

# All About Systems Engineering; Introductory Course

material by *Gerrit Muller*

presented by

## Abstract

This introductory course sketches all fundamentals of Systems Engineering. Starting at the business contexts, touching Project, Processes, and Organization. The role of the Systems Engineer is discussed, and the relation with other roles, e.g. project leader and product manager. The architecting and design tools are shown; from Stakeholder Needs to Requirements to Modeling and Analysis.

## Distribution

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

February 11, 2012  
status: preliminary  
draft  
version: 0.2

theory and cases		more theory and exercises	
day 1	introduction to SE, process, organization case, planning, V-Model, spiral model, relation with other business disciplines	day 3	organization and process in practice exercise and discussion product families, products vs projects
day 2	systems engineer role and task deliverables, responsibilities, activities, styles, characteristics	day 4	capturing customer understanding exercise and discussion customer key drivers, story telling, scenarios and use cases
	system context customer context, life cycle context, stakeholders, needs, concerns, requirements, story telling, concepts, use cases		creating the big picture exercise and discussion roadmap approach - key performance parameters
	system design concept selection, physical decomposition, functional decomposition, qualities, interface management, budgeting, modeling		design and concept selection in practice exercise and discussion example case to wrap-up

# Introduction Course Program

---

day 1	
<i>morning</i>	<i>afternoon</i>
<b>introduction to SE, process, organization</b> case, phasing, V-Model, spiral model, relation with other business disciplines	<b>systems engineer role and task</b> deliverables, responsibilities, activities, styles, characteristics
day 2	
<i>morning</i>	<i>afternoon</i>
<b>system context</b> customer context, life cycle context, stakeholders, needs, concerns, requirements, story telling, conops, use cases	<b>system design</b> concept selection, physical decomposition, functional decomposition, qualities, interface management, budgeting, modeling

# Project Systems Engineering Introduction; Phasing, Process, Organization

by *Gerrit Muller* Buskerud University College

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

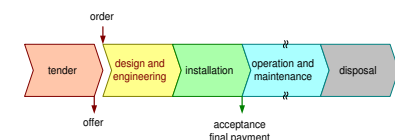
## Abstract

The fundamental concepts and approach to project oriented Systems Engineering are explained. We look at project phasing, phase transition, processes, and organization.

### Distribution

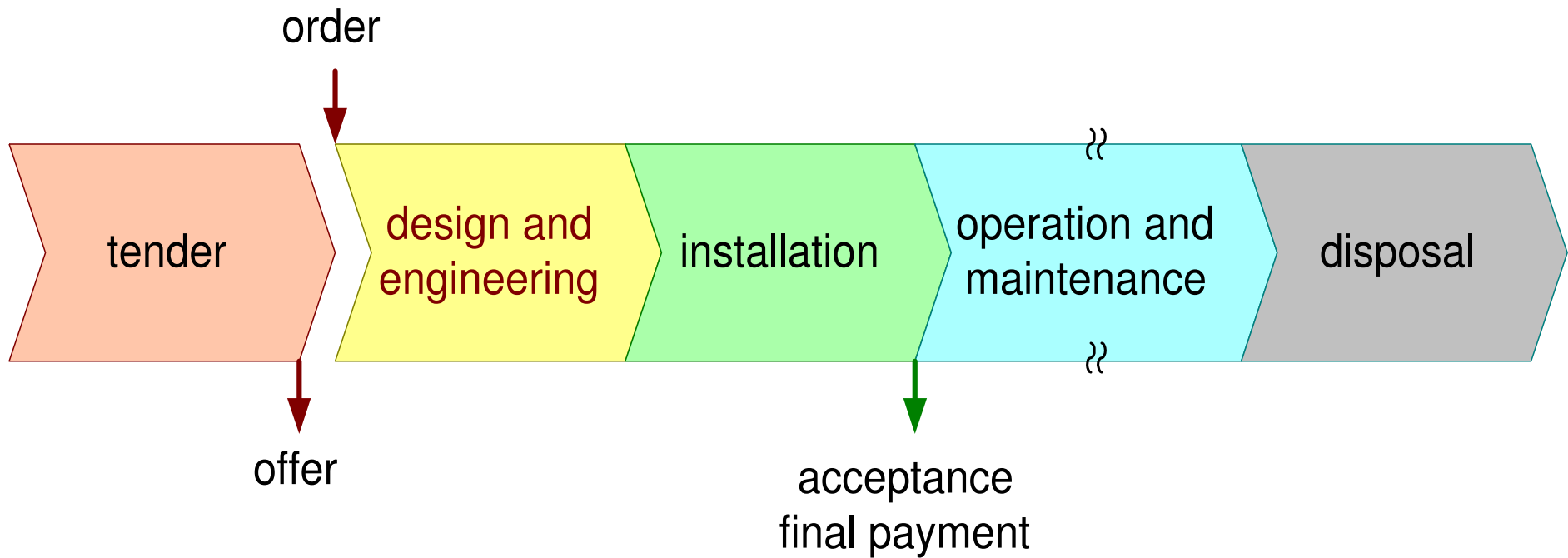
This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution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

February 11, 2012  
status: preliminary  
draft  
version: 0.1

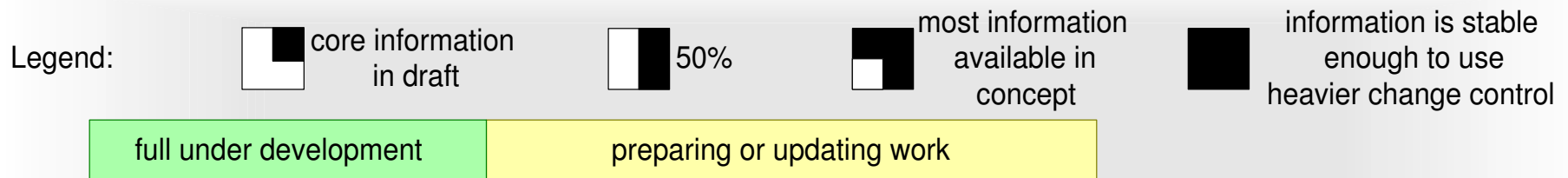
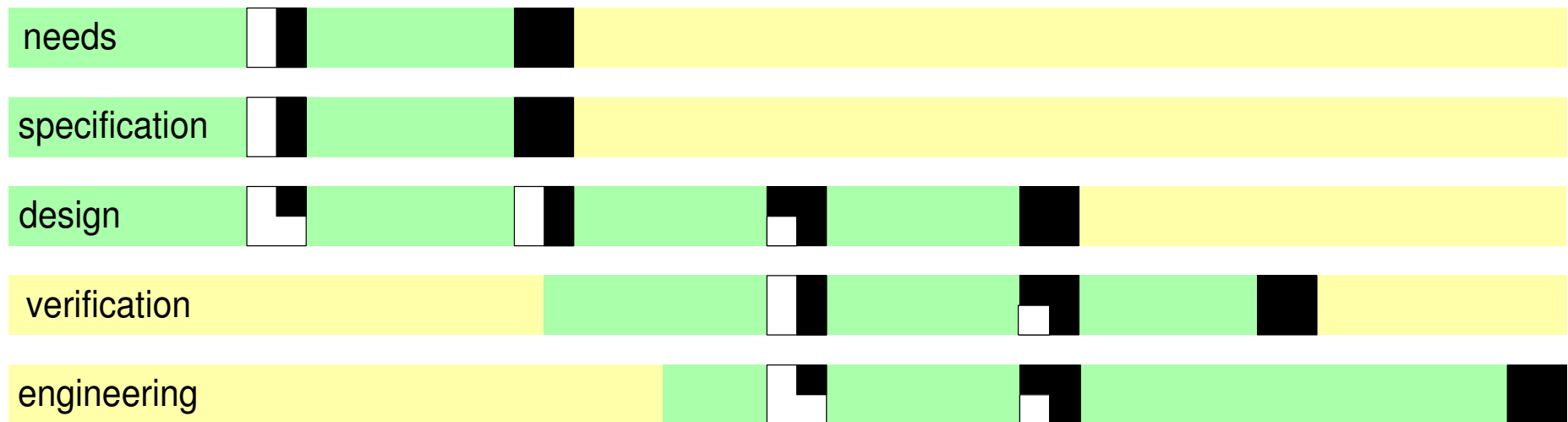
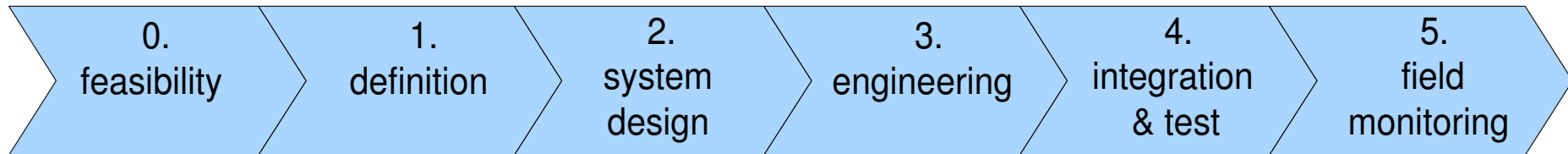
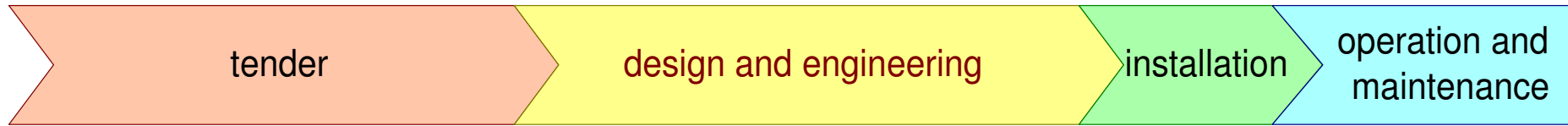


