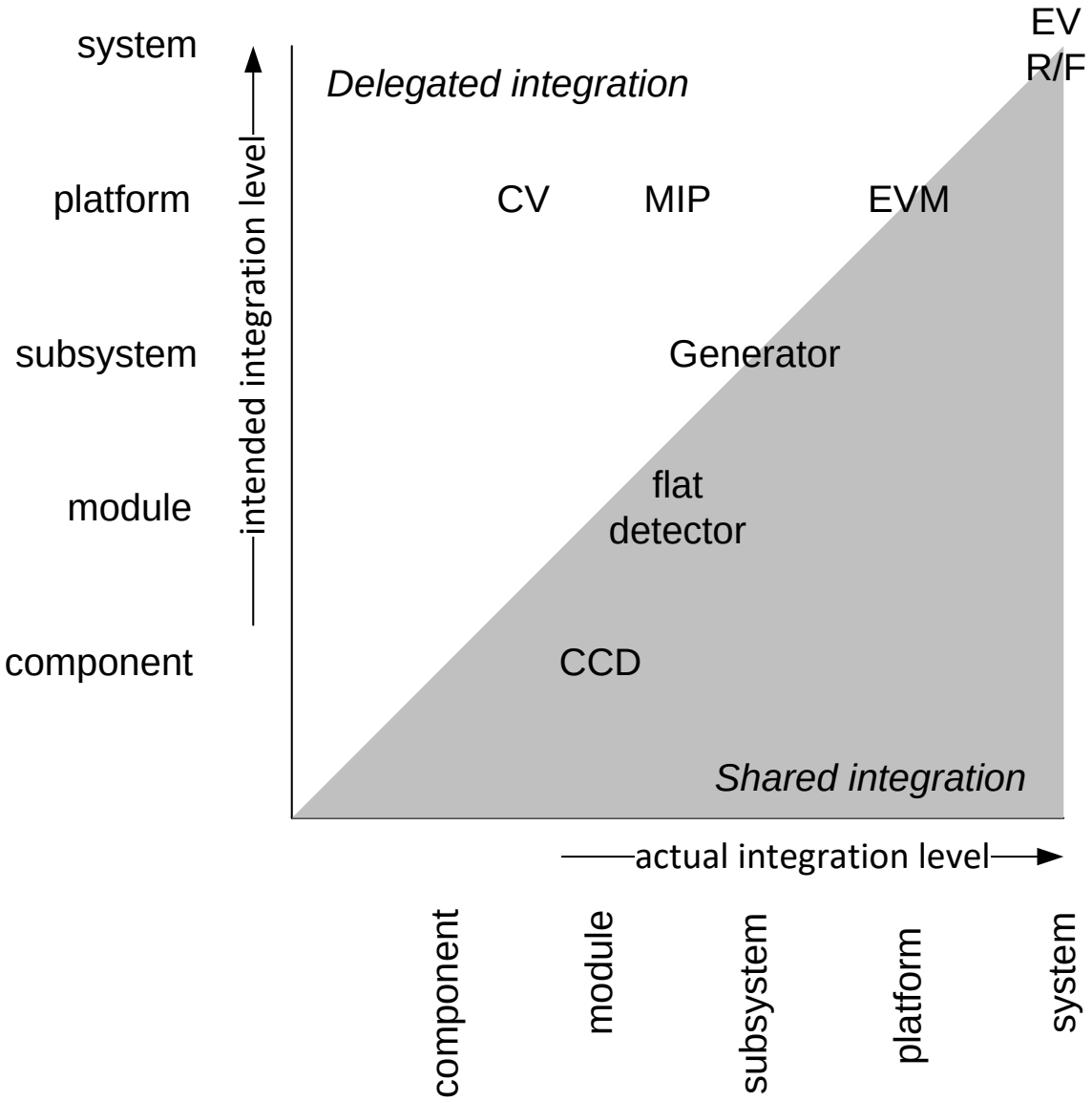
MRI: integration and test
wafersteppers: number of configurations

how to innovate?

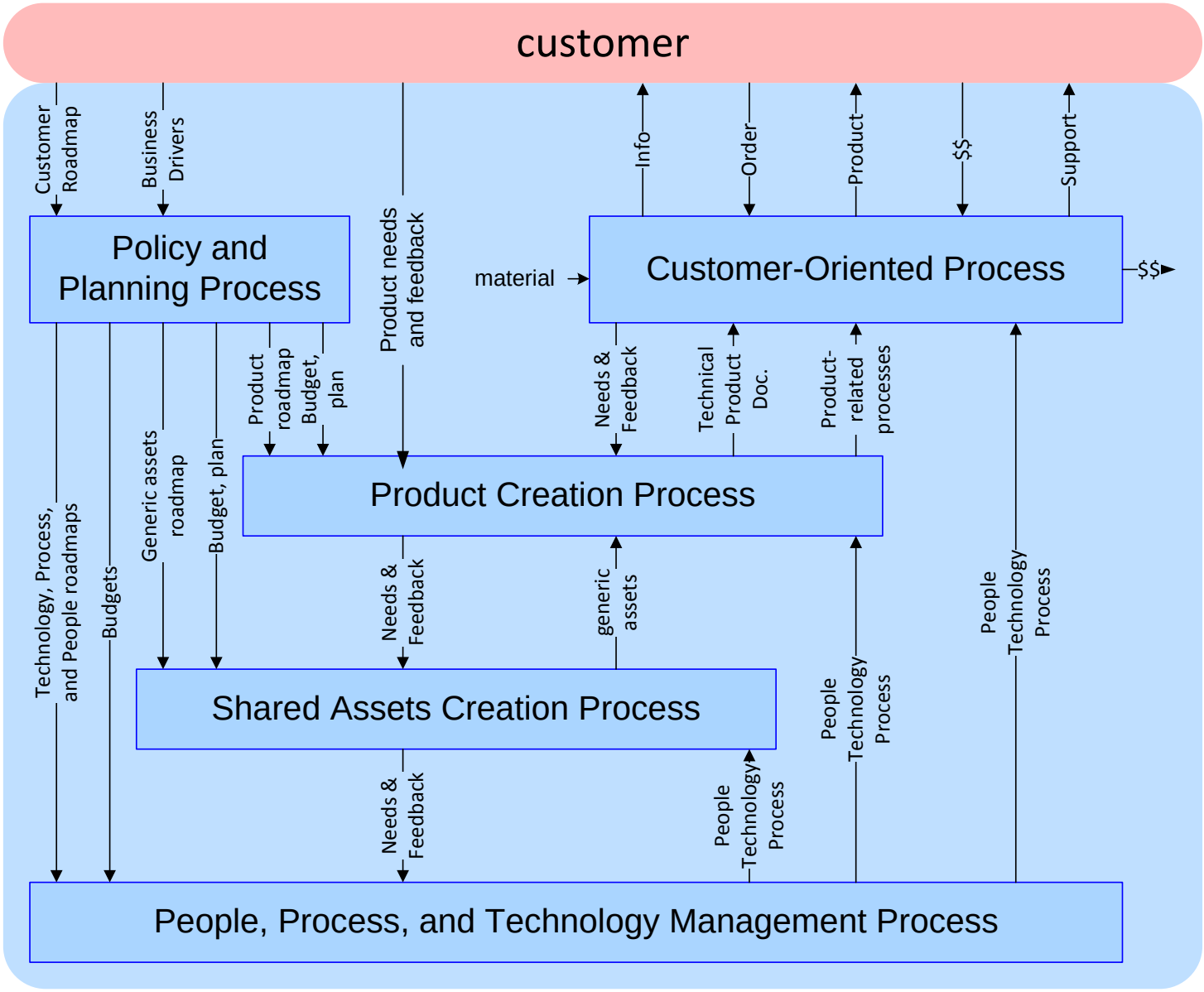
Drivers for Generic Developments