# Project Life Cycle

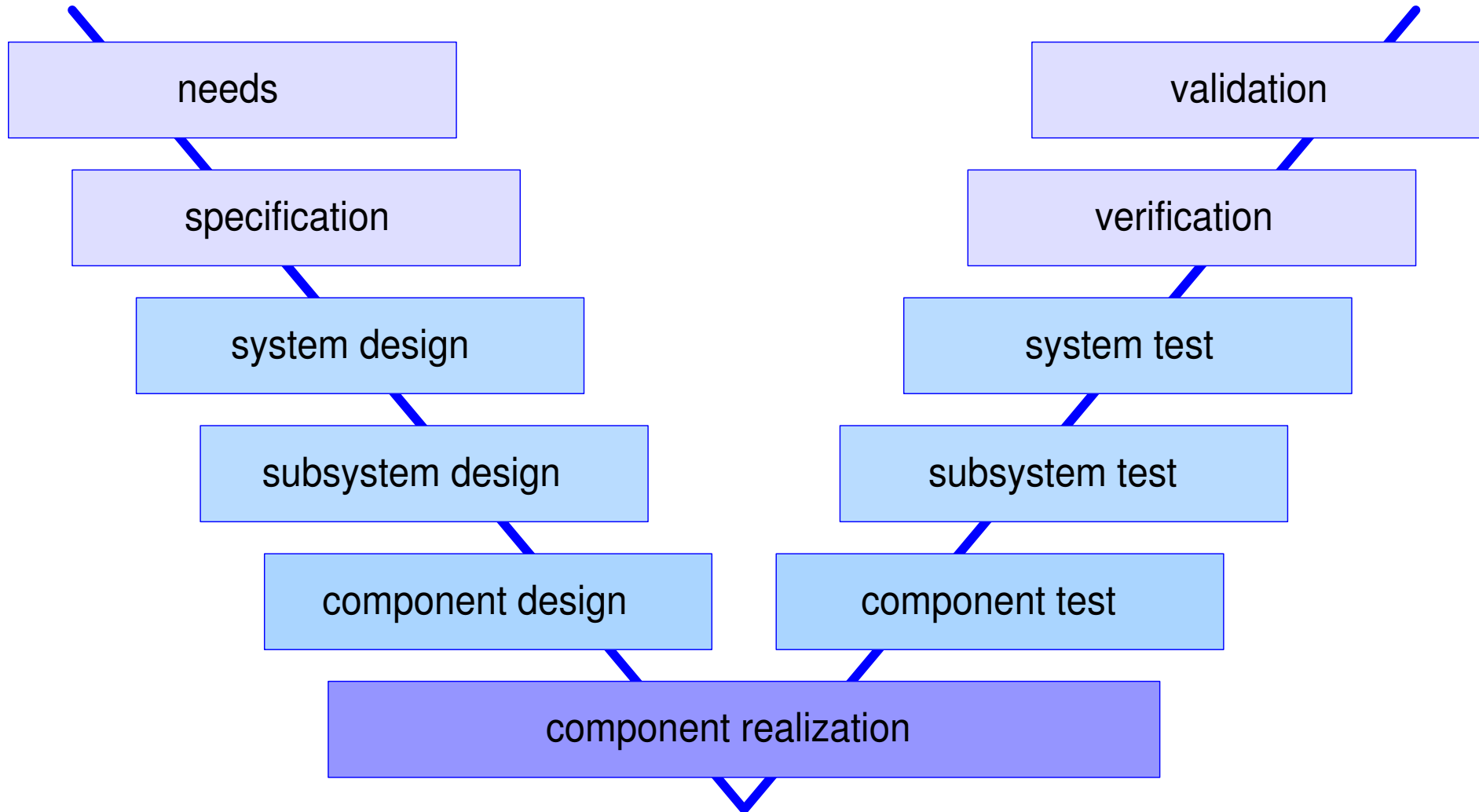
---



# Phased Project Approach

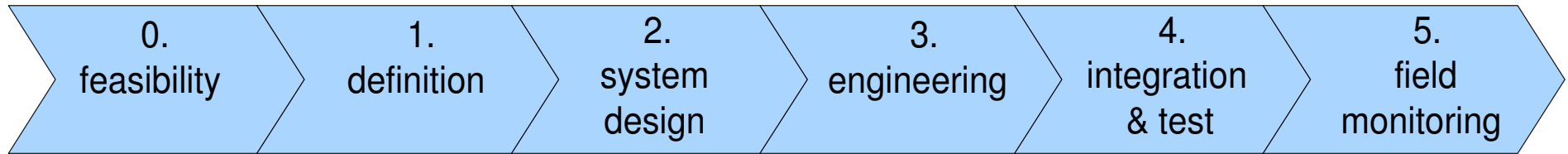


# V-Model



# All Business Functions Participate

---



sales

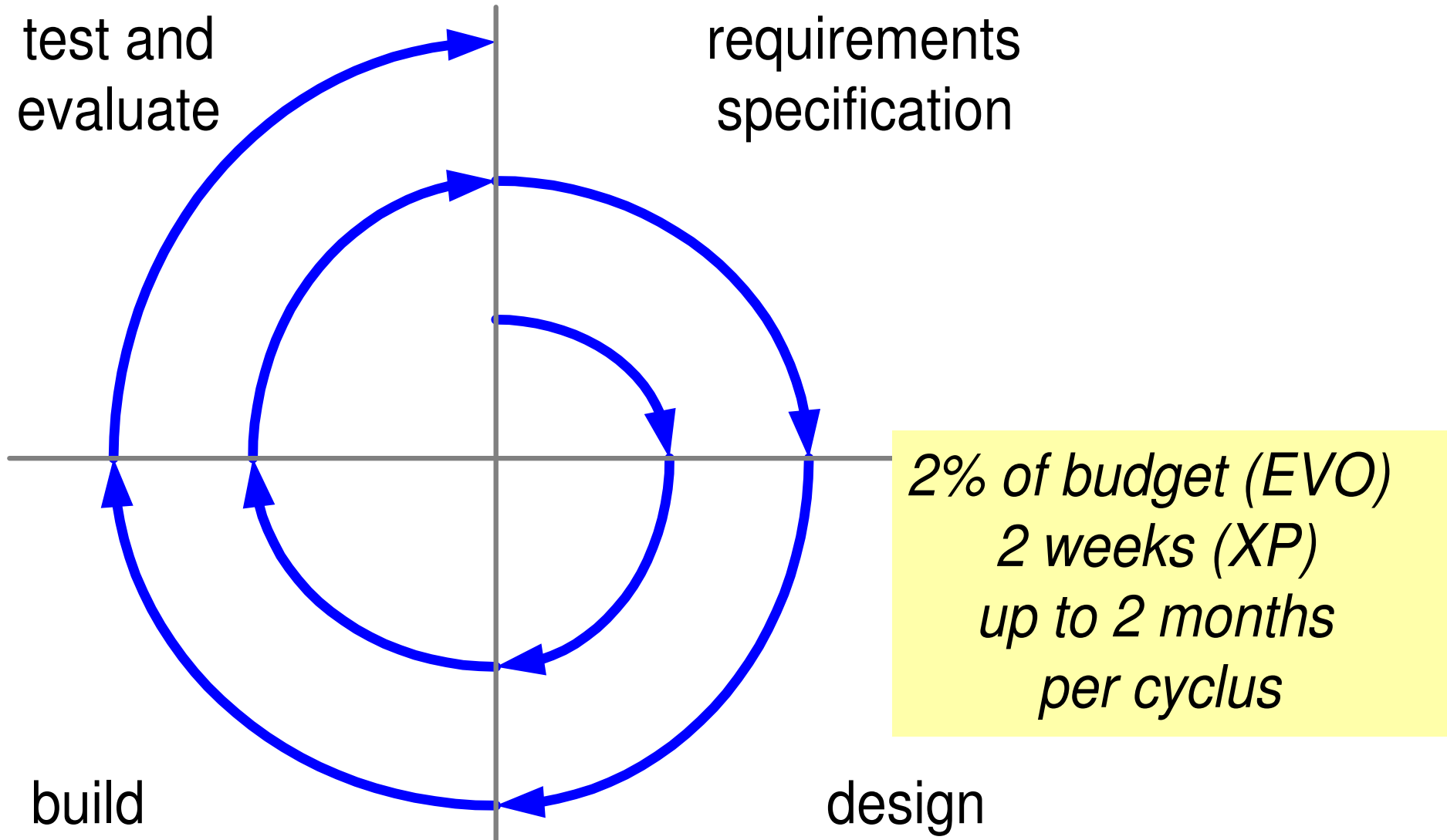
logistics

production

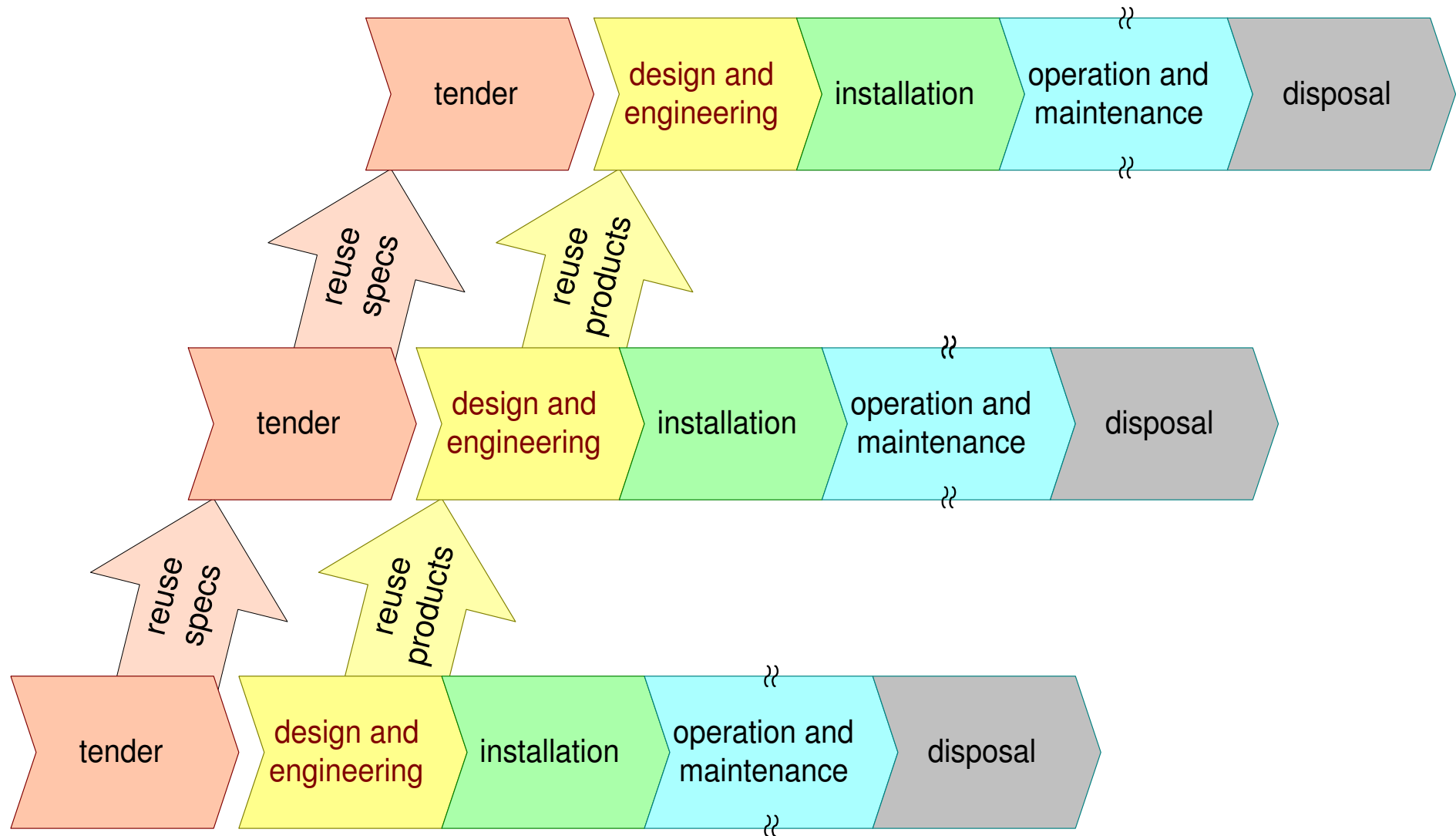
service

development & engineering: marketing, project management, design

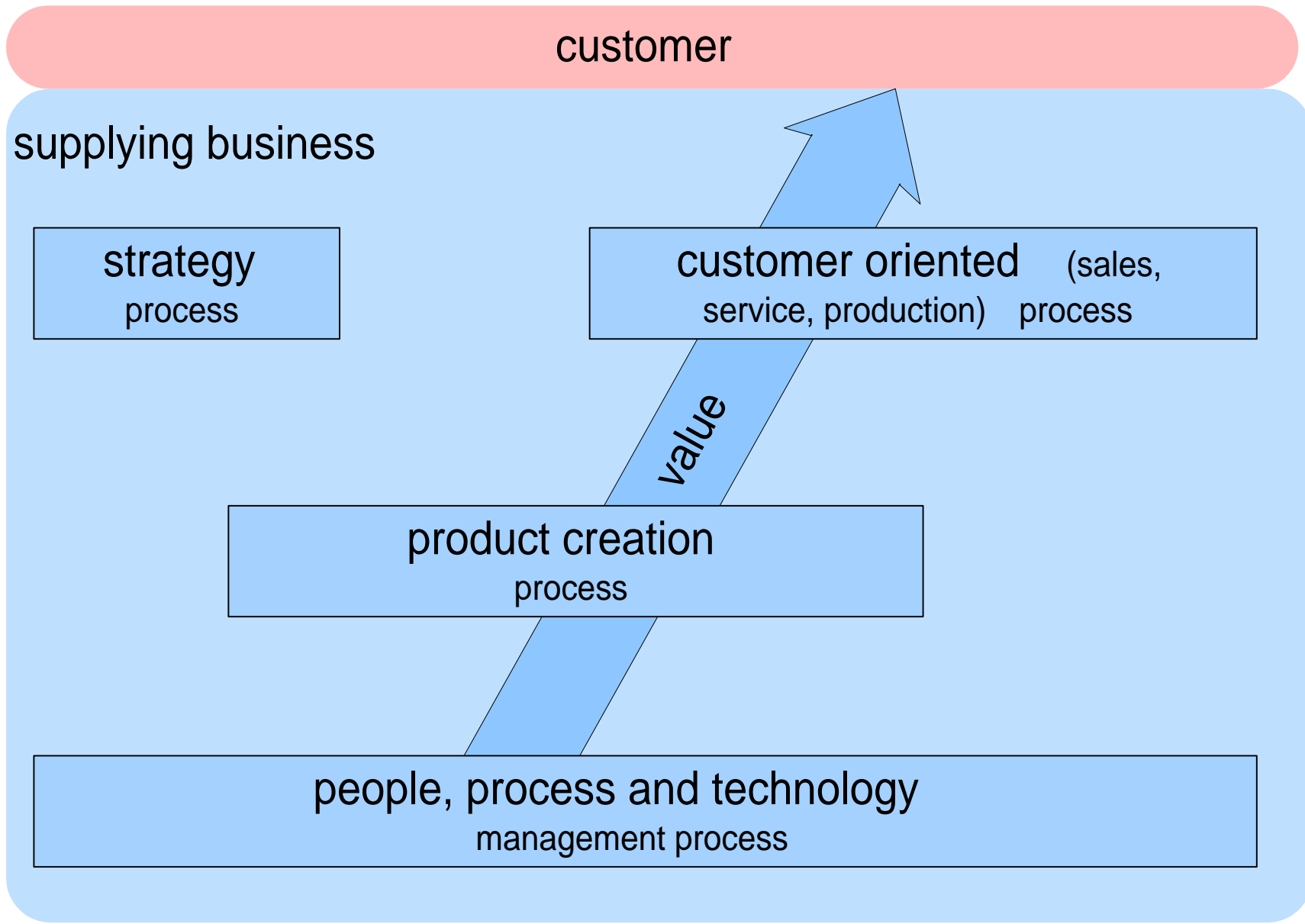
# Evolutionary PCP model



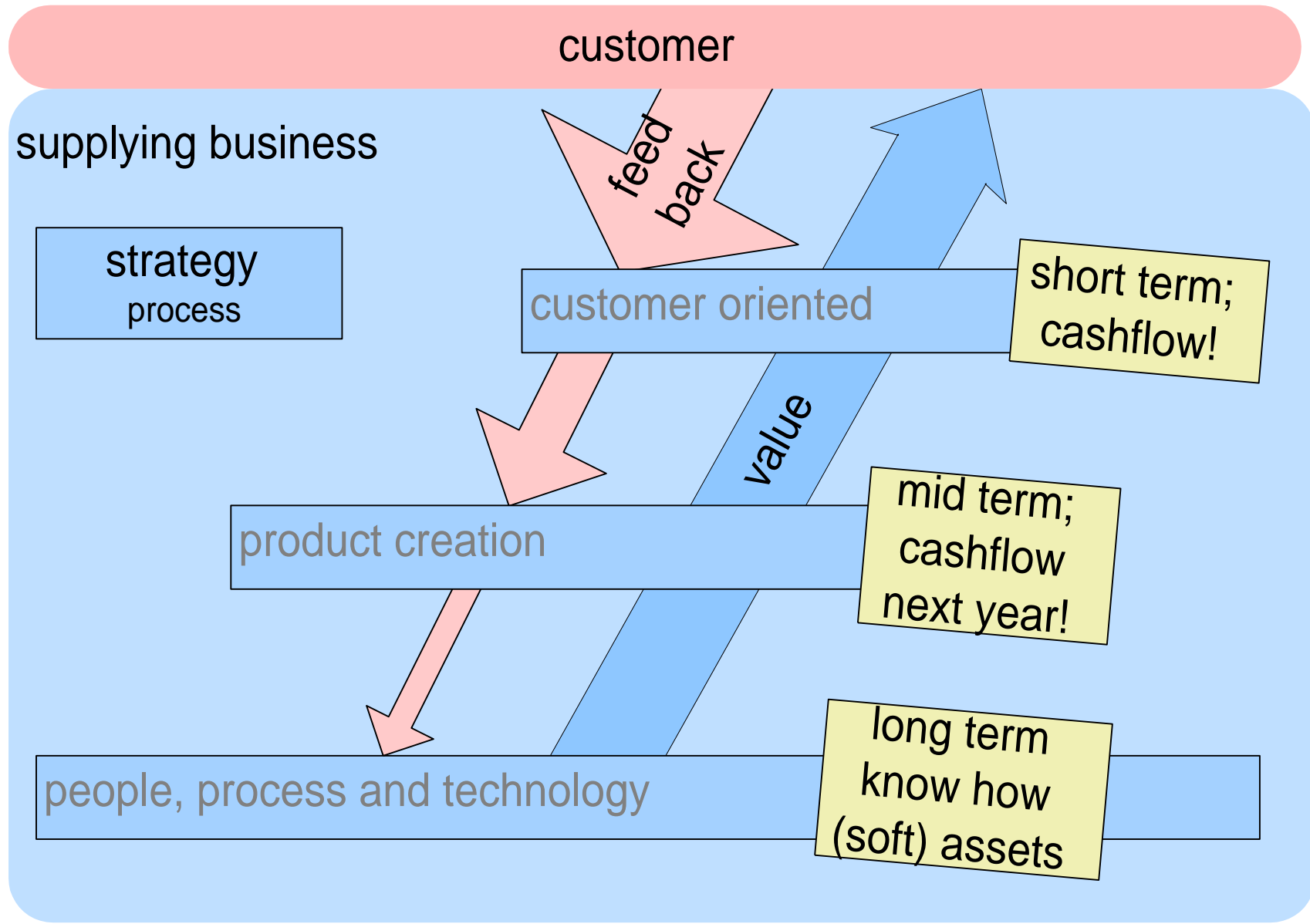
# Reuse and Products



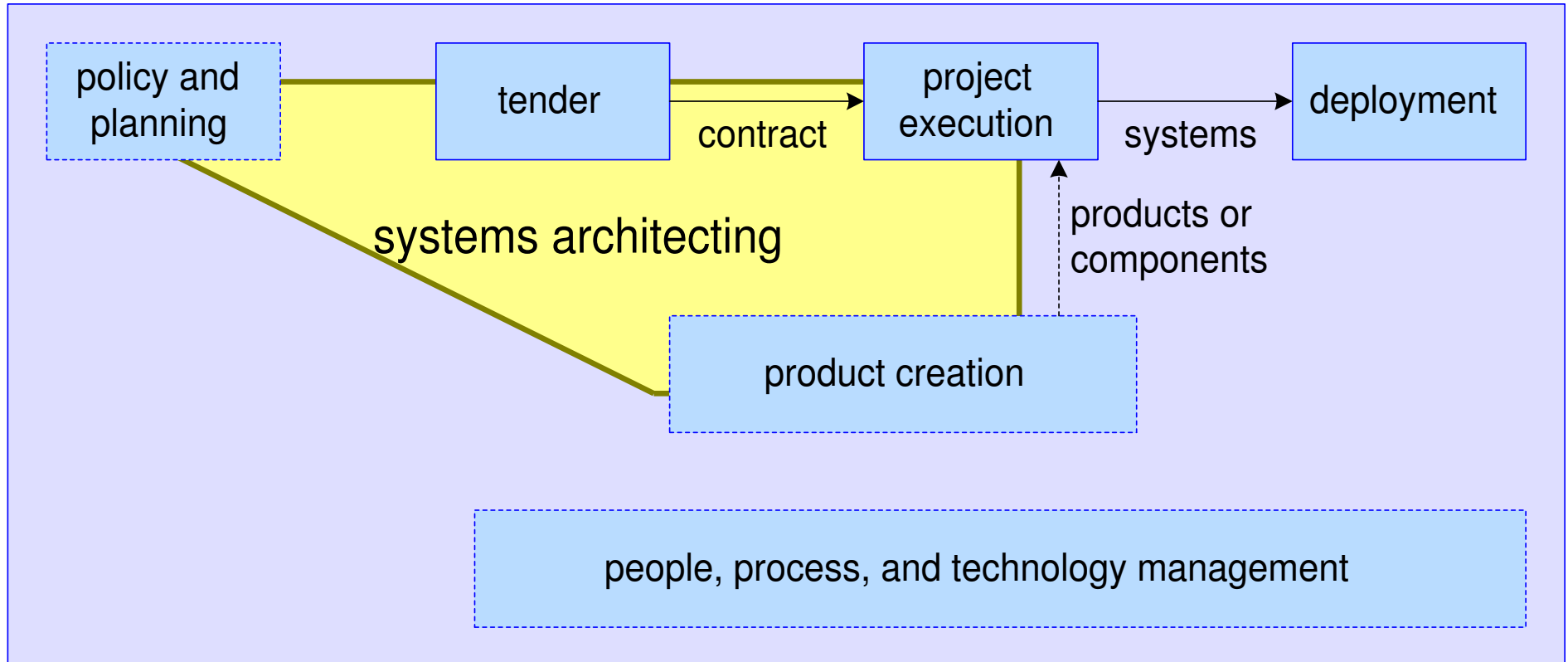
# Simplified Process View



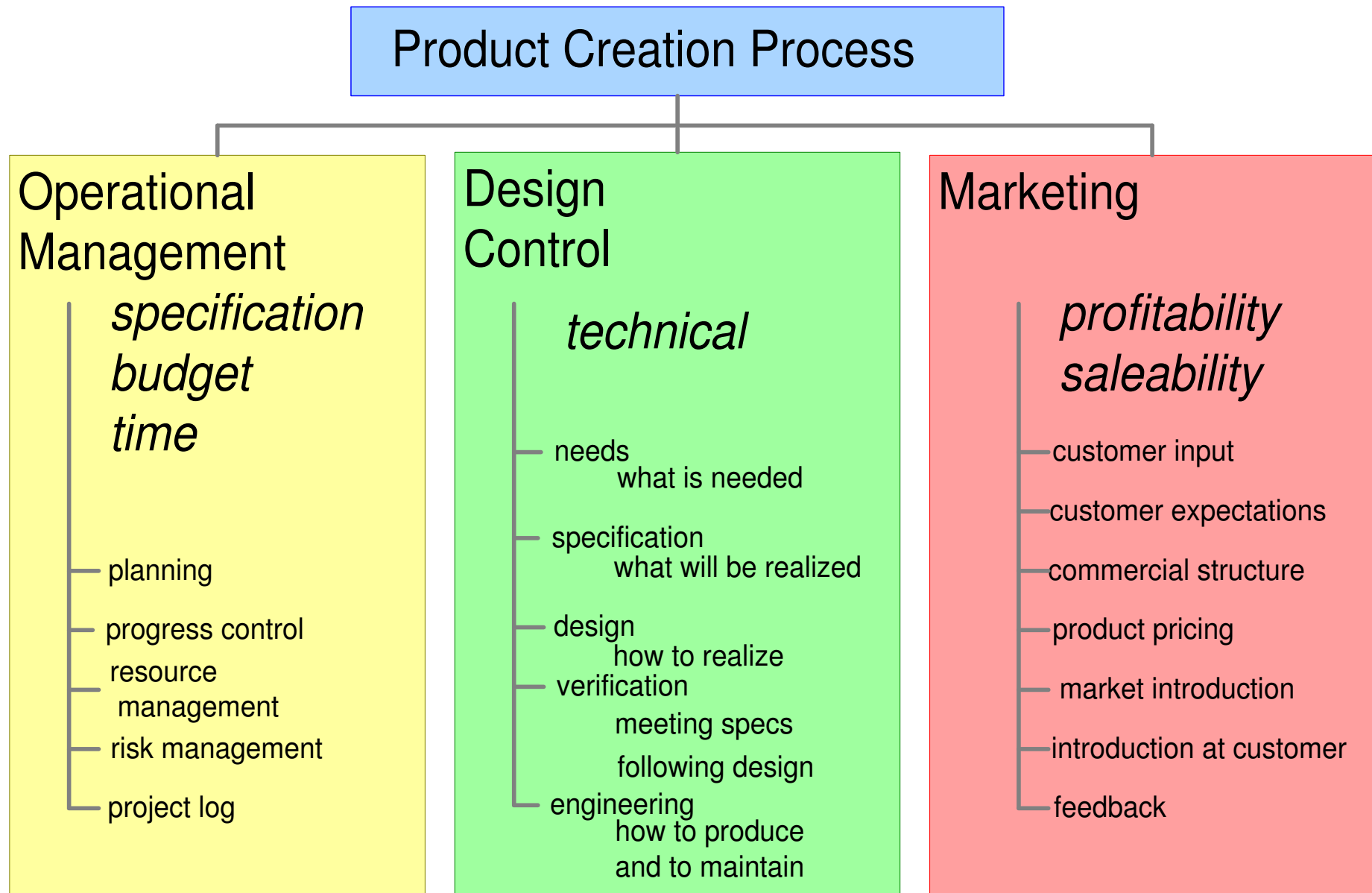
# Simplified Process; Money and Feedback



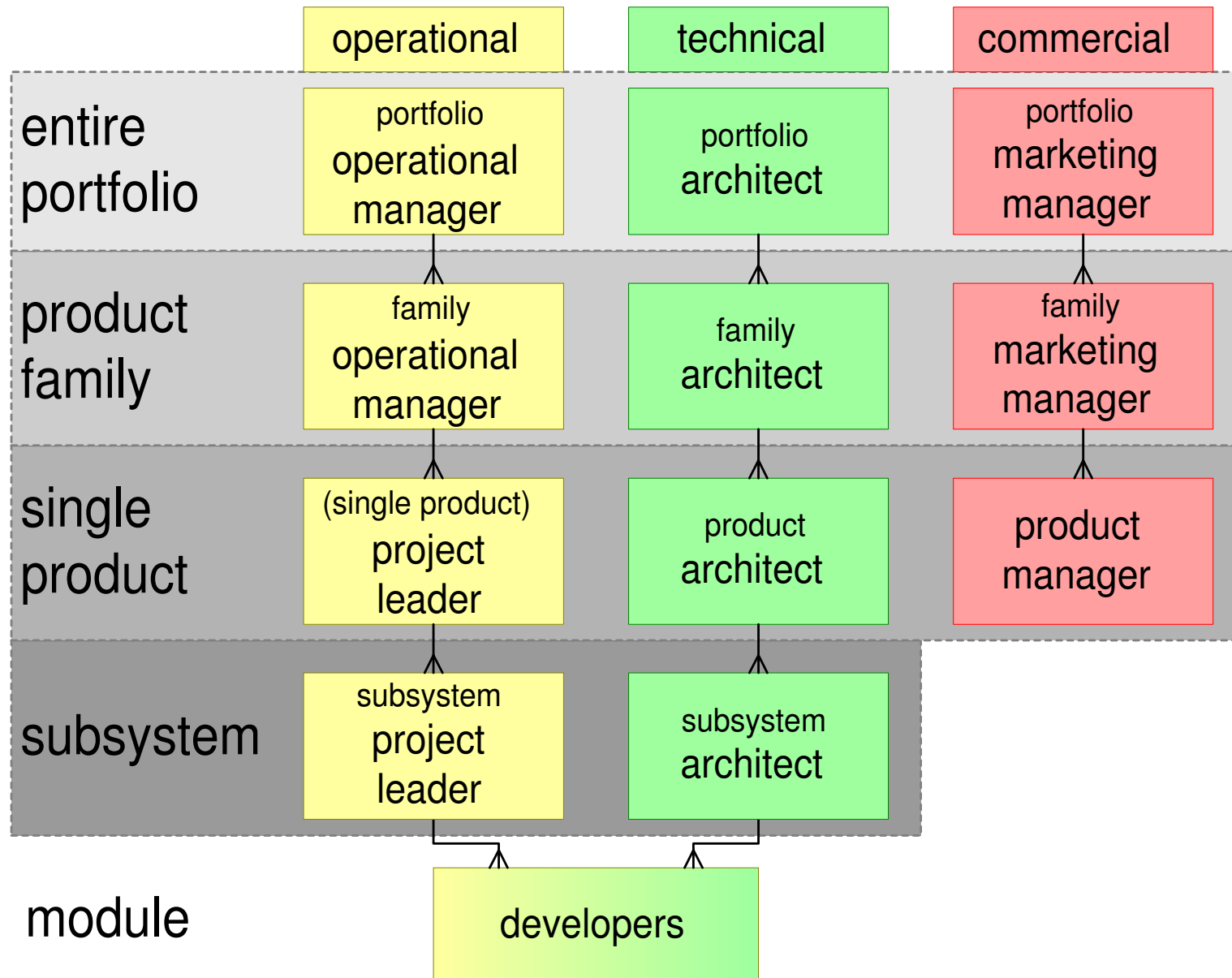
# Simplified process diagram for project business