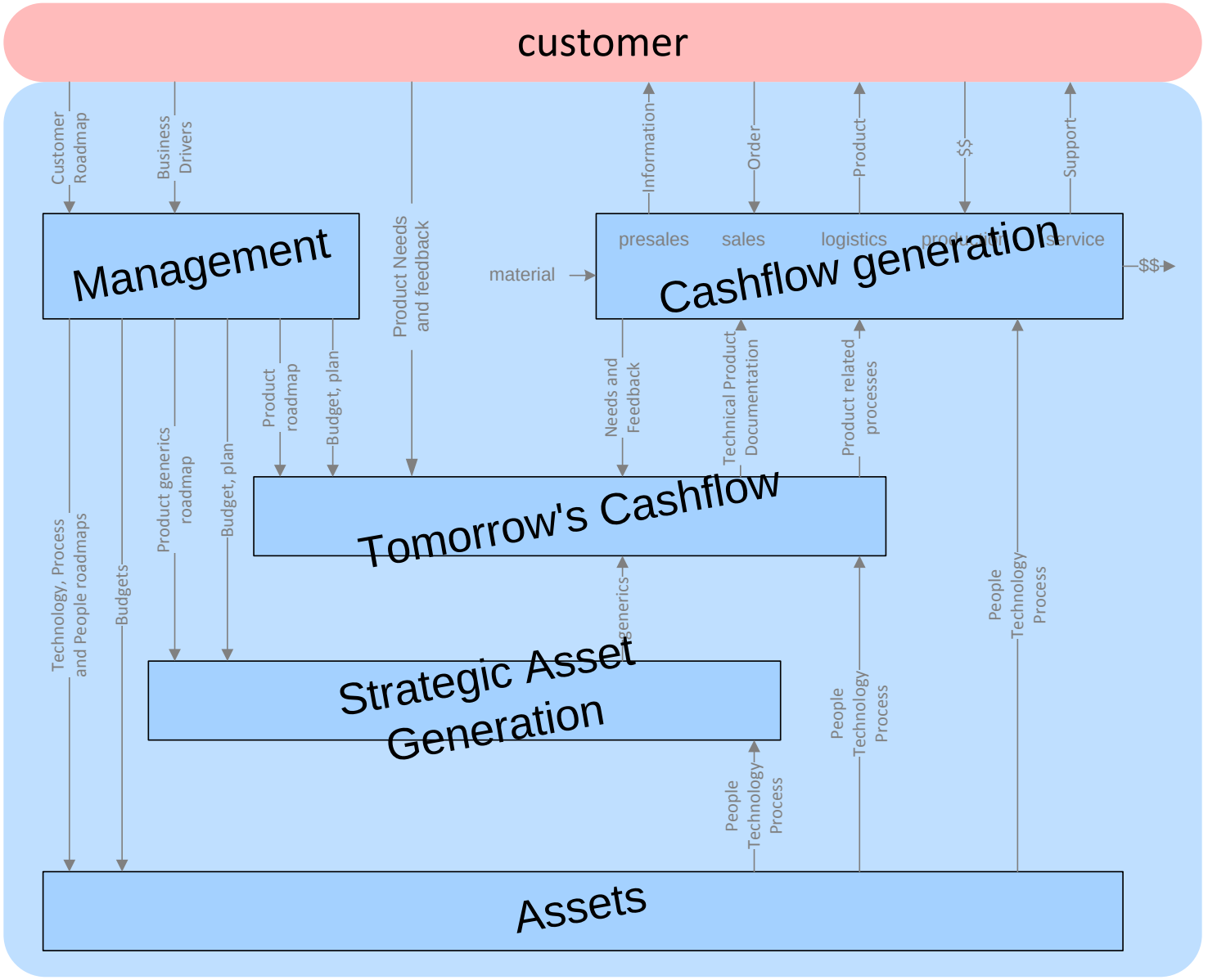
Granularity of generic developments shown in 2 dimensions



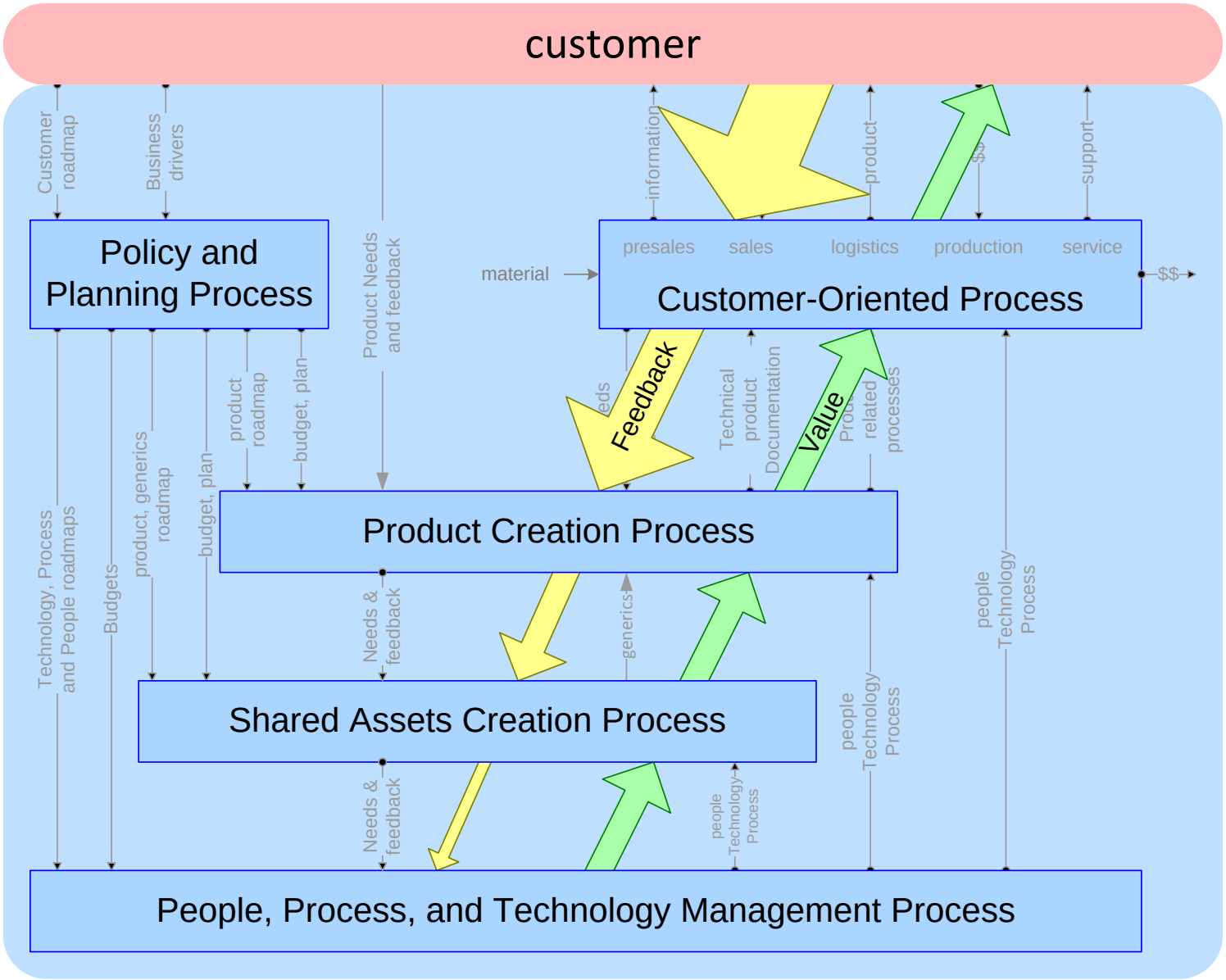
Modified Process Decomposition