# Decomposition of the Product Creation Process

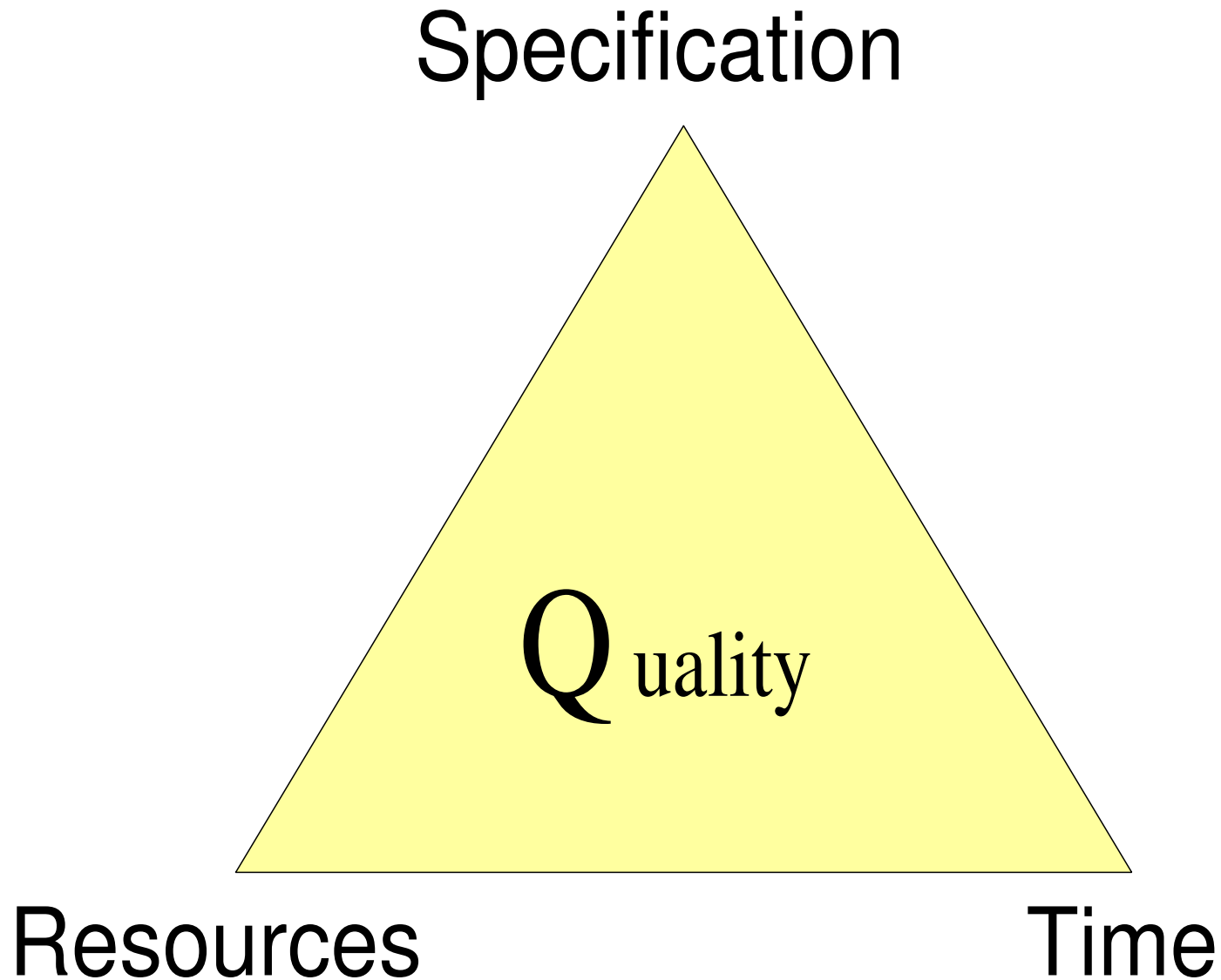


# Operational Organization of the PCP



# Prime Responsibilities of the Operational Leader

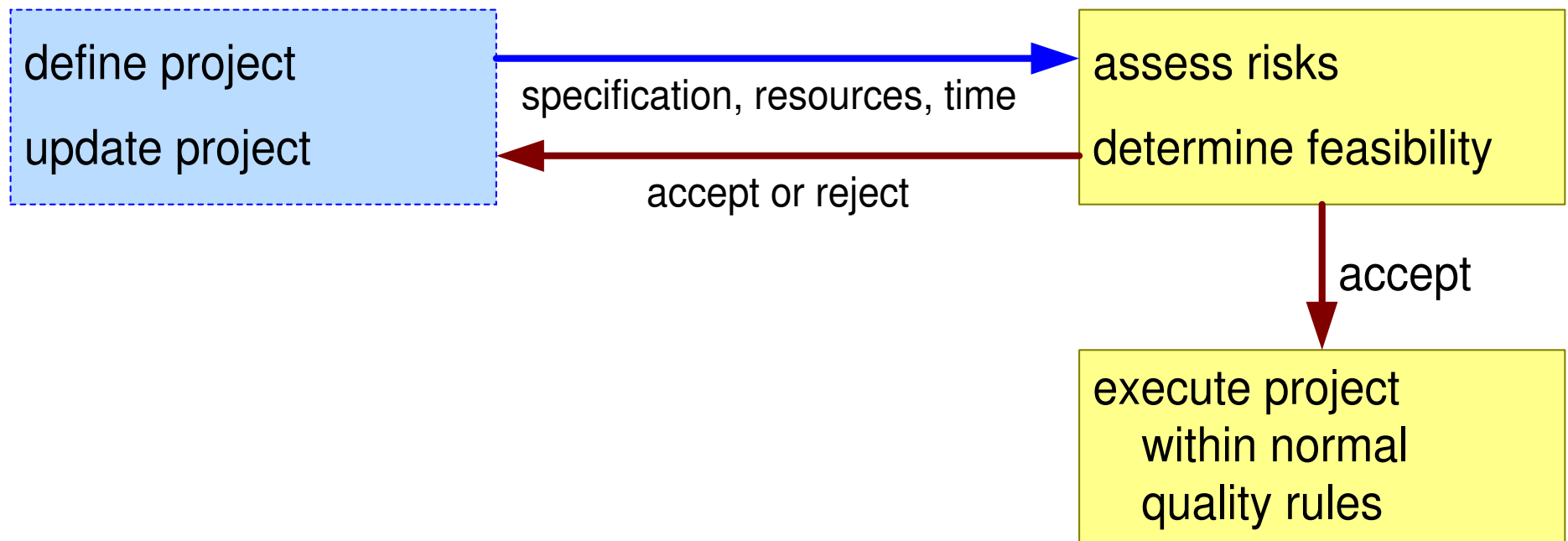
---



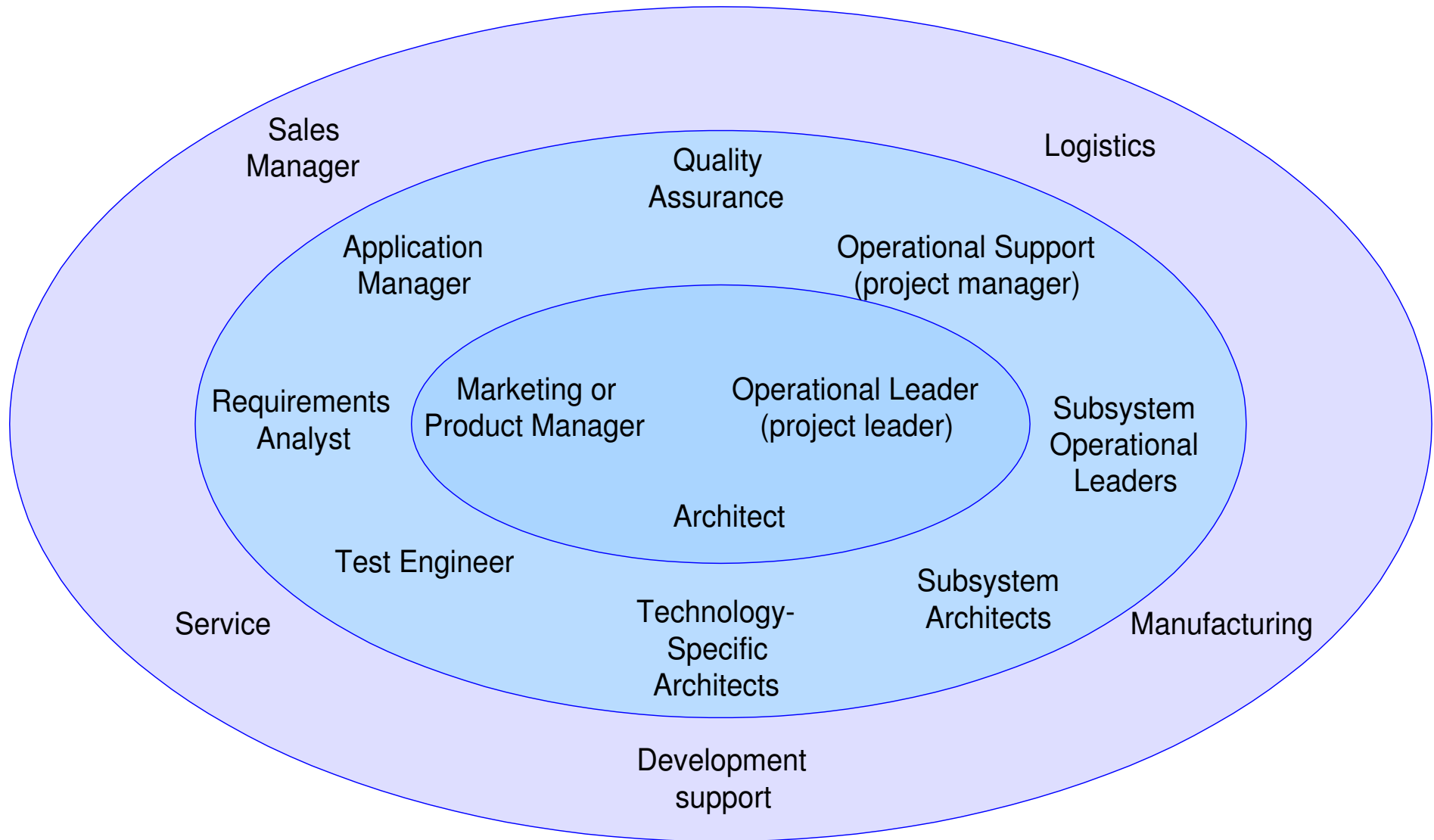
# The Rules of the Operational Game

business management

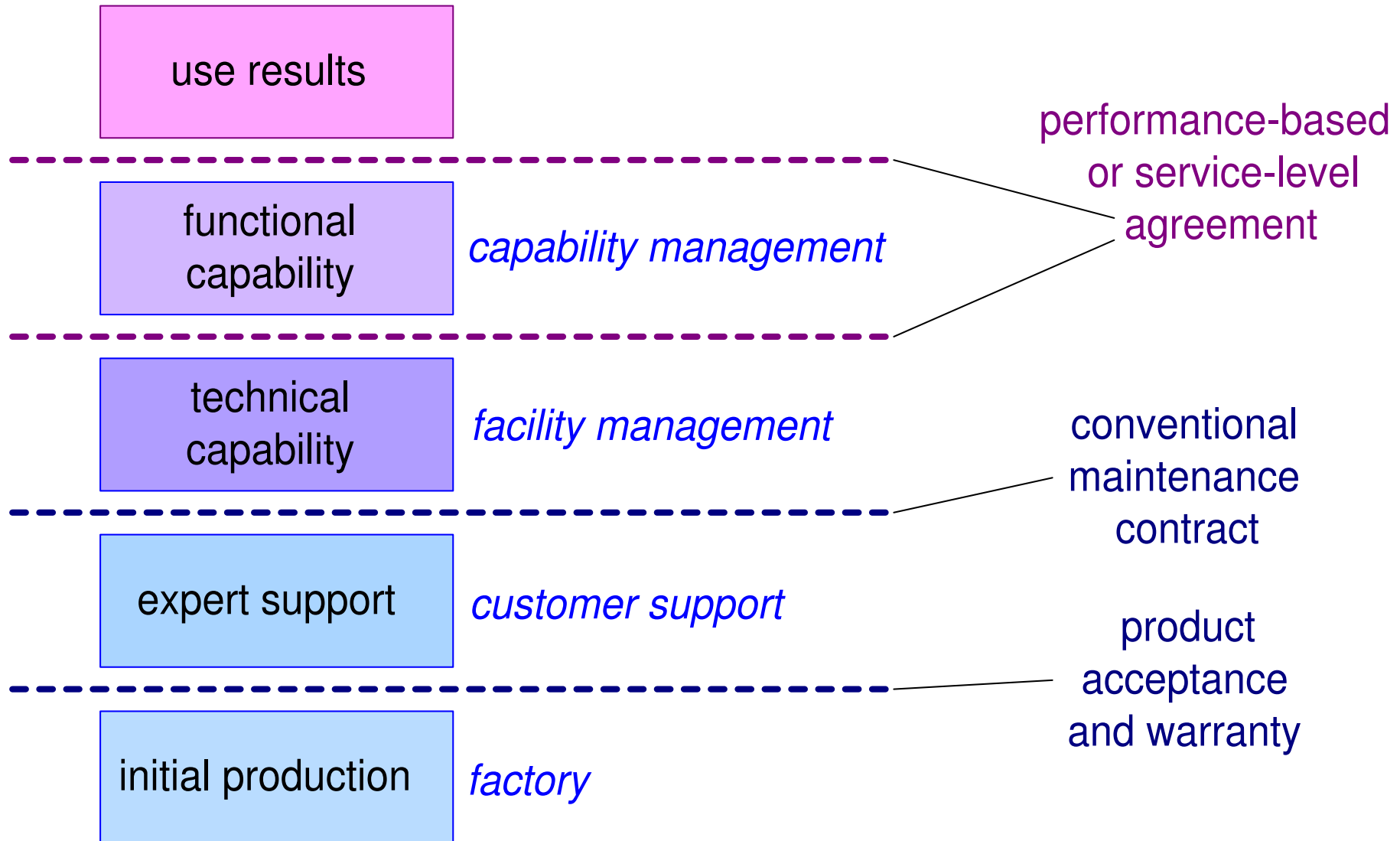
project leader



# Operational Teams



# What Service Level to Deliver?



# Role and Task of the System Architect

by *Gerrit Muller* Embedded Systems Institute  
e-mail: `gerrit.muller@embeddedsystems.nl`  
`www.gaudisite.nl`

## Abstract

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

### Distribution

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

February 11, 2012  
status: preliminary  
draft  
version: 1.0



# The Role and Task of the System Architect

by *Gerrit Muller* Buskerud University Collge

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

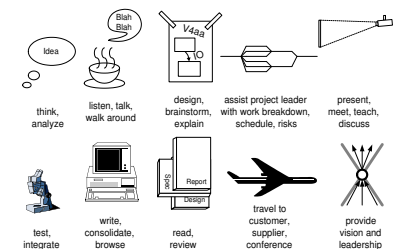
## Abstract

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.

## Distribution

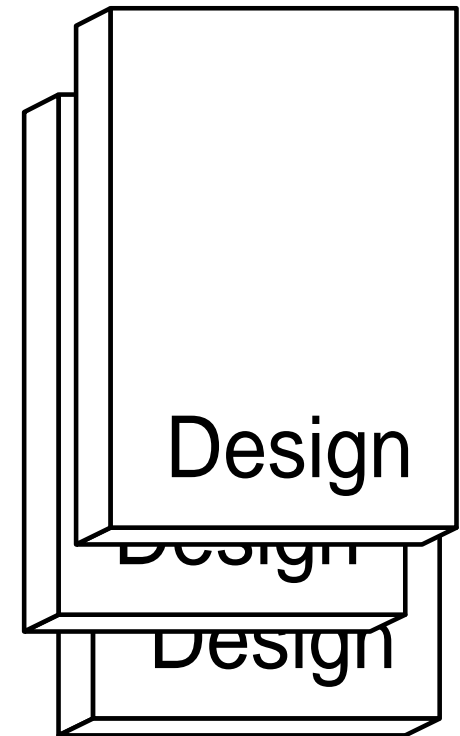
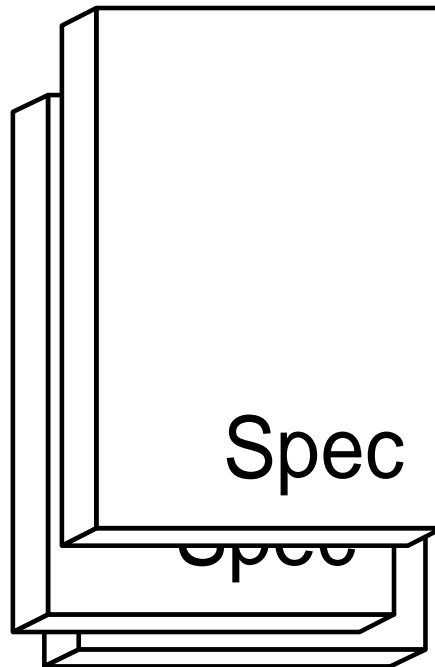
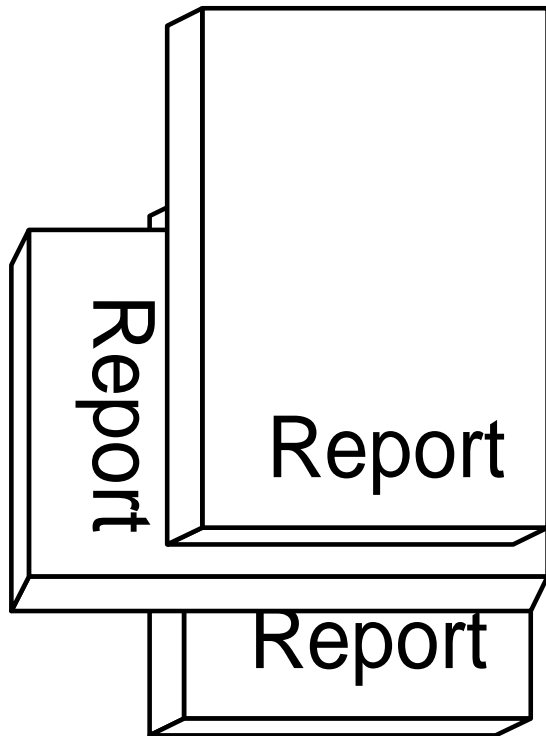
This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution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

February 11, 2012  
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)*

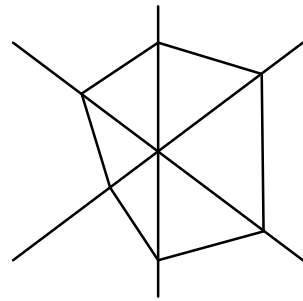
Verification Specification *(how the system will be verified)*

Verification Report *(the result of the verification)*

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

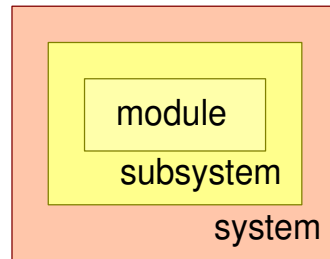
Roadmap

# Responsibilities of the System Architect



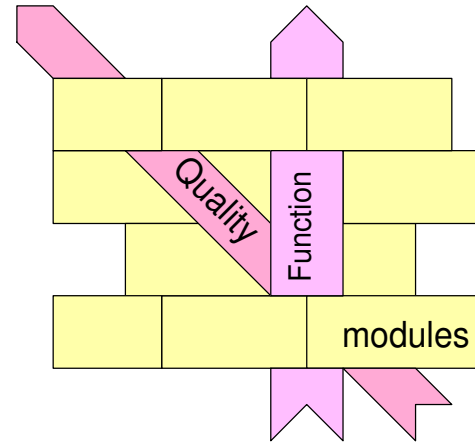
Balance

Requirement  
Spec  
Design  
Realization



Consistency

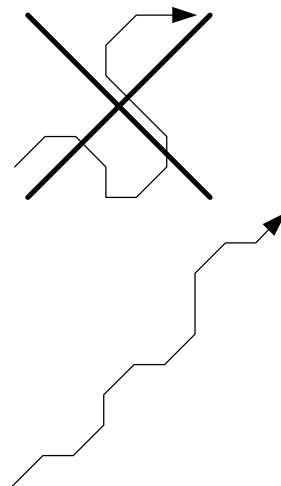
Decomposition  
Integration



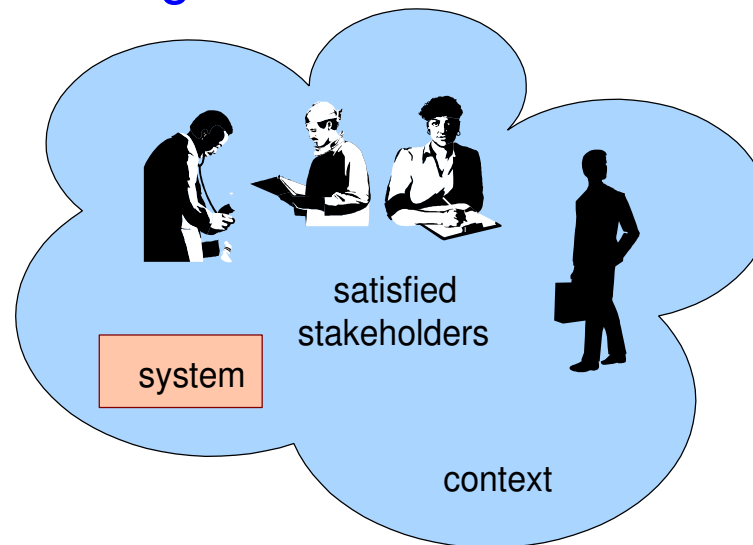
Overview

KISS

Elegance  
Simple



Integrity



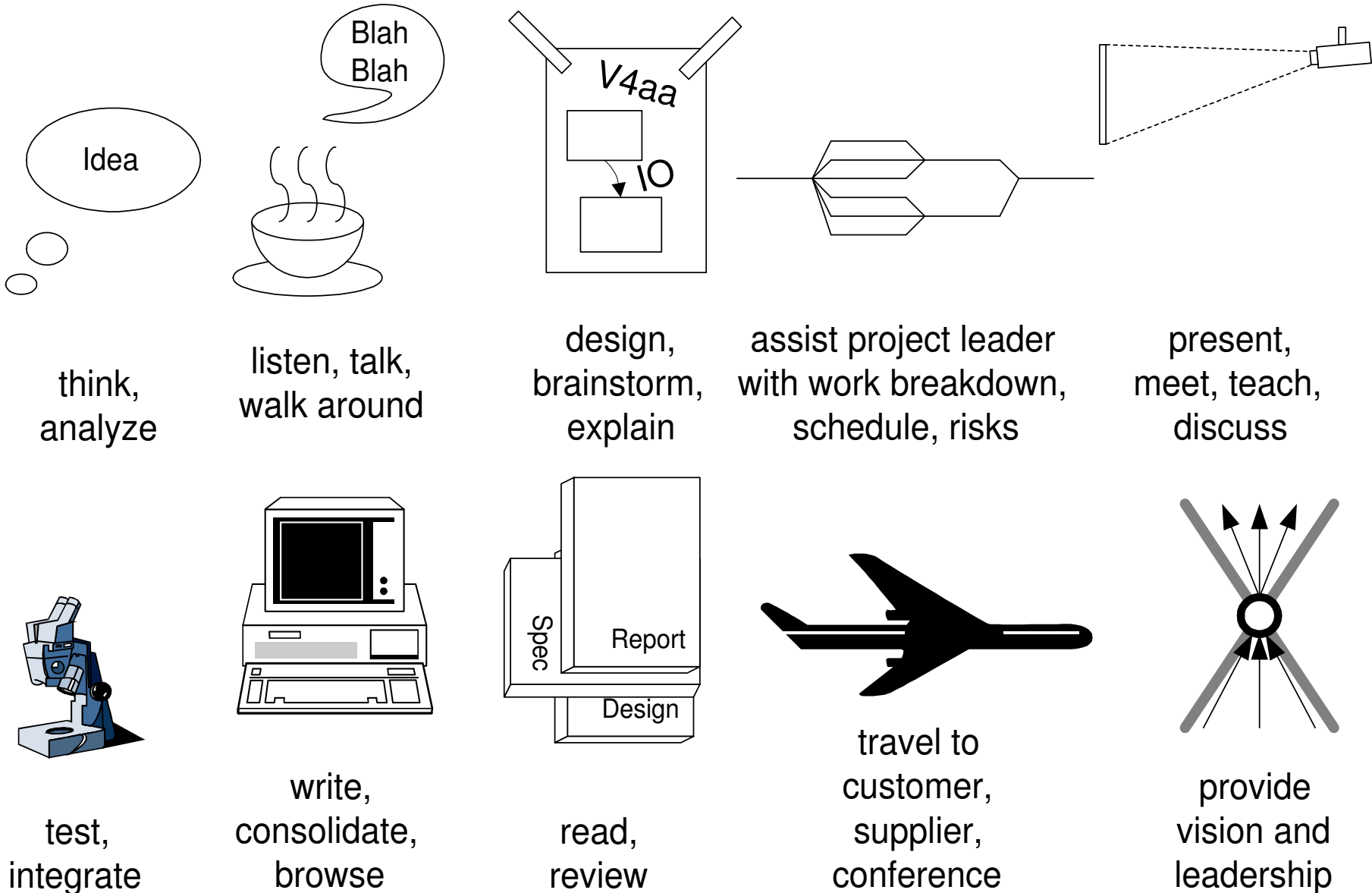
Fitting

# Examples of Secondary Responsibilities

---

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

# What does the System Architect do?



# From Detail to Overview

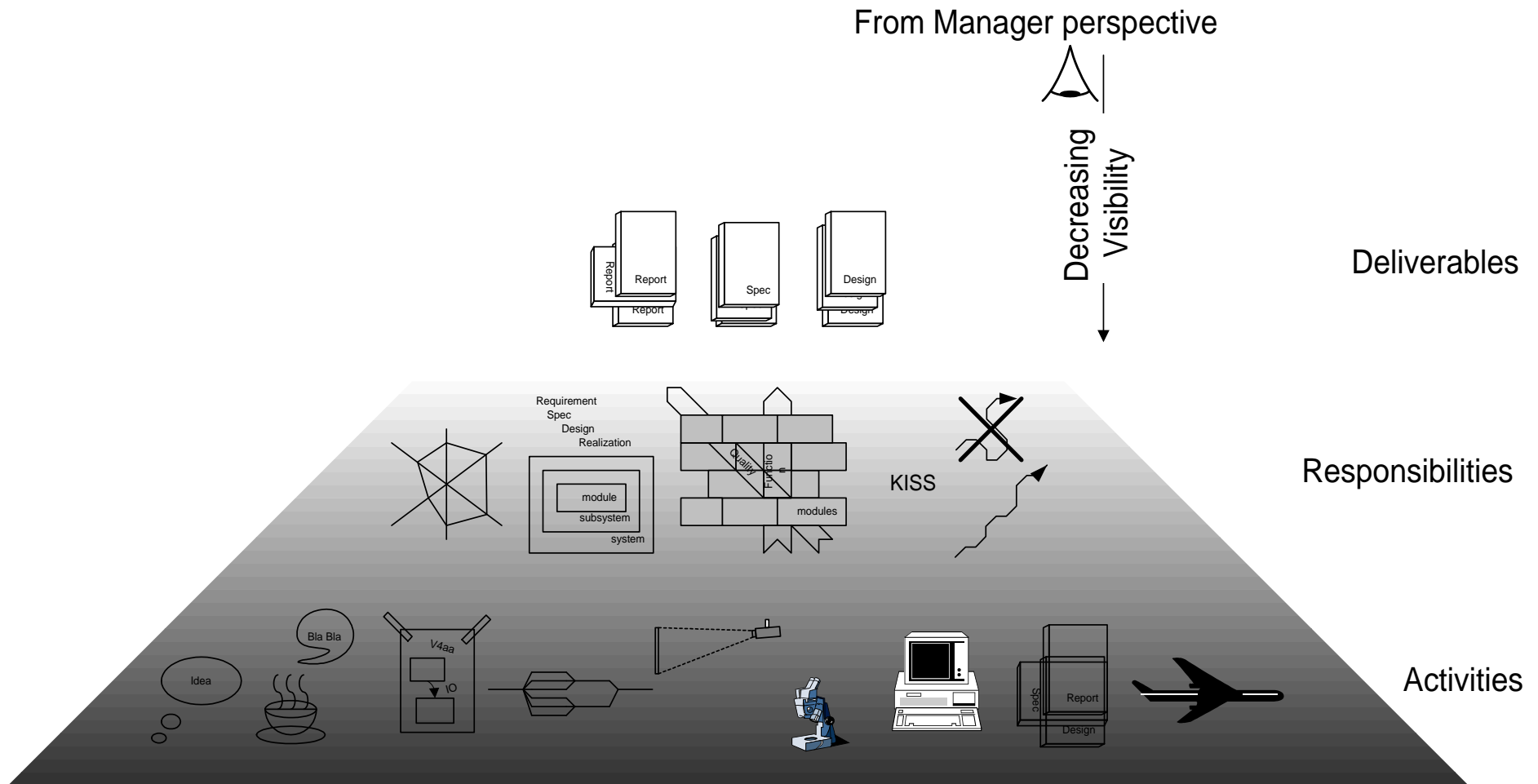
	Quantity per year (order-of- magnitude)	architect time per item	
consolidation in deliverables	driving views	10	100 h
meetings	shared issues	$10^2$	1 h
informal contacts	touched details	$10^4$	0.5 – 10 min
sampling scanning	seen details	$10^5 - 10^6$	0.1 – 1 sec
	product details	$10^7 - 10^{10}$	
	real-world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

# Reality or Virtuality?

---

Abstractions only exist for concrete facts.

# Visible Output versus Invisible Work



# The Awakening of a System Architect

by *Gerrit Muller* Buskerud University College

e-mail: `gerrit.muller@embeddedsystems.nl`

`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.

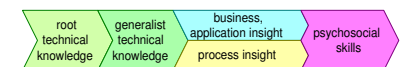
## Distribution

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

February 11, 2012

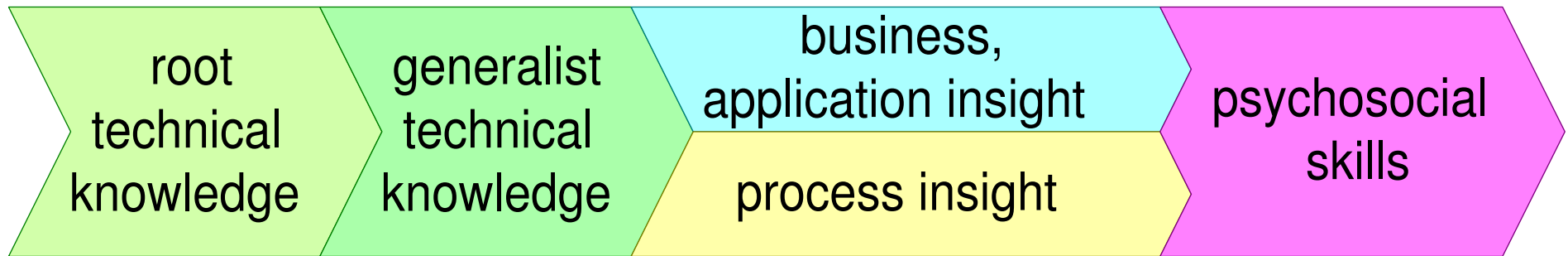
status: concept

version: 1.1

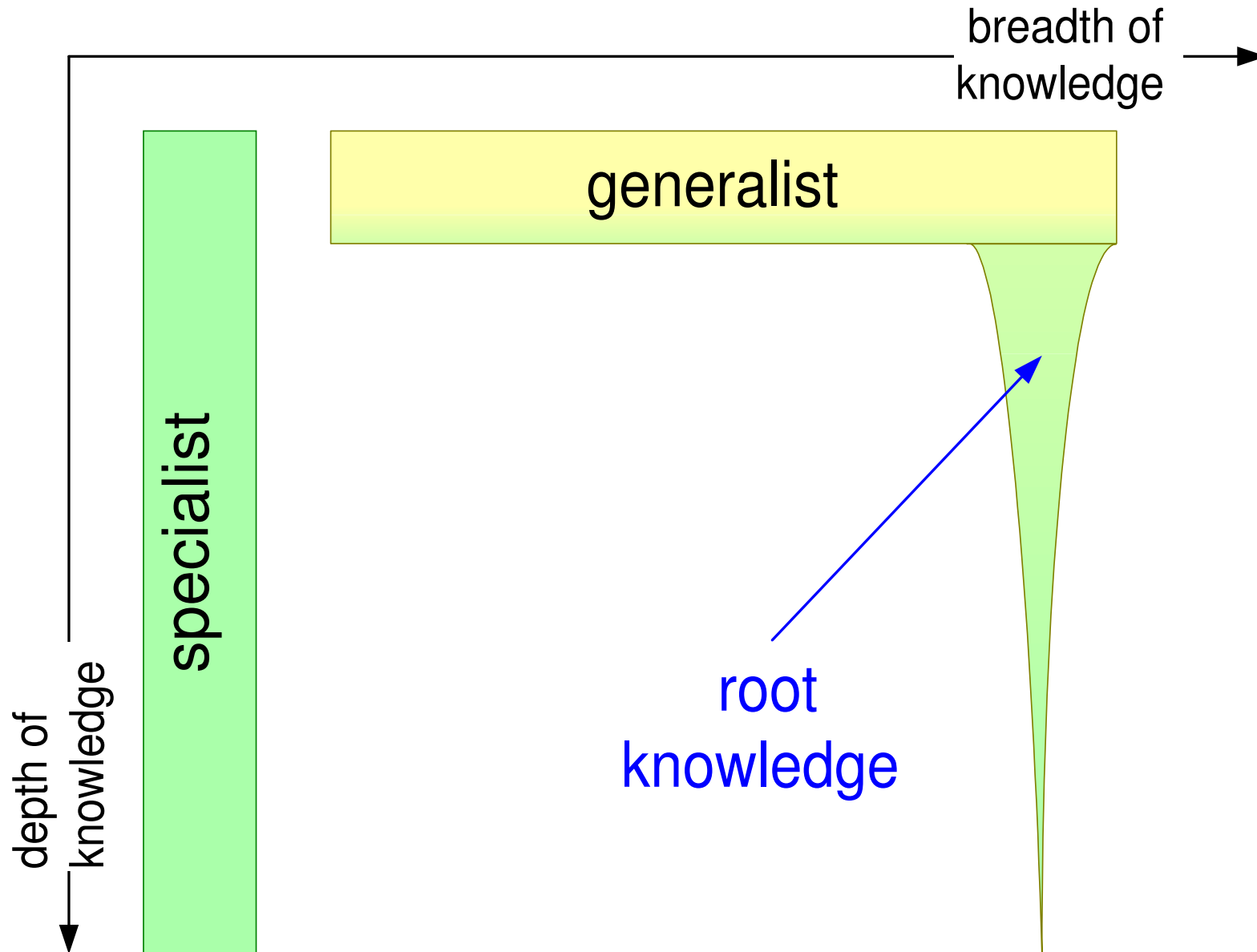


# Typical Growth of a System Architect

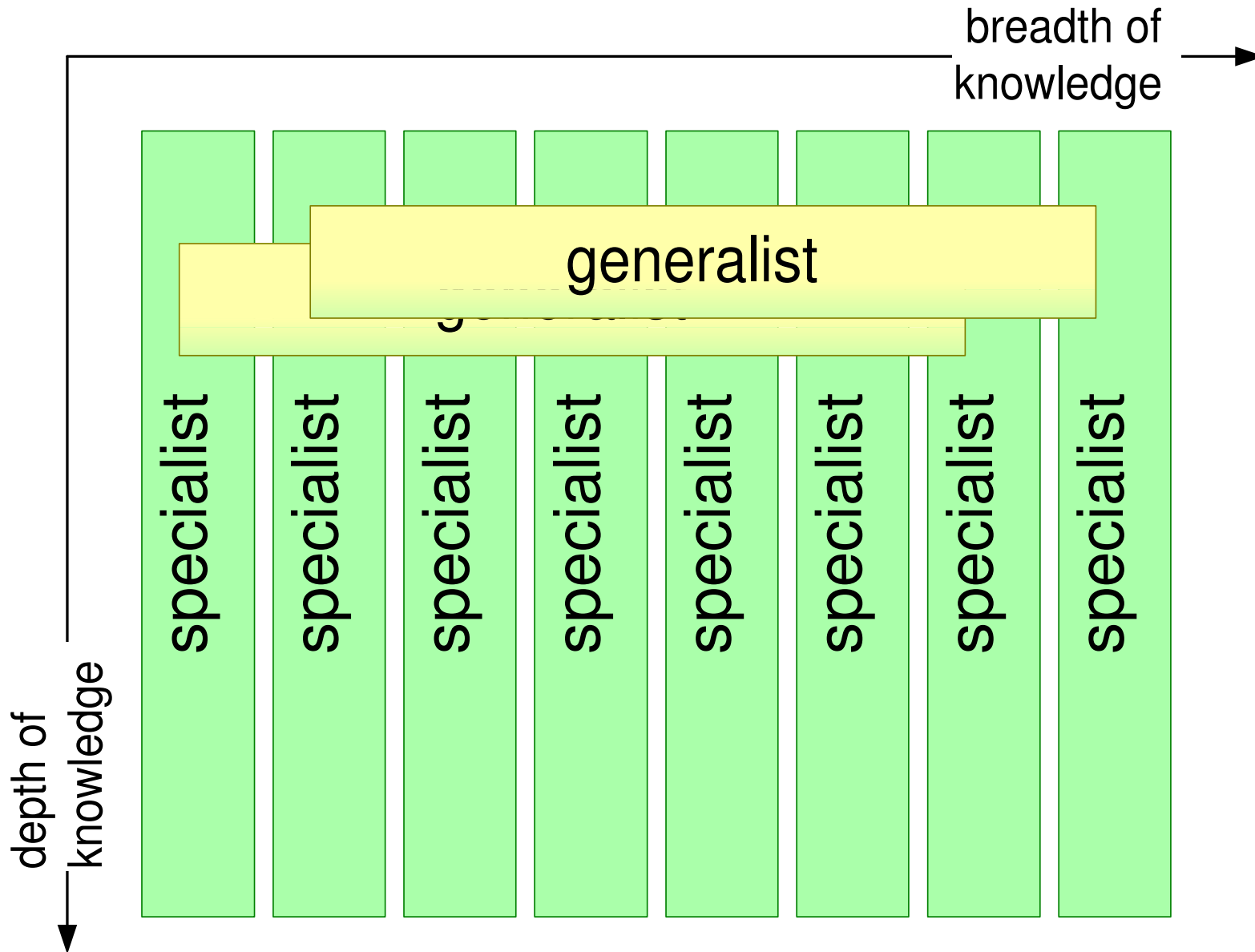
---



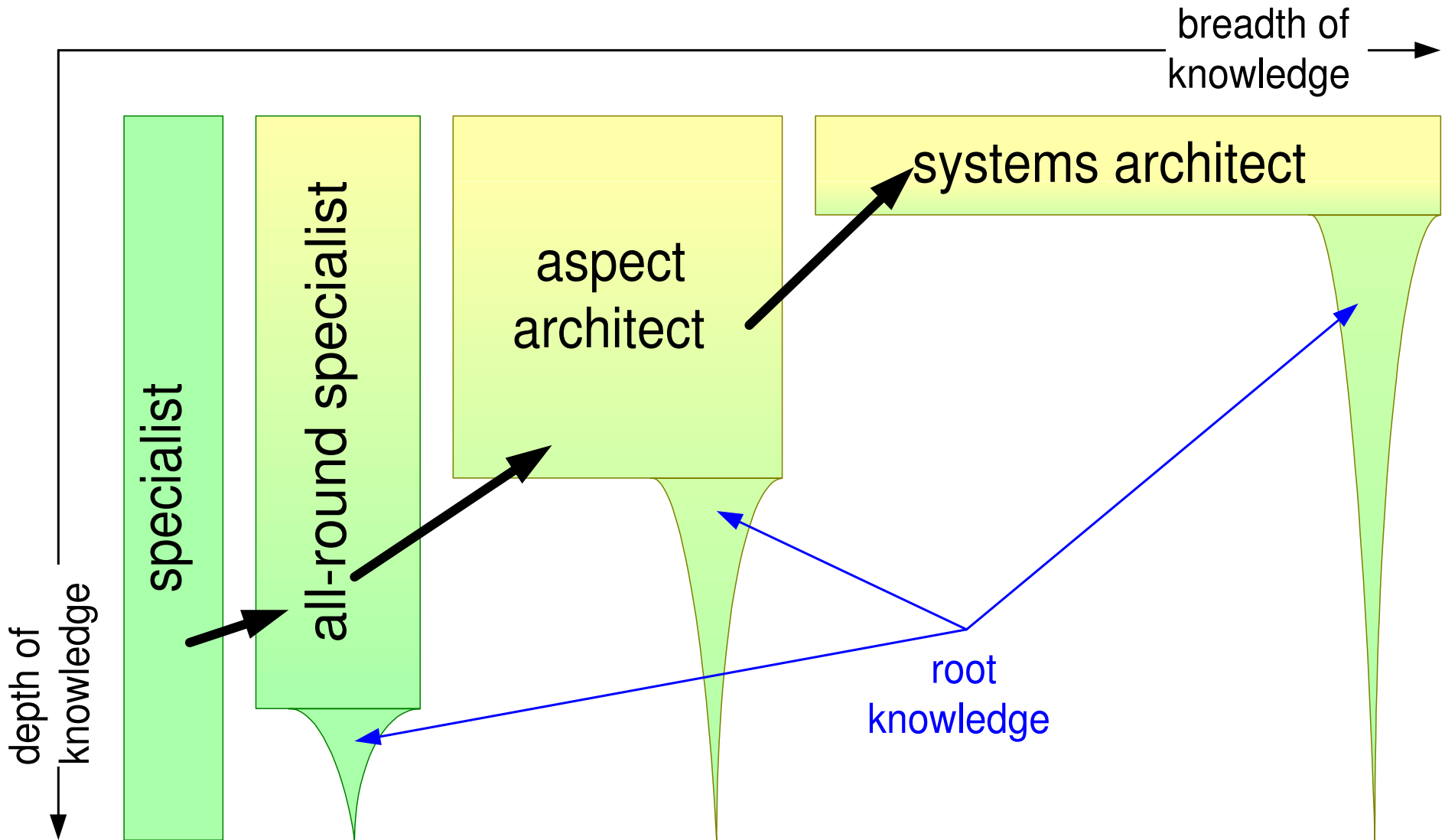
# Generalist versus Specialist



# Generalists and Specialists are Complementary



# Spectrum from Specialist to System Architect



# Architecting Interaction Styles

by *Gerrit Muller* Buskerud University College

e-mail: `gerrit.muller@embeddedsystems.nl`

`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.

## Distribution

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

February 11, 2012

status: draft

version: 0.2

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

# 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:

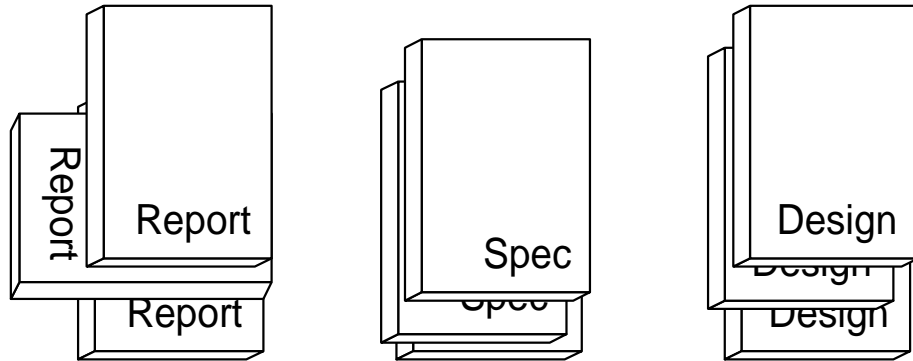
- 1 operational leader (project leader)
- 1 system architect
- 1 marketing manager
- 1 observer (optional)

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

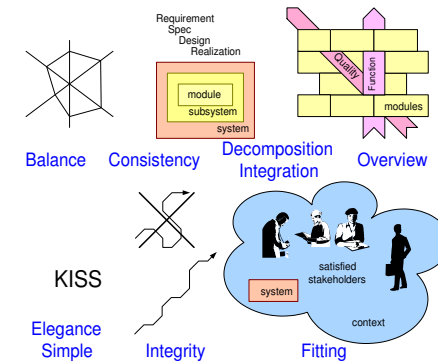
Present (max. 2 flips) the result and the process (the relation and interaction of the three roles).