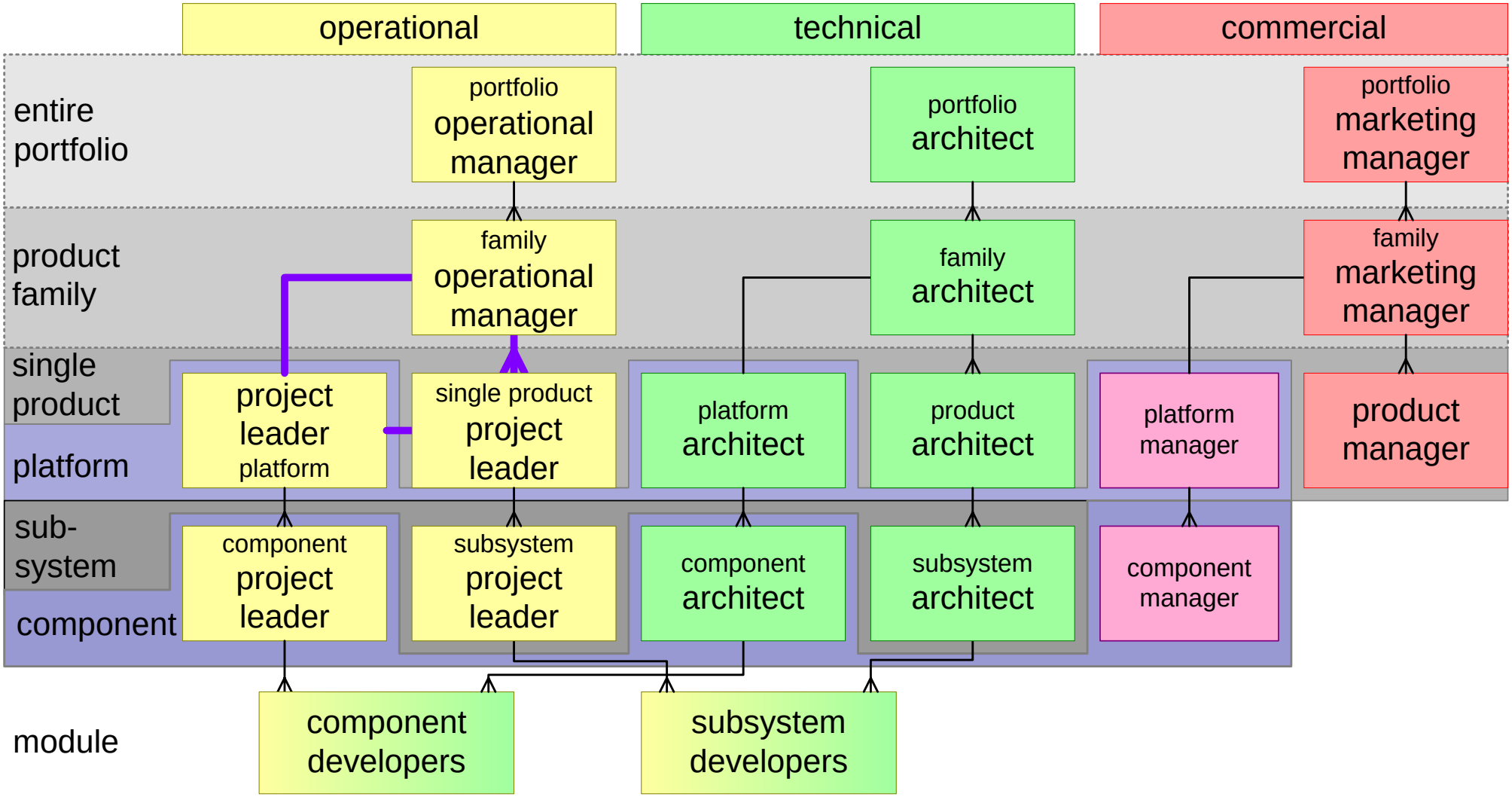
Financial Viewpoint on Process Decomposition



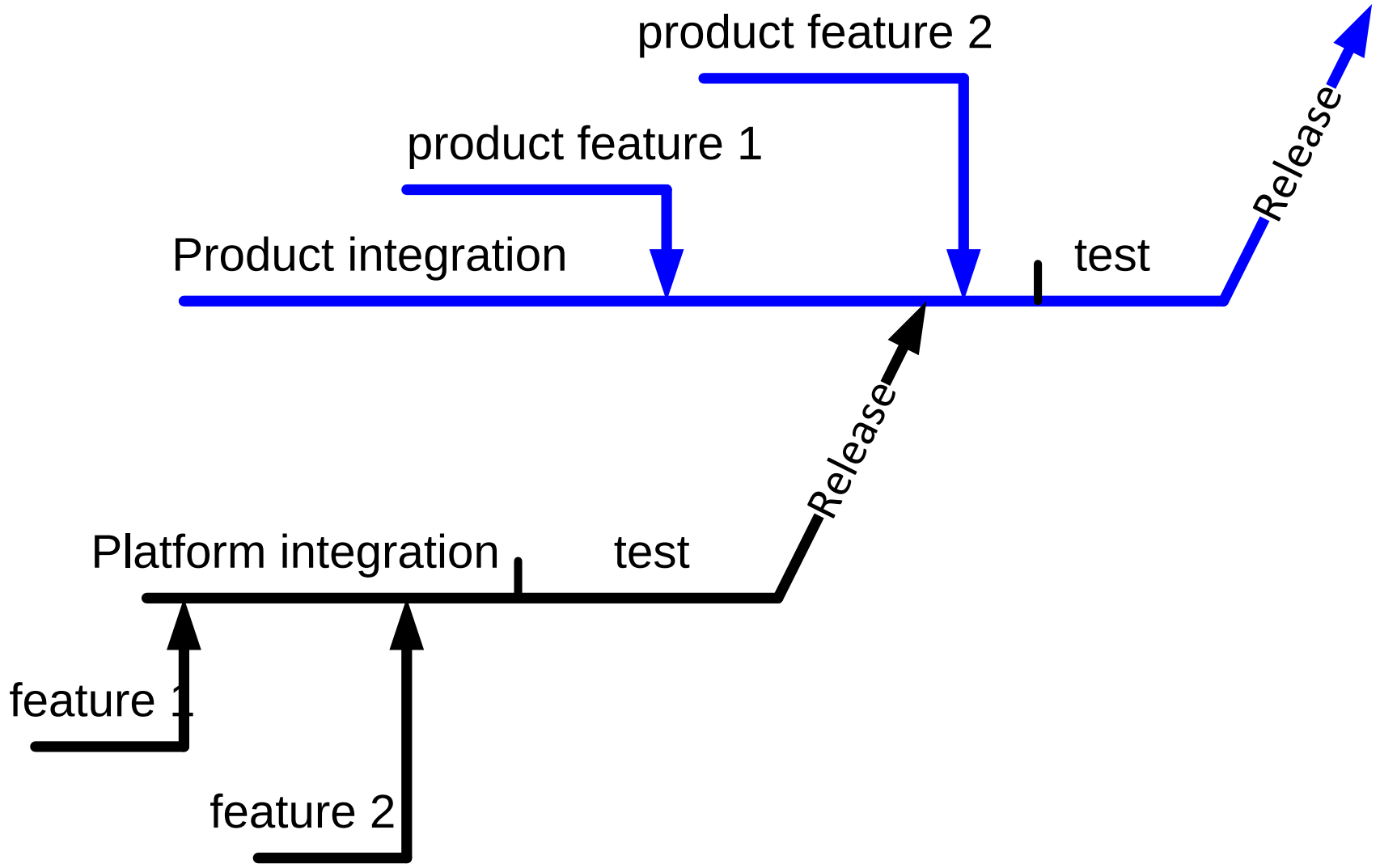
Value and Feedback Flow



Modified Operational Organization PCP



Propagation Delay Platform Feature to Market



Sources of Failure in Generic Developments

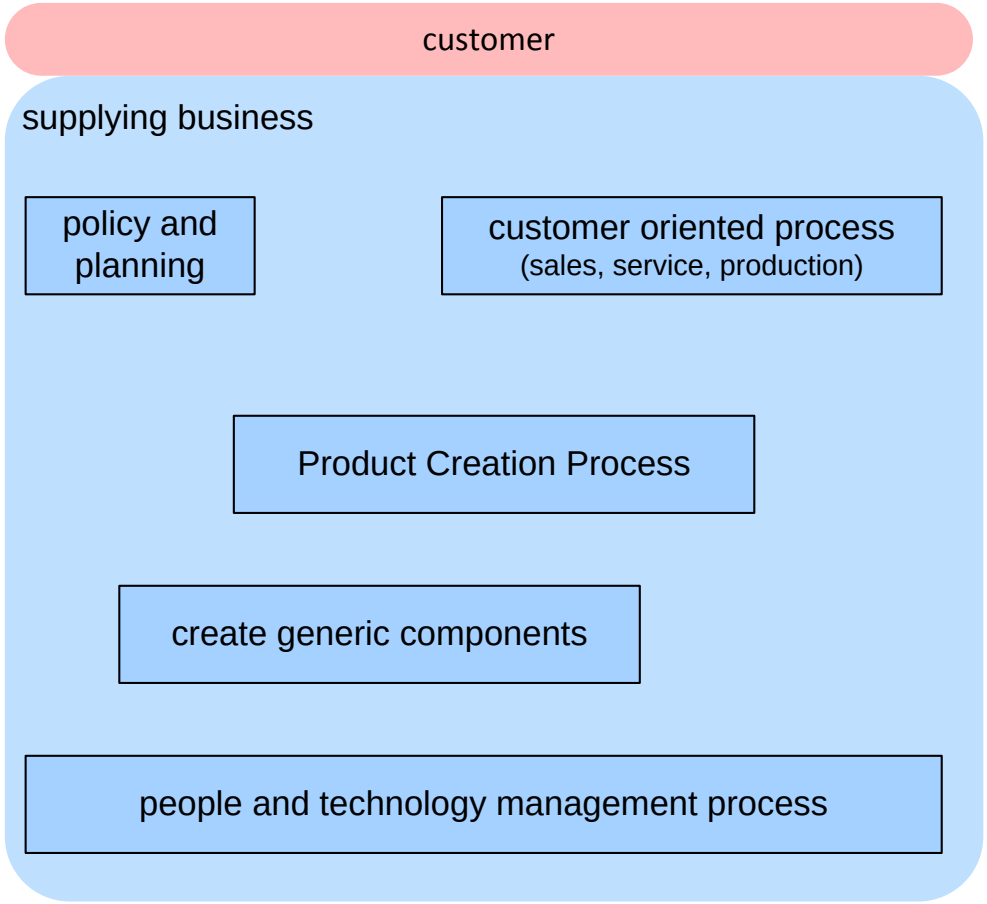
Technical

- Too generic
- Innovation stops (stable interfaces)
- Vulnerability

Process/People/Organization

- Forced cooperation
- Time platform feature to market
- Unrealistic expectations
- Distance platform developer to customer
- No marketing ownership
- Bureaucratic process (no flexibility)
- New employees, knowledge dilution
- Underestimation of platform support
- Overstretching of product scope
- Nonmanagement, organizational scope increase
- Underestimation of integration
- Component/platform determines business policy
- Subcritical investment

Models for Generic Development



lead customer — direct feedback, too specific?

carrier product — product feedback, product specific?