# Role and Task of a System Architect

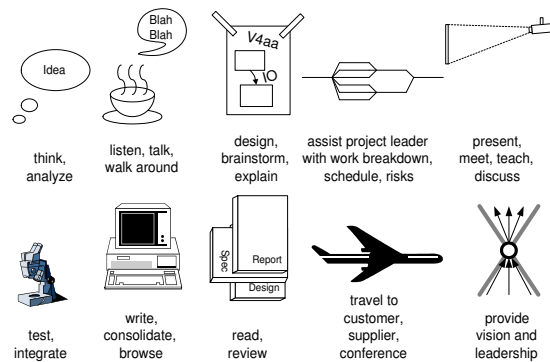
## Deliverables



## Responsibilities



## Daily Activities

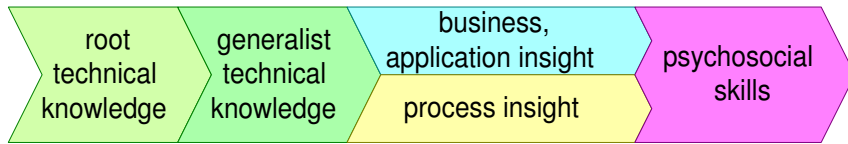


## From detail to overview

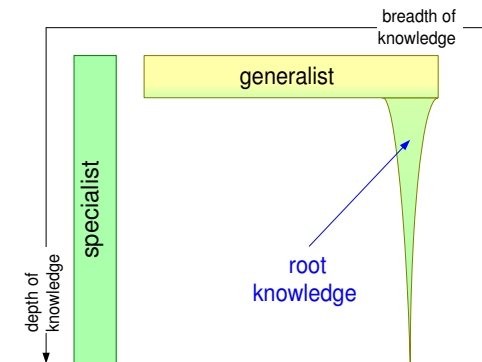
	Quantity per year (order-of-magnitude)	architect time per item
driving views	10	100 h
shared issues	$10^2$	1 h
touched details	$10^4$	0.5 – 10 min
seen details	$10^5 - 10^6$	0.1 – 1 sec
product details	$10^7 - 10^{10}$	
real-world facts	infinite	

# Personal characteristics of a System Architect

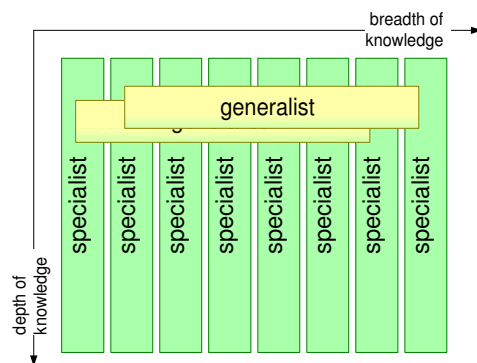
## Typical growth of a Architect



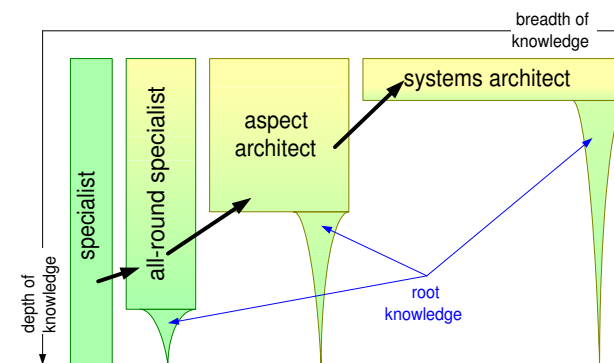
## Generalist vs Specialist



## Complementary Roles



## Role Spectrum



# Module Requirements

by *Gerrit Muller* Embedded Systems Institute  
e-mail: `gerrit.muller@embeddedsystems.nl`  
`www.gaudisite.nl`

## Abstract

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

### Distribution

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

February 11, 2012  
status: concept  
version: 1.3



Embedded Systems  
INSTITUTE

# Fundamentals of Requirements Engineering

by *Gerrit Muller* Buskerud University College

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

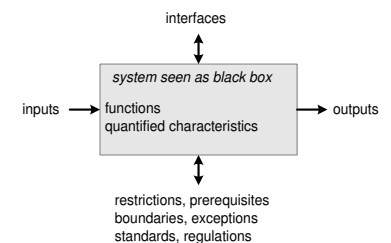
## Abstract

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.

## Distribution

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

February 11, 2012  
status: concept  
version: 0.1



# Definition of “Requirement”

---

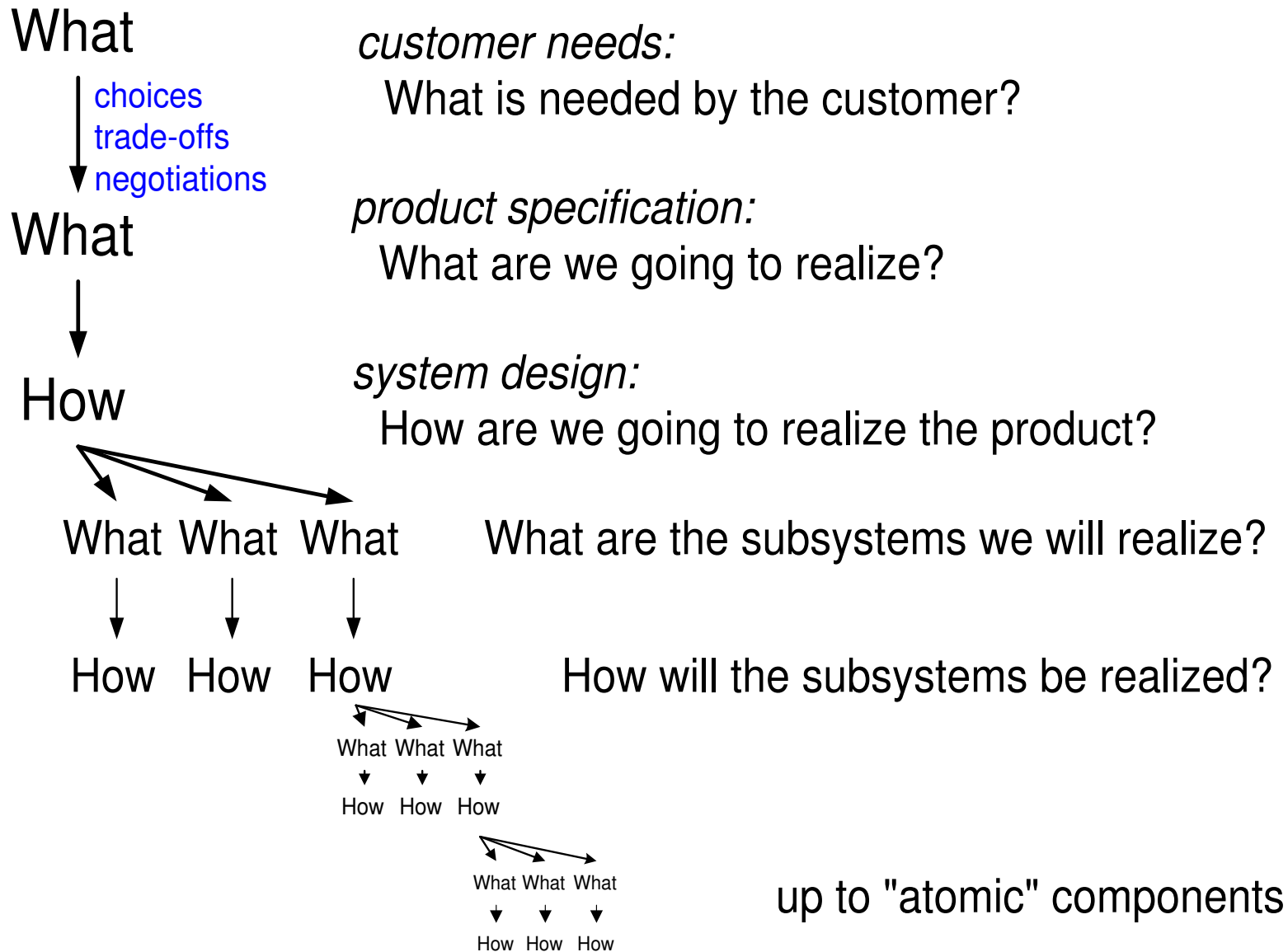
Requirements describing the needs of the customer:  
*Customer Needs*

Requirements describing the characteristics of the final resulting product: *Product Specification*

The requirements management process recursively applies definition 2 for every level of decomposition.

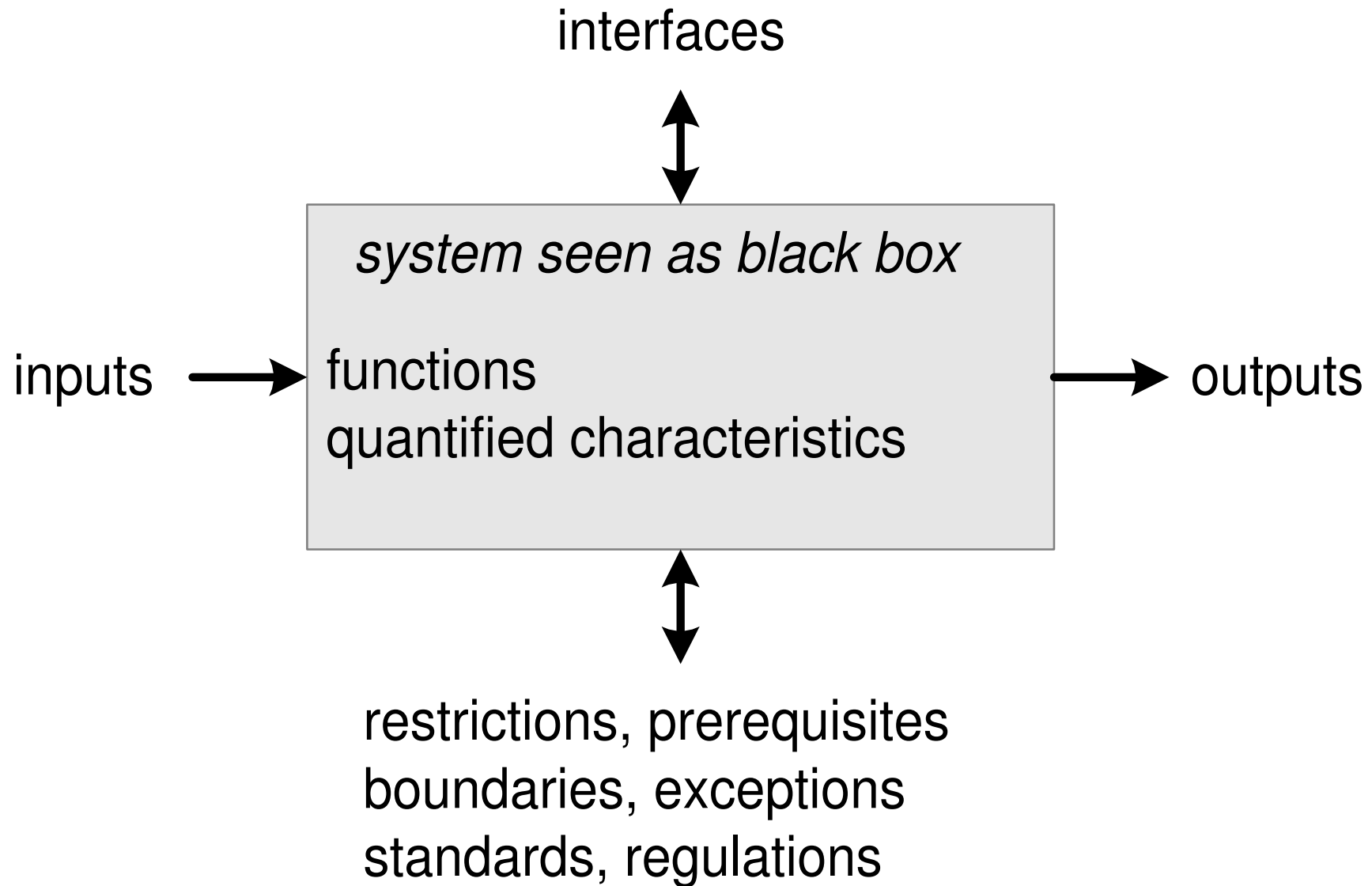
Requirements describing the needs of the company itself over the life cycle: *Life Cycle Needs*

# Flow of Requirements

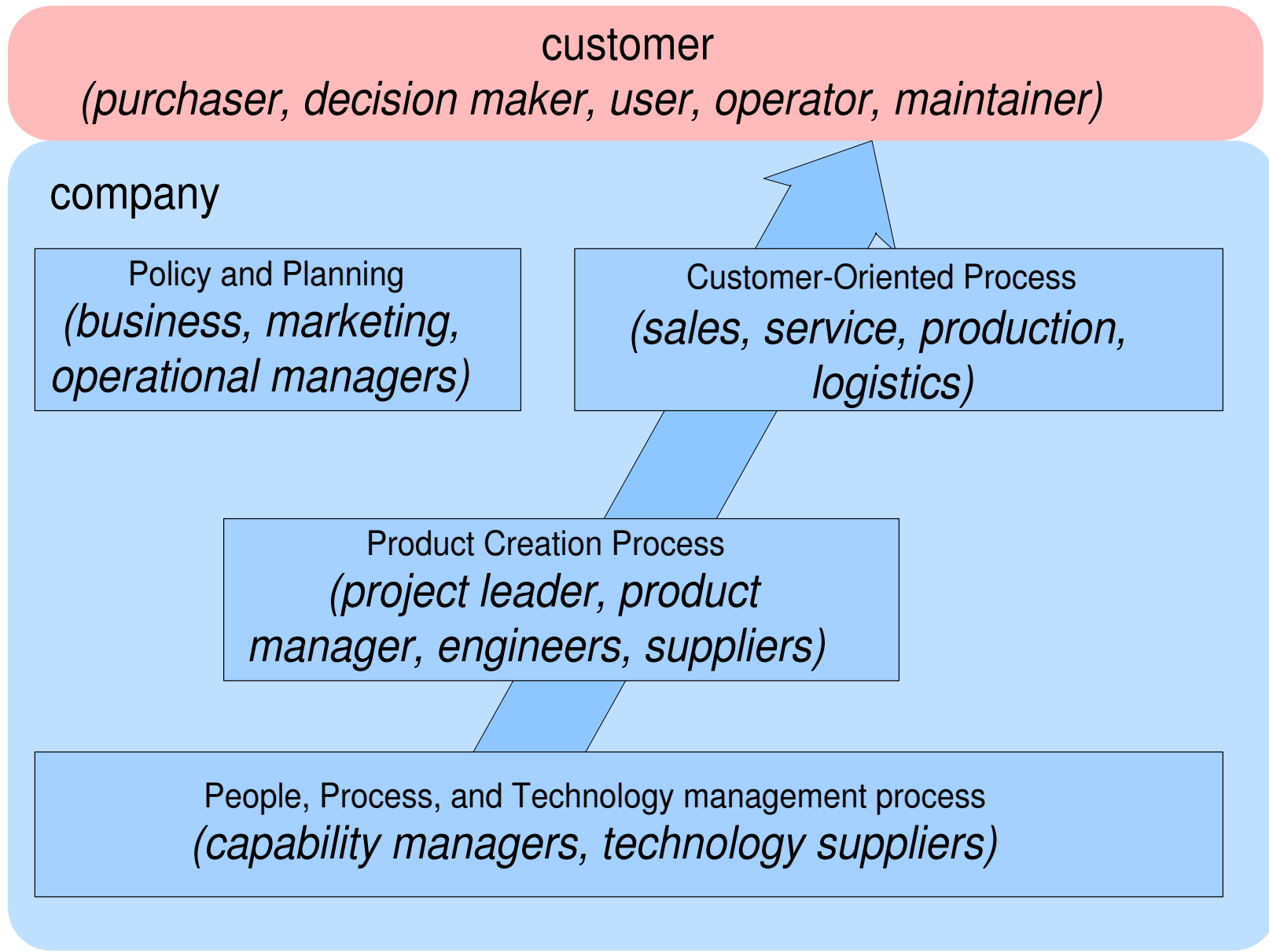


# System as a Black Box

---



# 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* Embedded Systems Institute  
e-mail: `gerrit.muller@embeddedsystems.nl`  
`www.gaudisite.nl`

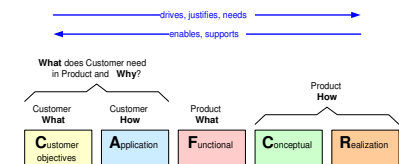
## Abstract

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.

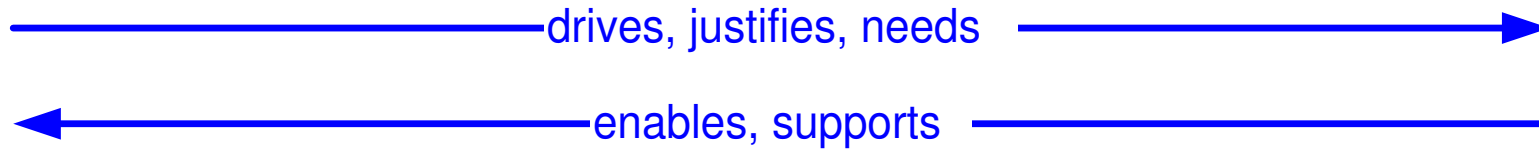
## Distribution

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

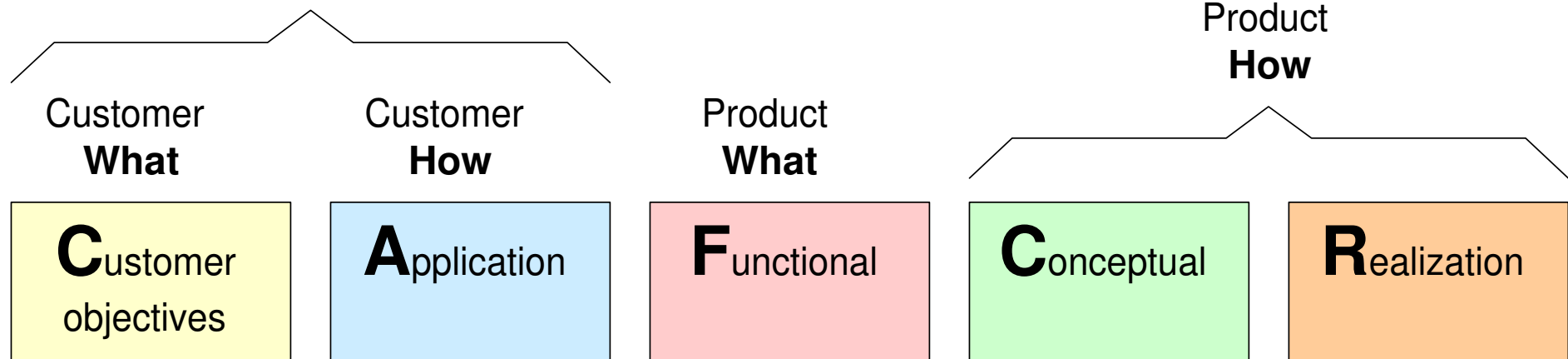
February 11, 2012  
status: draft  
version: 0.4