platform — feedback problem, too generic

technology push — no feedback

Exercise Generic Developments

What are the top 3 benefits for your product family or generic development?

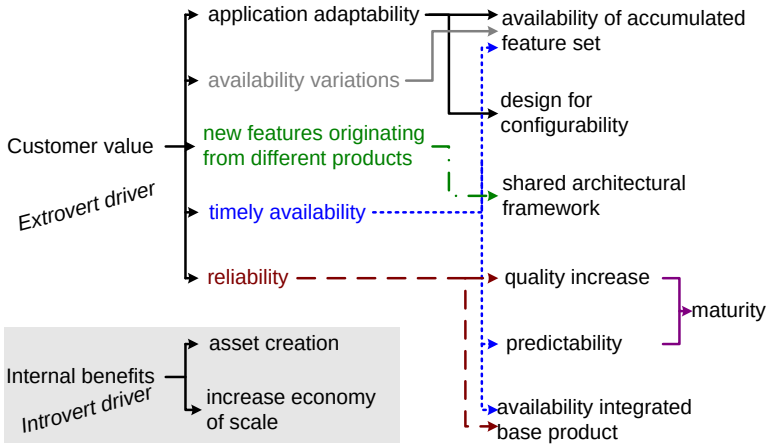
What are the top 3 disadvantages?

Harvesting Synergy

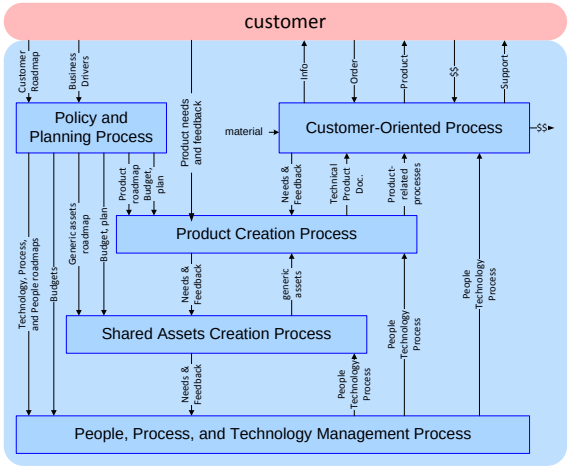
Contradicting Experiences

bad	good
longer time to market	reduced time to market
high investments	reduced investment
lots of maintenance	reduced (shared) maintenance cost
poor quality	improved quality
poor reliability	improved reliability
diversity is opposed	easier diversity management
lot of know how required	understanding of one base system
predictable too late	improved predictability
dependability	larger purchasing power
knowledge dilution	means to consolidate knowledge
lack of market focus	increase added value
interference	enables parallel developments
but integration required	free feature propagation

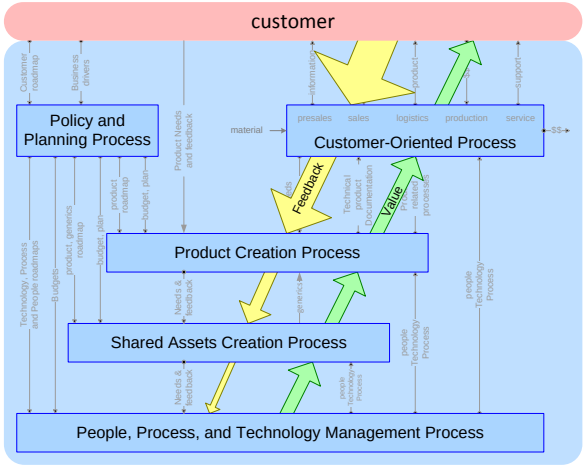
Drivers



Shared Asset Creation Process

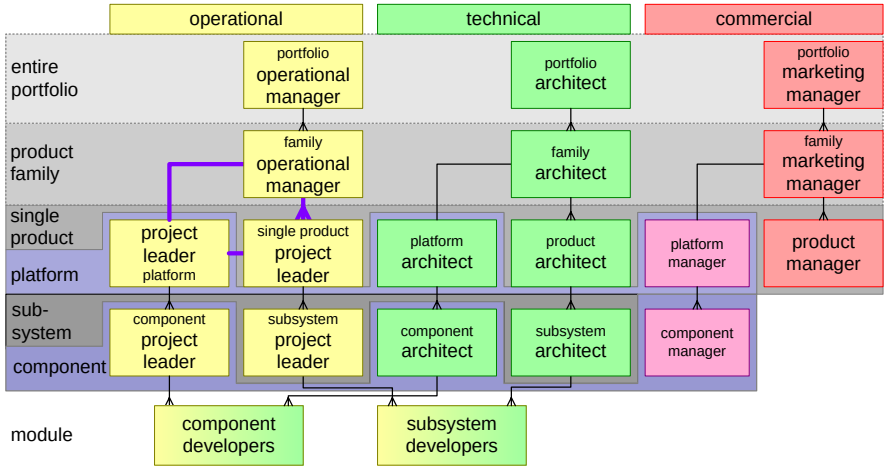


Longer Chains

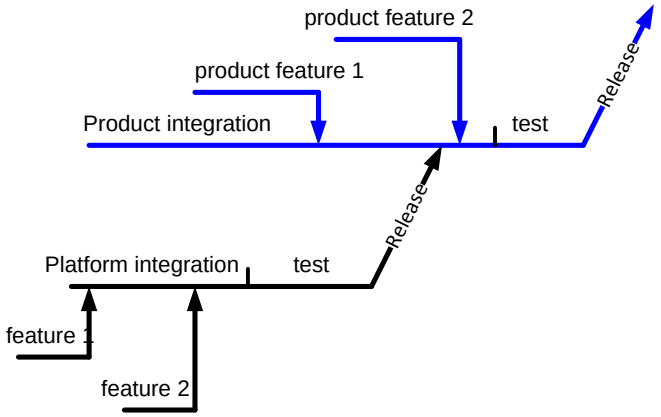


Some Architecting Means

Organizational Complexity



Delay to Market



Pitfalls

Technical

- Too generic
- Innovation stops (stable interfaces)
- Vulnerability

Process/People/Organization

- Forced cooperation
- Time platform feature to market
- Unrealistic expectations
- Distance platform developer to customer
- No marketing ownership
- Bureaucratic process (no flexibility)
- New employees, knowledge dilution
- Underestimation of platform support
- Overstretching of product scope
- Nonmanagement, organizational scope increase
- Underestimation of integration
- Component/platform determines business policy
- Subcritical investment

Successful and Failing Models