# The “CAFCR” model

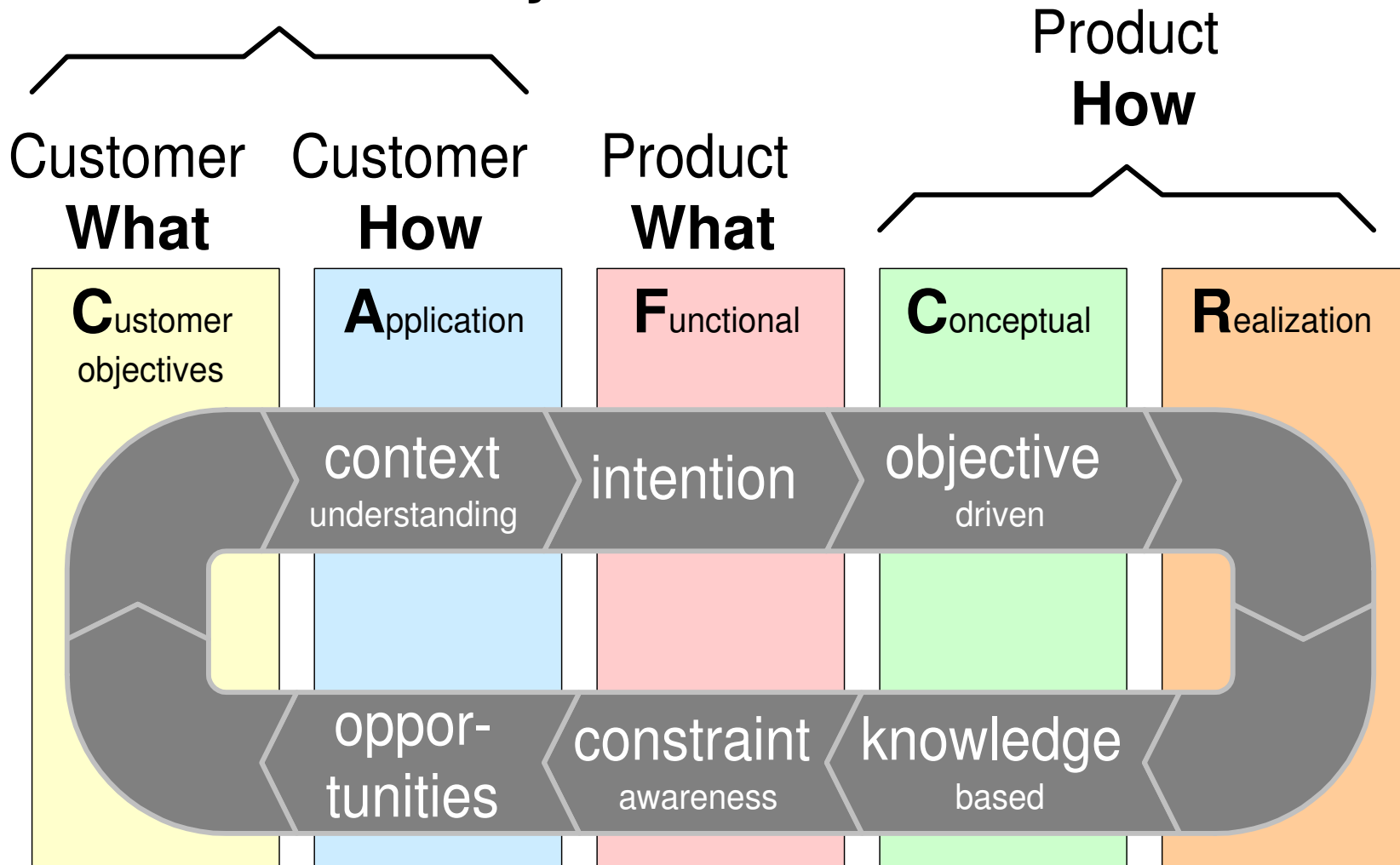


**What** does Customer need  
in Product and **Why?**

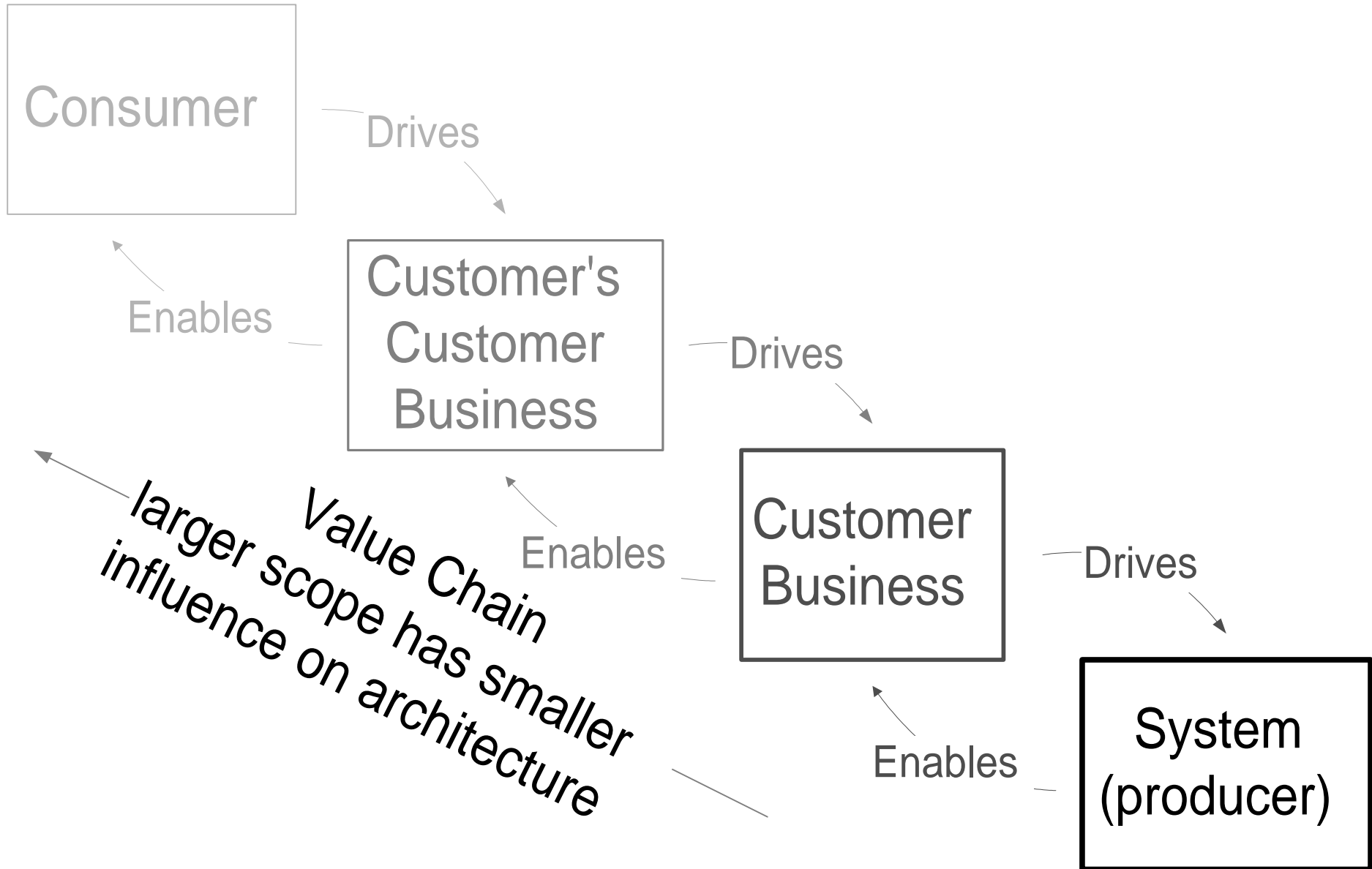


# Integrating CAFCR

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



# CAFCR can be applied recursively

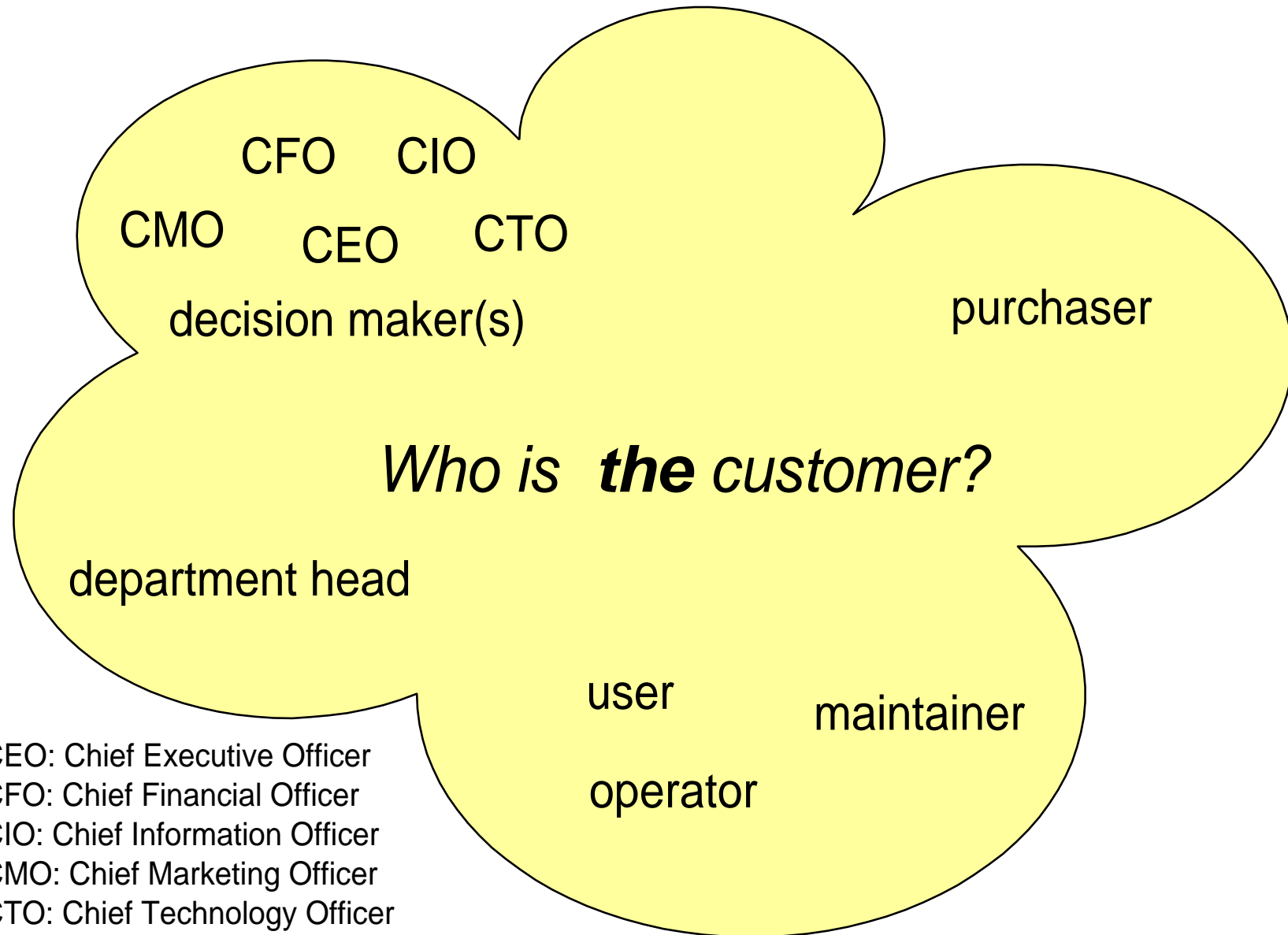


# Market segmentation

---

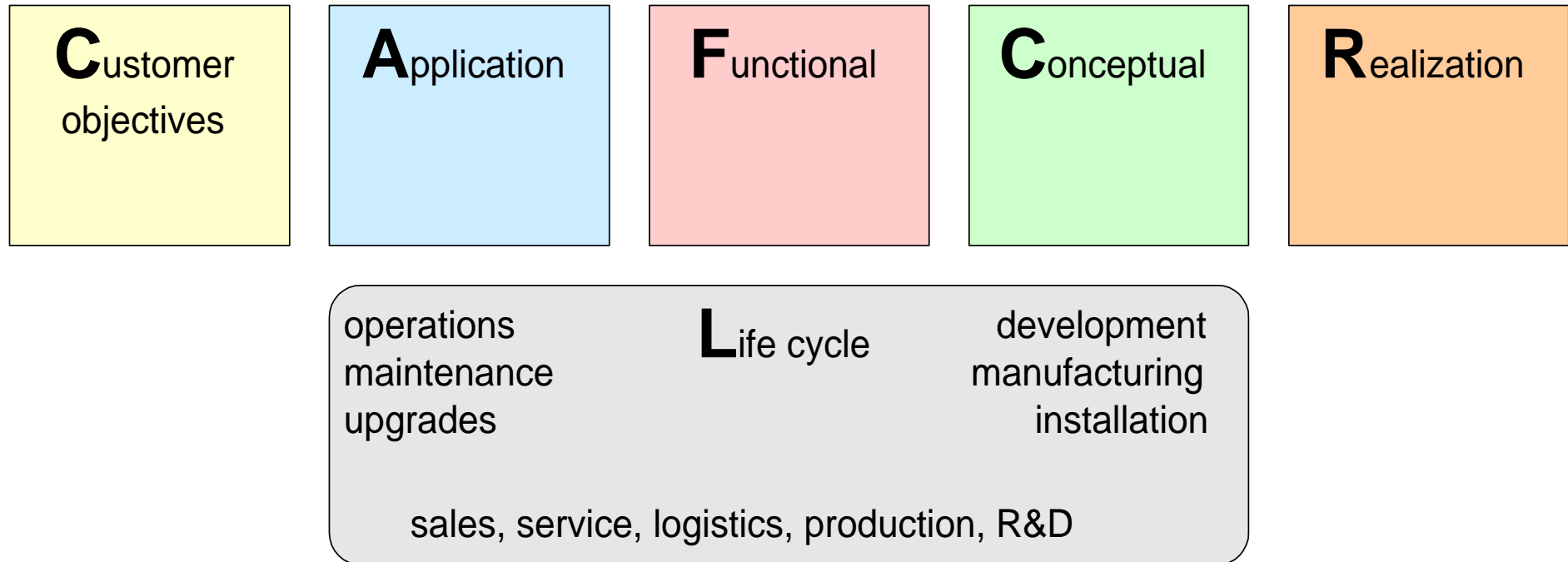
segmentation axis	examples
geographical	USA, UK, Germany, Japan, China
business model	profit, non profit
economics	high end versus cost constrained
consumers	youth, elderly
outlet	retailer, provider, OEM, consumer direct

# Example of a small buying organization



# CAFCR+ model; Life Cycle View

---



# Key Drivers How To

by *Gerrit Muller* Buskerud University College

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

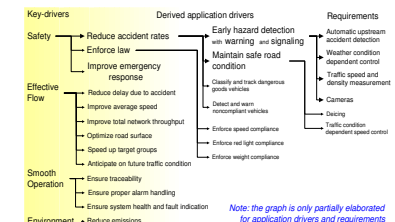
## Abstract

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

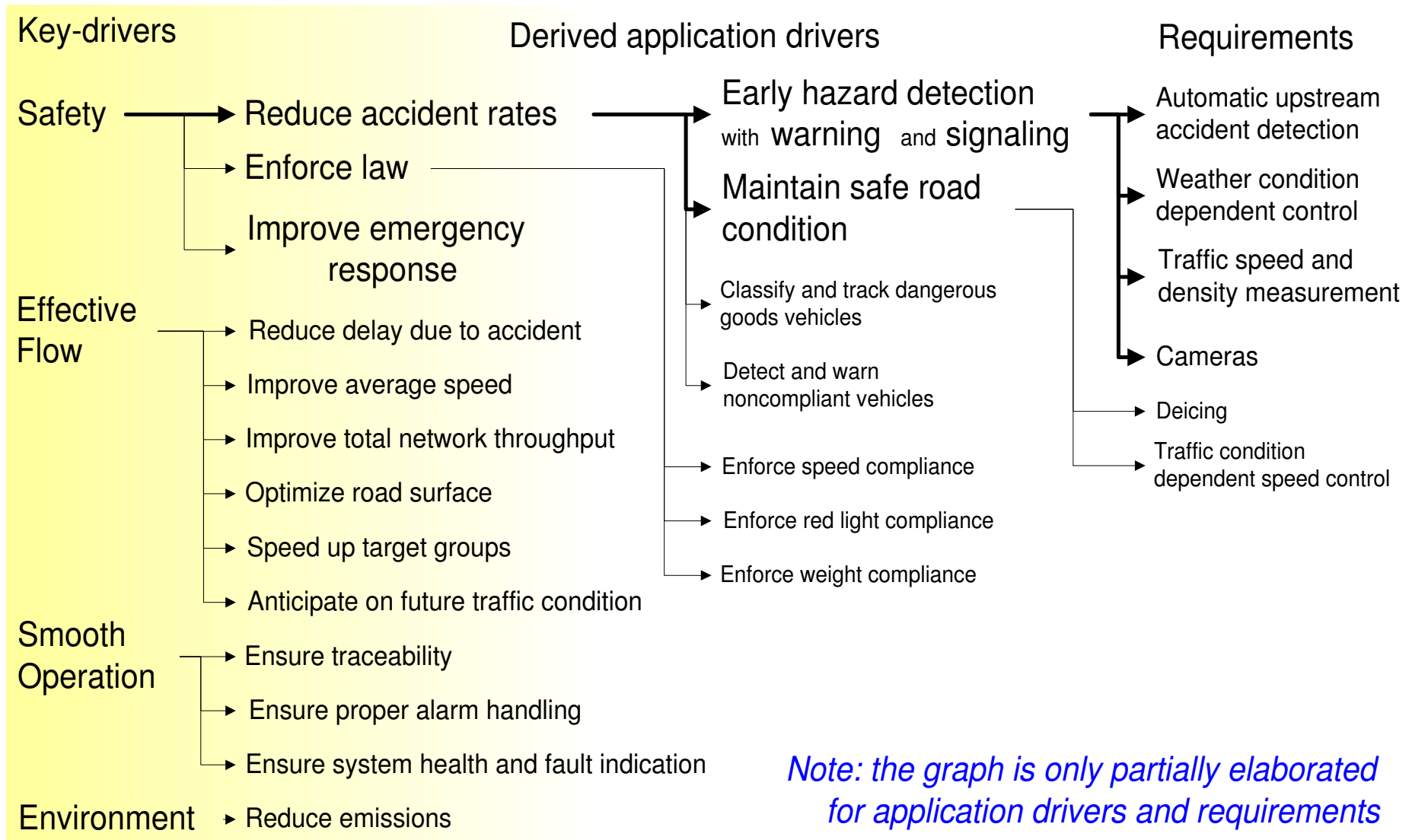
## Distribution

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

February 11, 2012  
status: draft  
version: 0.2



# Example Motorway Management Analysis



# Method to create Key Driver Graph

---

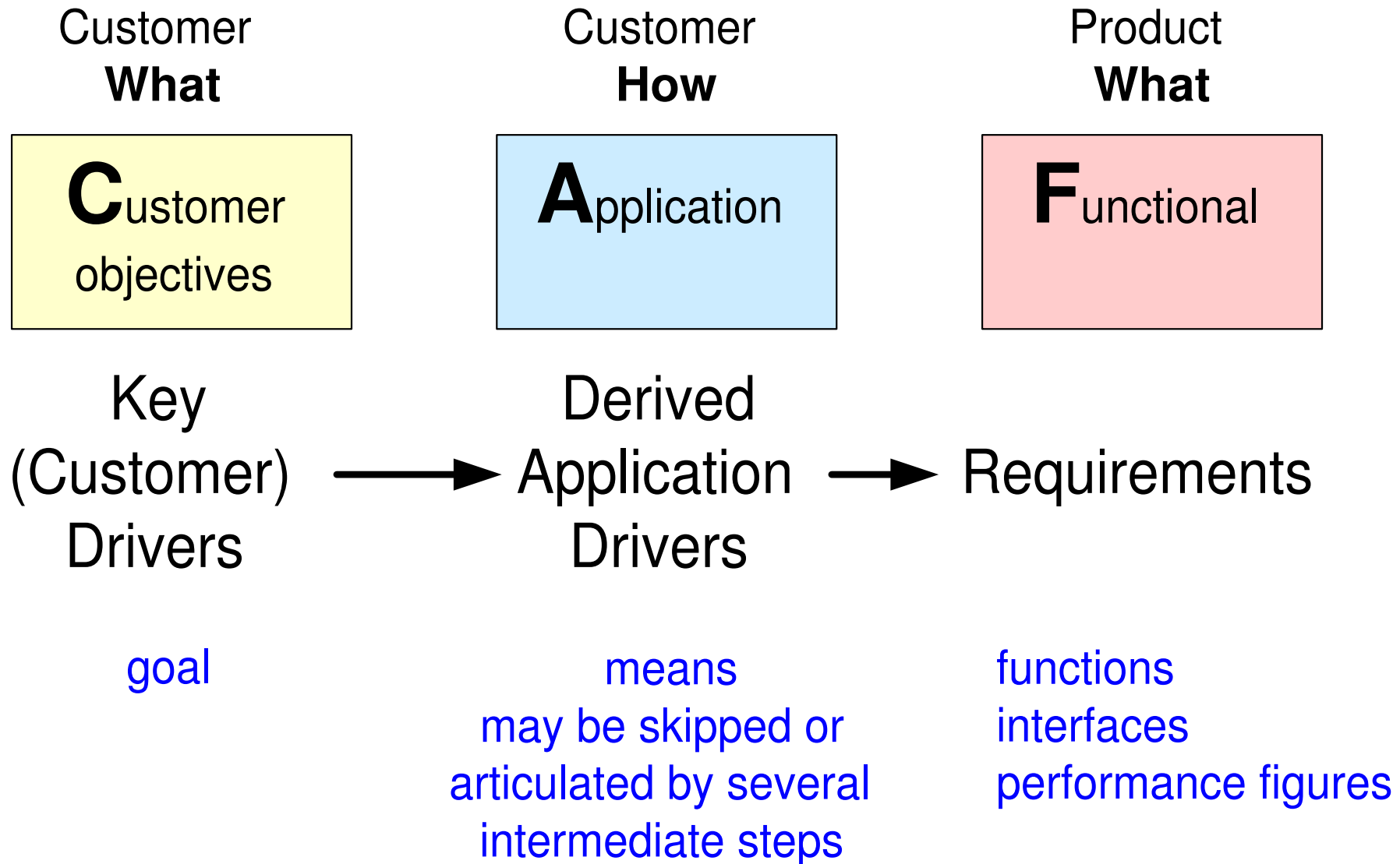
- |  |  |
|--|--|
| • Define the scope specific.   | in terms of stakeholder or market segments   |
| • Acquire and analyze facts  | extract facts from the product specification<br>and ask why questions about the specification of existing products . |
| • Build a graph of relations between drivers and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

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

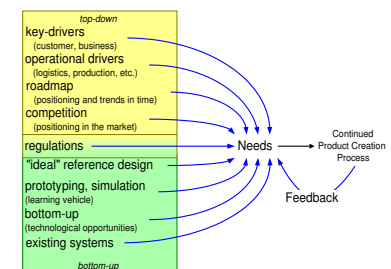
## Abstract

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.

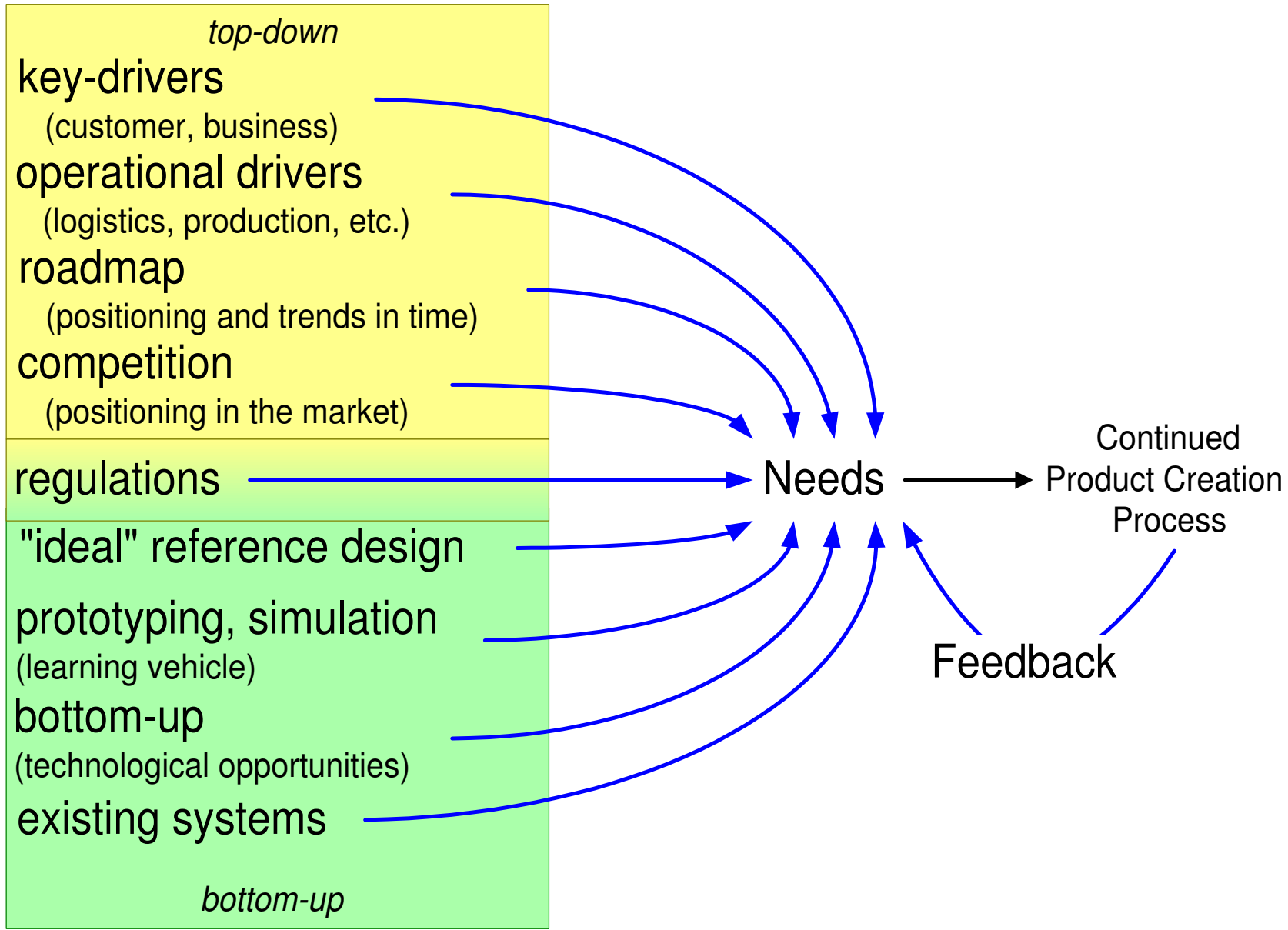
## Distribution

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

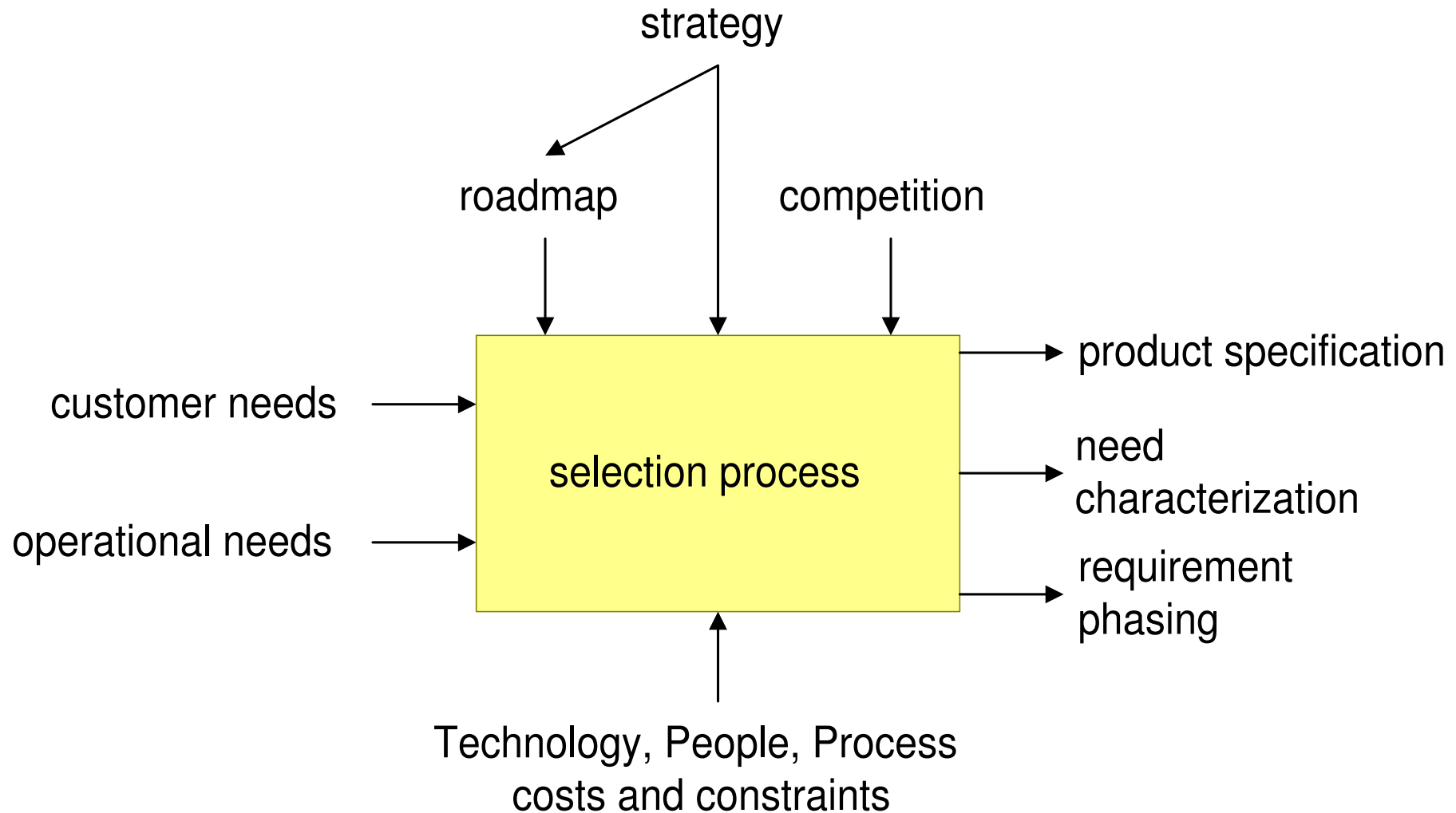
February 11, 2012  
status: draft  
version: 0



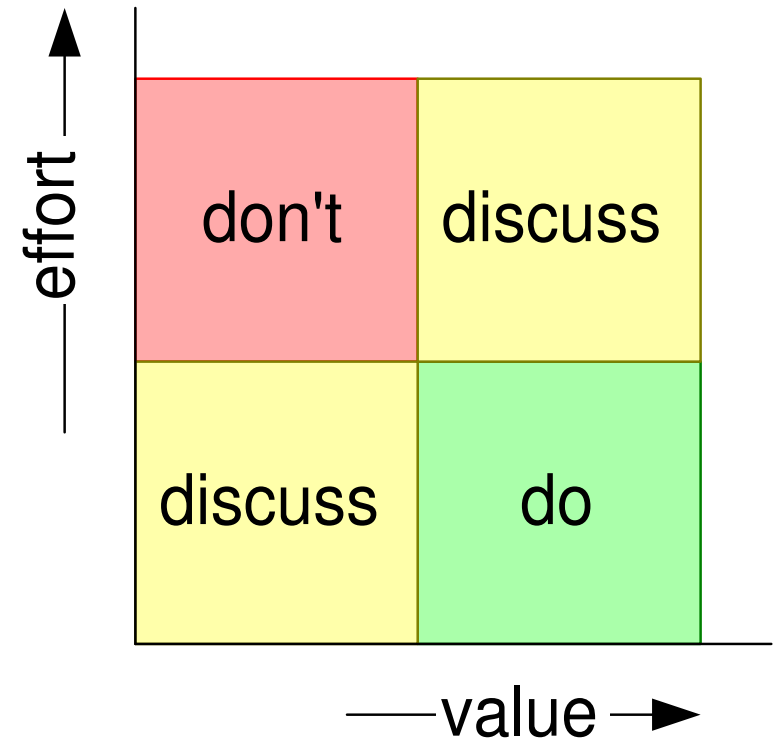
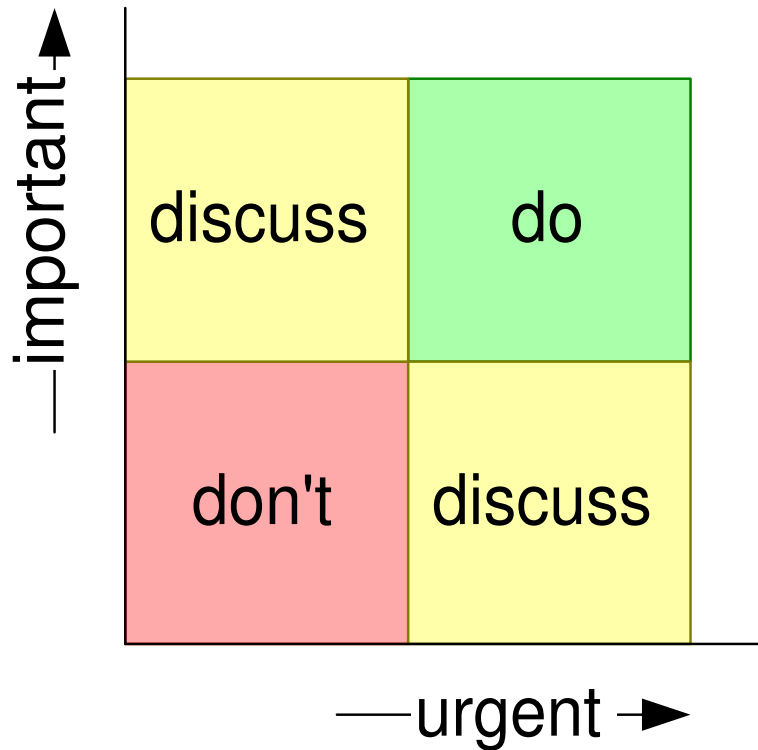
# Complementary Viewpoints to Capture Requirements



# 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.

# Story How To

by *Gerrit Muller* Embedded Systems Institute  
e-mail: `gerrit.muller@embeddedsystems.nl`  
`www.gaudisite.nl`

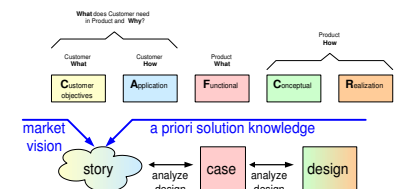
## Abstract

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.

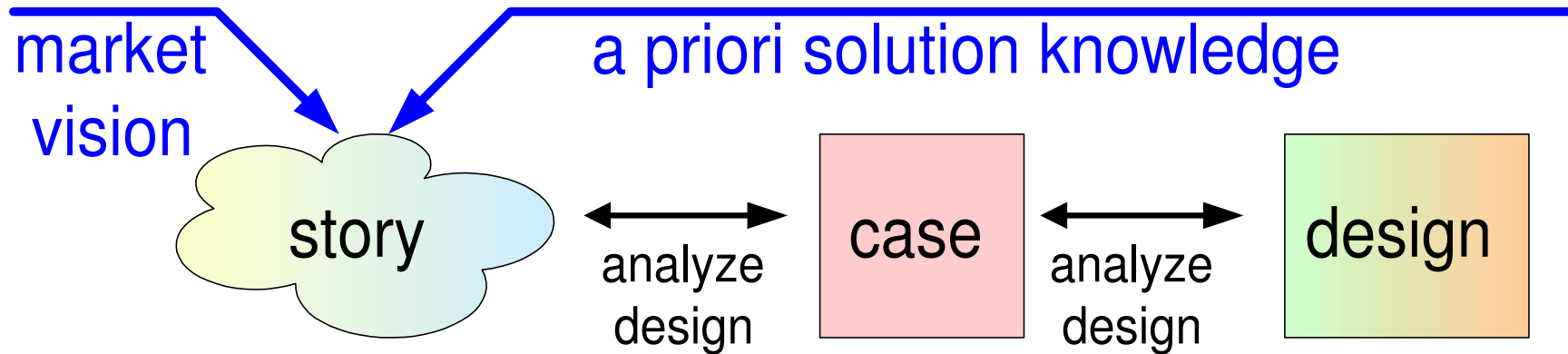
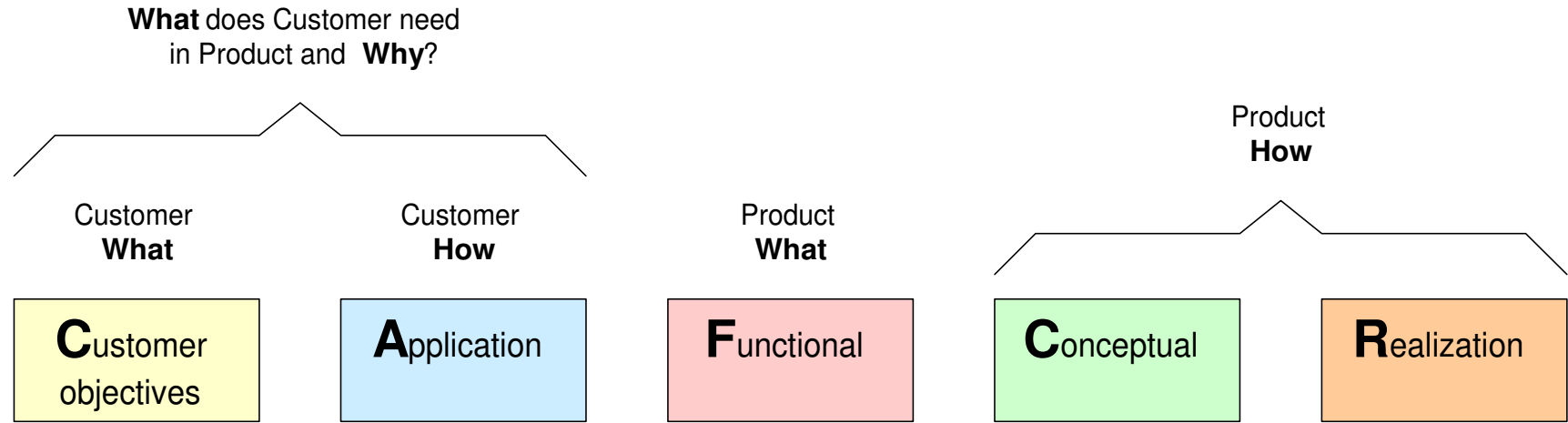
## Distribution

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

February 11, 2012  
status: concept  
version: 1.1



# From story to design



# Example story layout

ca. half a page of plain English text

A day in the life of Bob

bla blah bla, rabarber music  
bla bla composer bla bla  
qwerty30 zepps.

nja nja njet nijppie est quo  
vadis? Pjotr jaleski bla bla  
bla brree fgfg gsg hgrg

mjimm bas engel heeft een  
interessant excuus, lex stelt  
voor om vanavond door te  
werken.

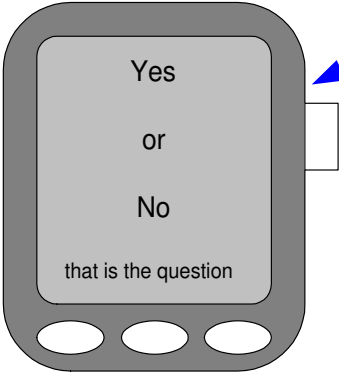
In the middle of the night he  
is awake and decides to  
change the world forever.

The next hour the great  
event takes place:

This brilliant invention will change the world foreverbecause it is so unique and  
valuable that nobody believes the feasibility. It is great and WOW at the same time,  
highly exciting.

Vtables are seen as the sollution for an indirection problem. The invention of Bob will  
obsolete all of this in one incredibke move, which will make him famous forever.

He opens his PDA, logs in and enters his provate secure unqie non trivial  
password, followed by a thorough authentication. The PDA asks for the fingerprint of  
this little left toe and to pronounce the word shit. After passing this test Bob can  
continue.



draft or sketch of  
some essential  
appliance

# Points of attention

---

- purpose
- scope
- viewpoint, stakeholders
- visualization
- size (max 1 A4)
- recursive decomposition, refinement

# Criteria for a good story

---

**C**ustomer objectives • accessible, understandable

**A**pplication

"Do you see it in front of you?"

**C**ustomer objectives • valuable, appealing

**A**pplication

attractive, important

"Are customers queuing up for this?"

**C**onceptual • critical, challenging

**R**ealization

"What is difficult in the realization?"

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

**A**pplication • frequent, no exceptional niche

"Does it add significantly to the bottom line?"

**A**pplication • specific

**F**unctional

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

**C**ustomer  
objectives

**A**pplication

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

**C**onceptual

**R**ealization

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

# Concept Selection, Set Based Design and Late Decision Making

by *Gerrit Muller* Buskerud University College

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

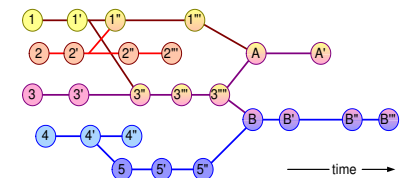
## Abstract

We discuss a systems design approach where several design options are maintained concurrently. In LEAN Product Development this is called set-based design. Conventional systems engineering also promotes the concurrent evaluation of multiple concepts, the so-called concept selection. Finally, LEAN product development advocates to keep options open as long as feasible; the so-called late decision making.

## Distribution

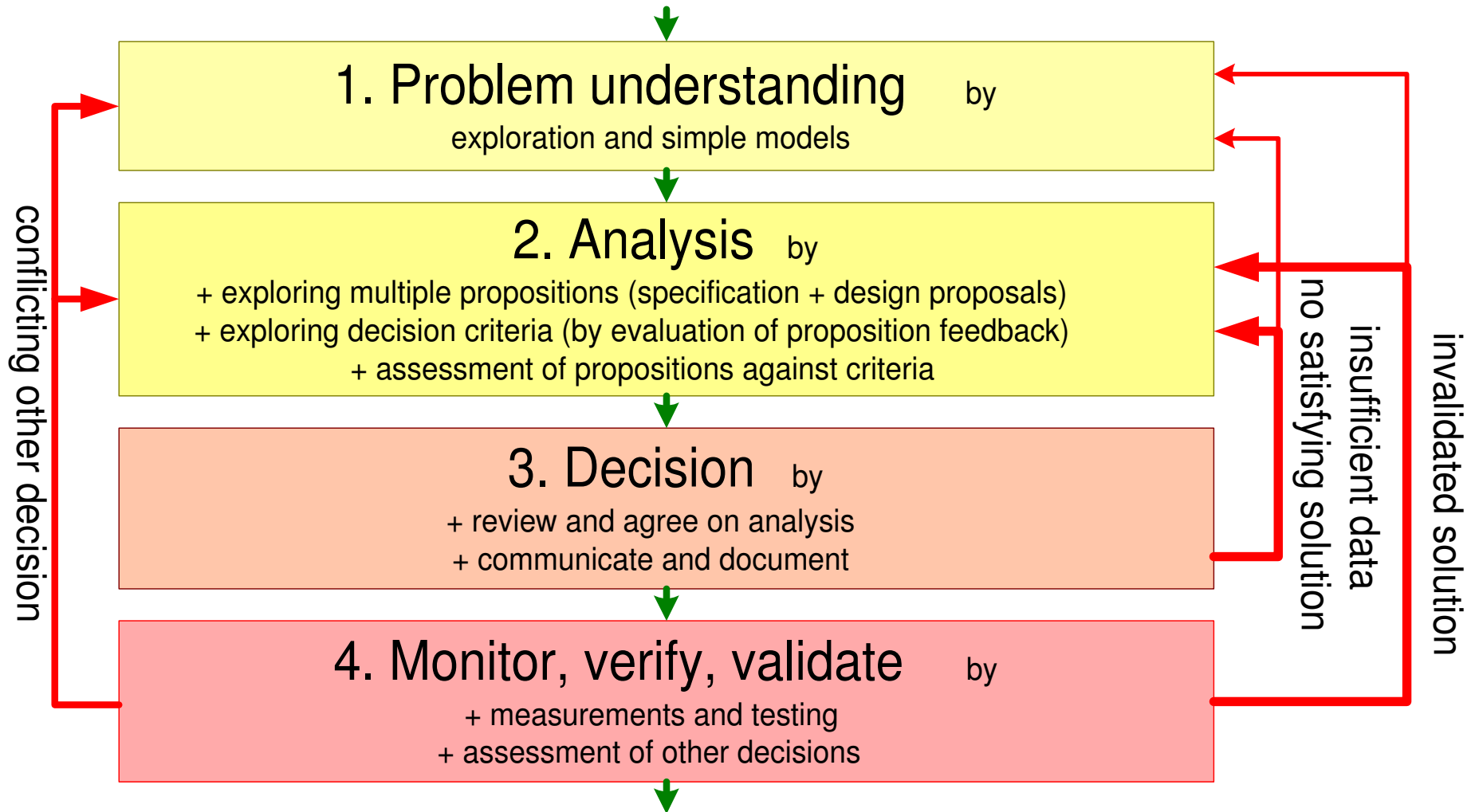
This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution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

February 11, 2012  
status: planned  
version: 0



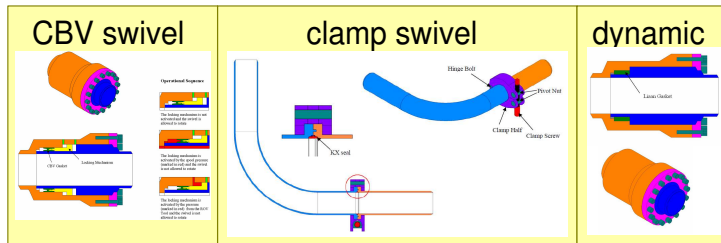
# Problem Solving Approach

vague problem statement



# Examples of Pugh Matrix Application

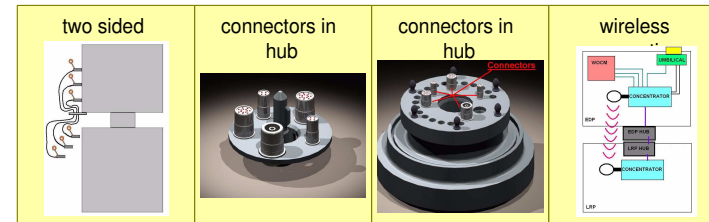
## Swivel concept selection



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

from master paper Halvard Bjørnsen, 2009

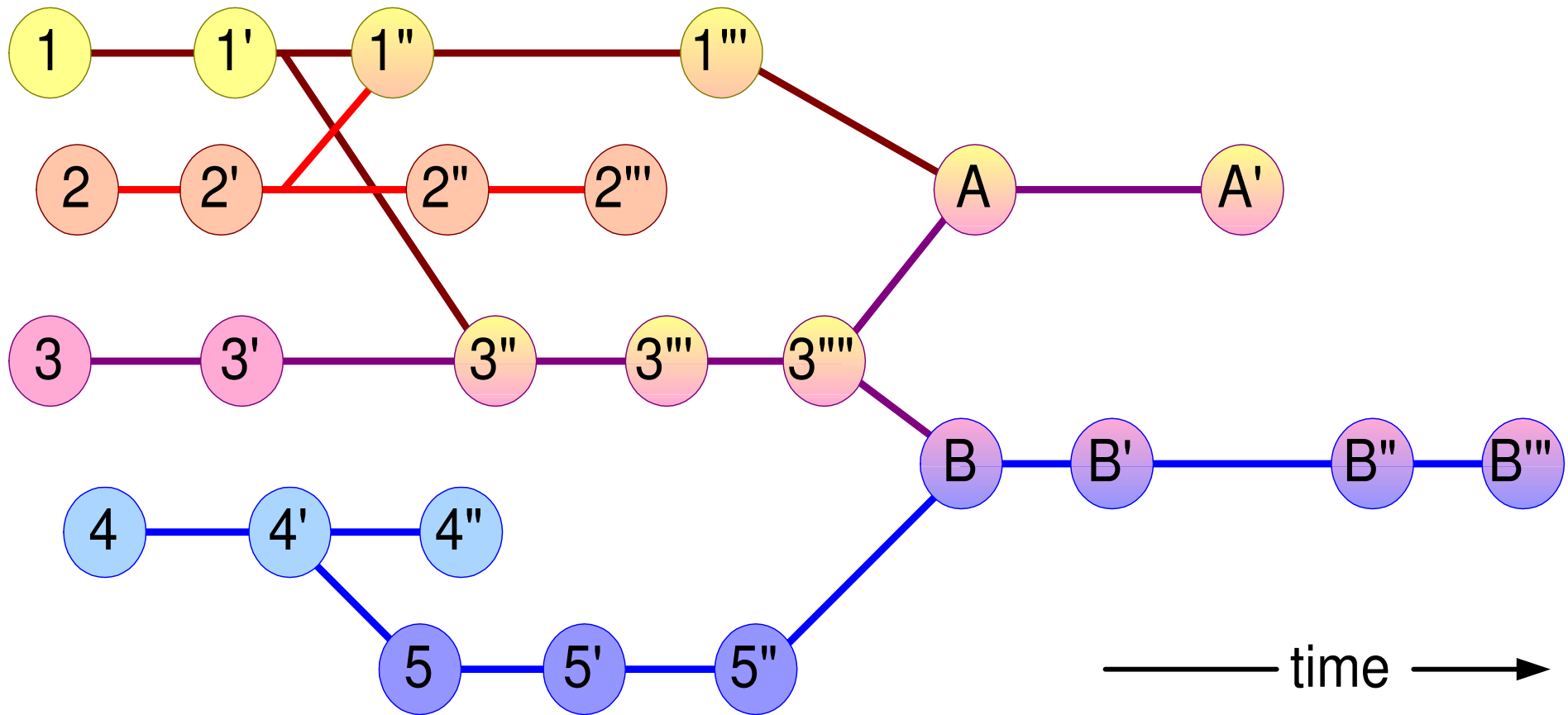
## EDP-LRP connection



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

from master paper Dag Jostein Klever, 2009

# Evolution of Design Options



# Conclusions

---

Evolving multiple concepts increases insight and understanding  
(LEAN product development: set-based design, SE: Pugh matrix)

Articulation of criteria sharpens evaluation

The discussion about the Pugh matrix is more valuable than final  
bottomline summation

Delaying decisions may help to keep options (Lean Product  
Development: late decision making, finance: real options)

# Qualities as Integrating Needles

by *Gerrit Muller* Embedded Systems Institute  
e-mail: `gerrit.muller@embeddedsystems.nl`  
`www.gaudisite.nl`

## Abstract

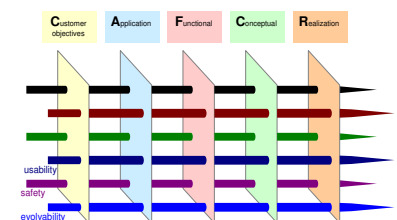
Many stakeholder concerns can be specified in terms of qualities. These qualities can be viewed from all 5 “CAFCE” viewpoints. In this way qualities can be used to relate the views to each other.

The meaning of qualities for the different views is described. A checklist of qualities is provided as a means for architecting. All qualities in the checklist are described briefly.

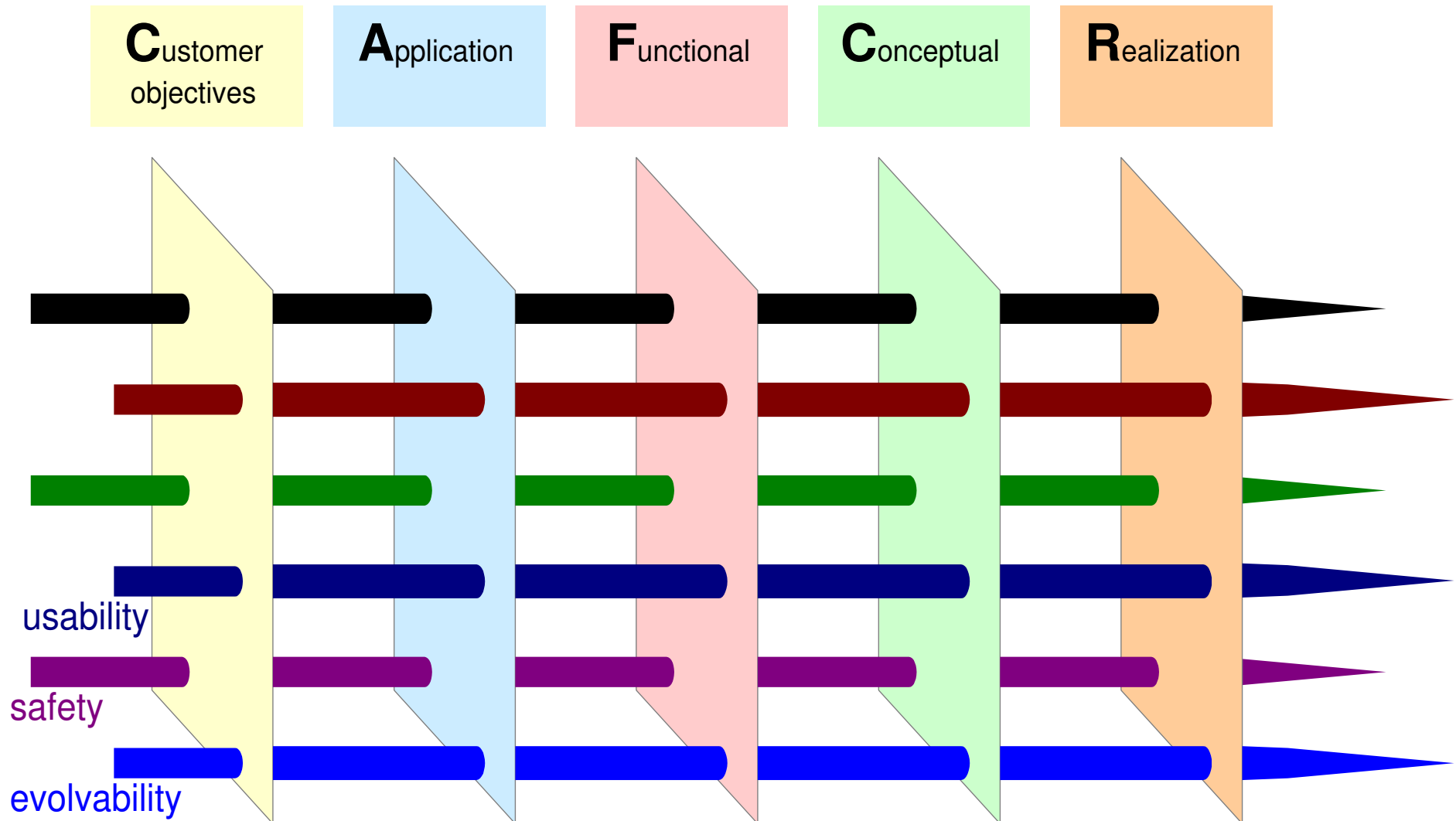
## Distribution

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

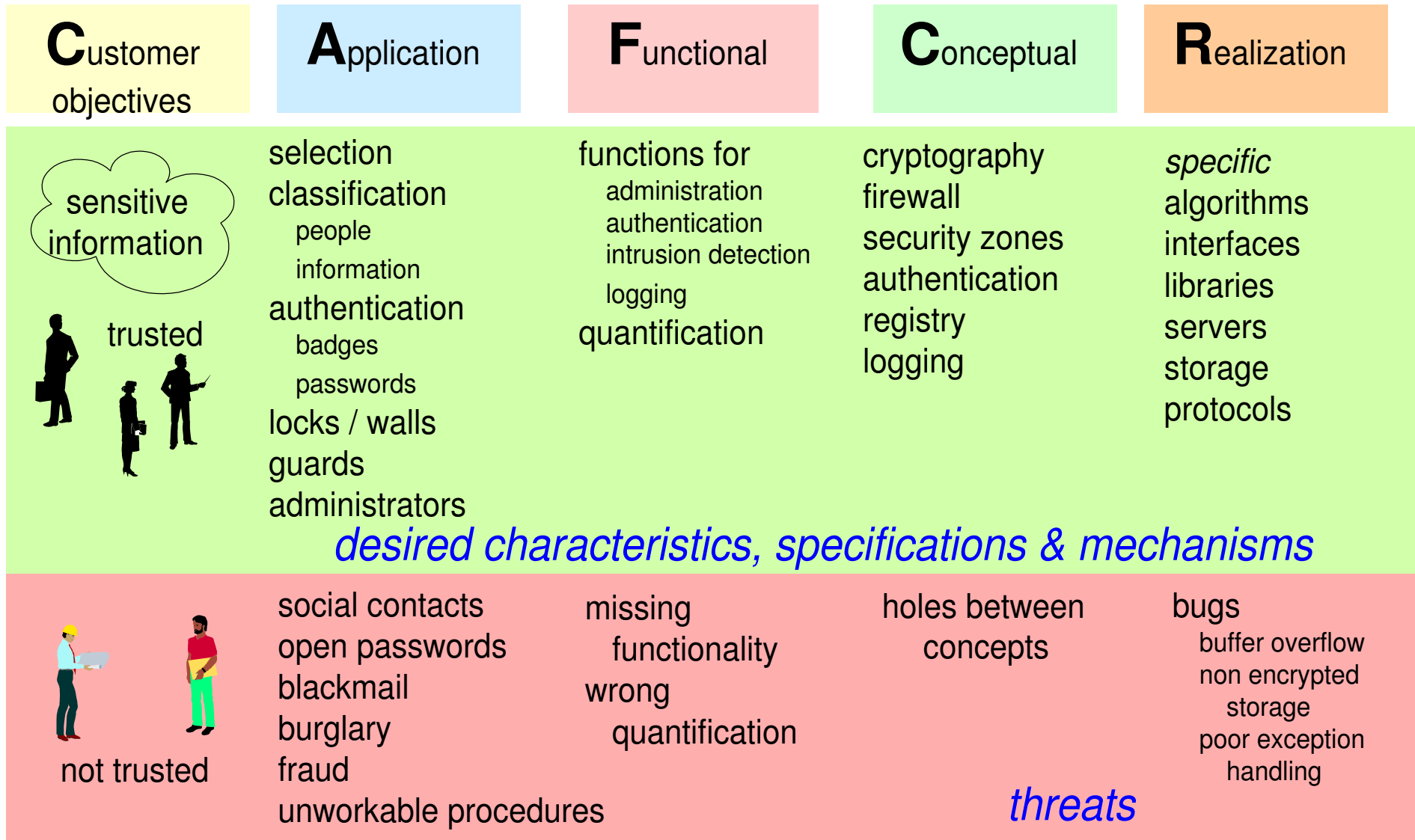
February 11, 2012  
status: finished  
version: 1.3



# Quality needles as generic integrating concepts



# Security as example through all views



# Quality Checklist

---

## usable

usability  
attractiveness  
responsiveness  
image quality  
wearability  
storability  
transportability

## dependable

safety  
security  
reliability  
robustness  
integrity  
availability

## effective

throughput or  
productivity

## interoperable

connectivity  
3<sup>rd</sup> party extendible

## liable

liability  
testability  
traceability  
standards compliance

## efficient

resource utilization  
cost of ownership

## consistent

reproducibility  
predictability

## serviceable

serviceability  
configurability  
installability

## future proof

evolvability  
portability  
upgradeability  
extendibility  
maintainability

## logistics friendly

manufacturability  
logistics flexibility  
lead time

## ecological

ecological footprint  
contamination  
noise  
disposability

## down to earth attributes

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

# System Partitioning Fundamentals

by *Gerrit Muller* Buskerud University College

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

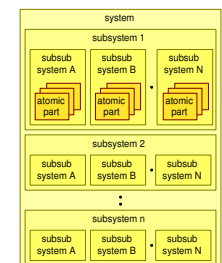
## Abstract

The fundamental concepts and approach system partitioning are explained. We look at physical decomposition and functional decomposition in relation to supply chain, lifecycle support, project management, and system specification and design.

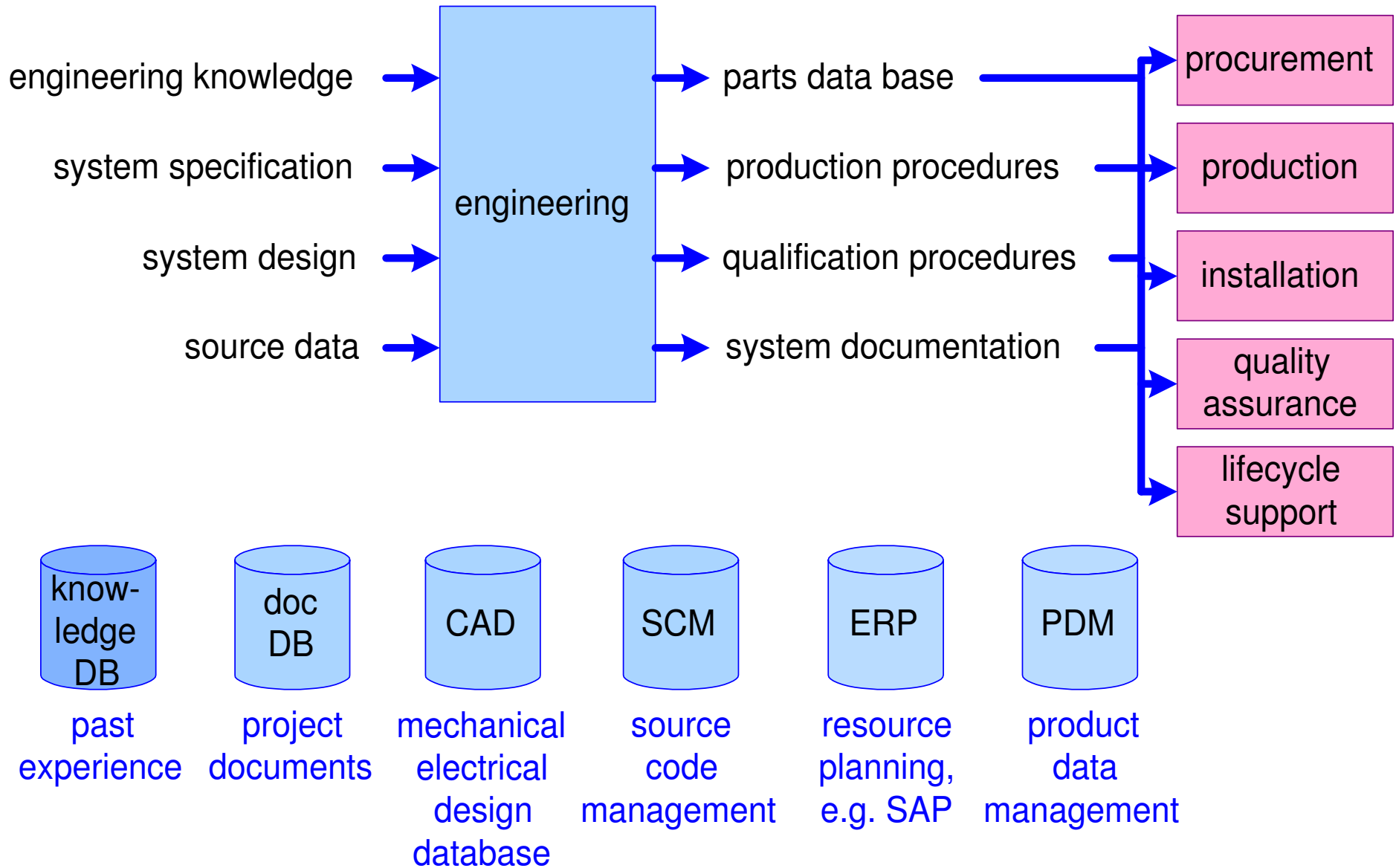
## Distribution

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

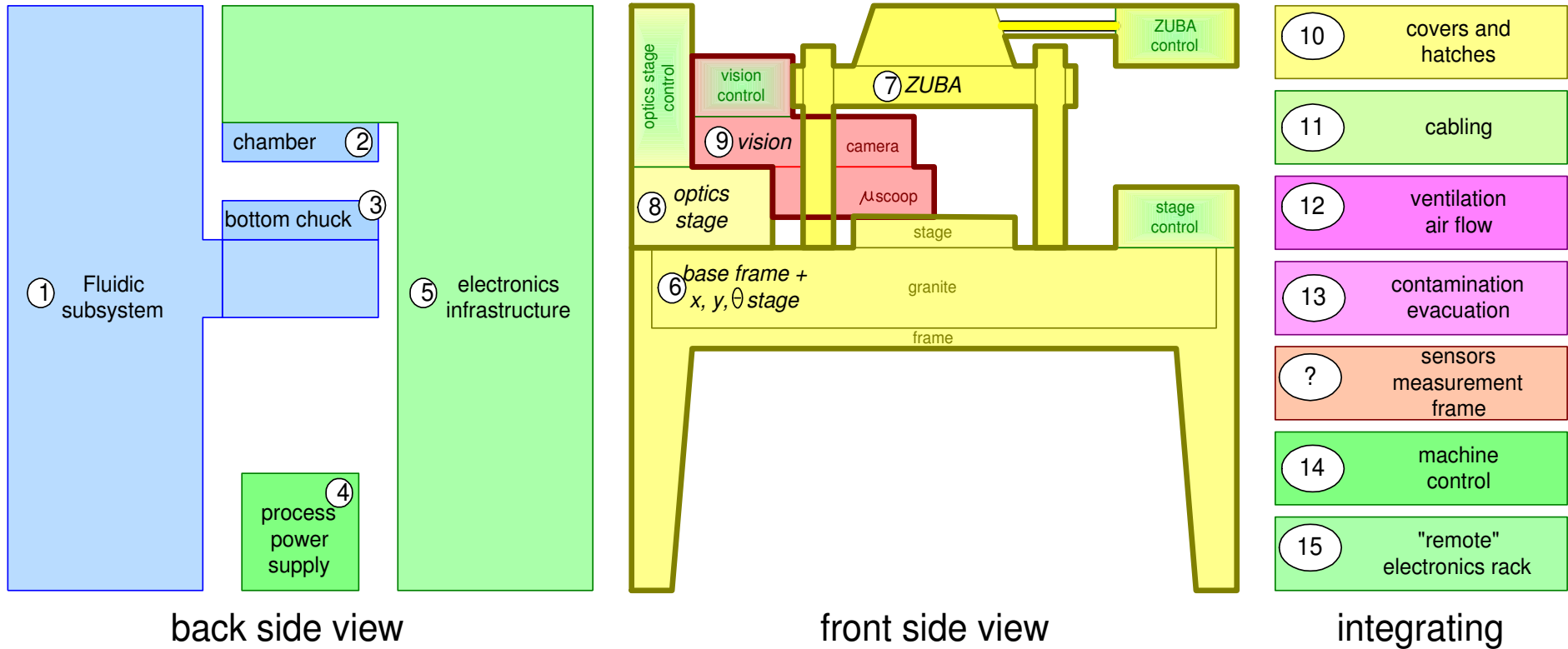
February 11, 2012  
status: preliminary  
draft  
version: 0.1



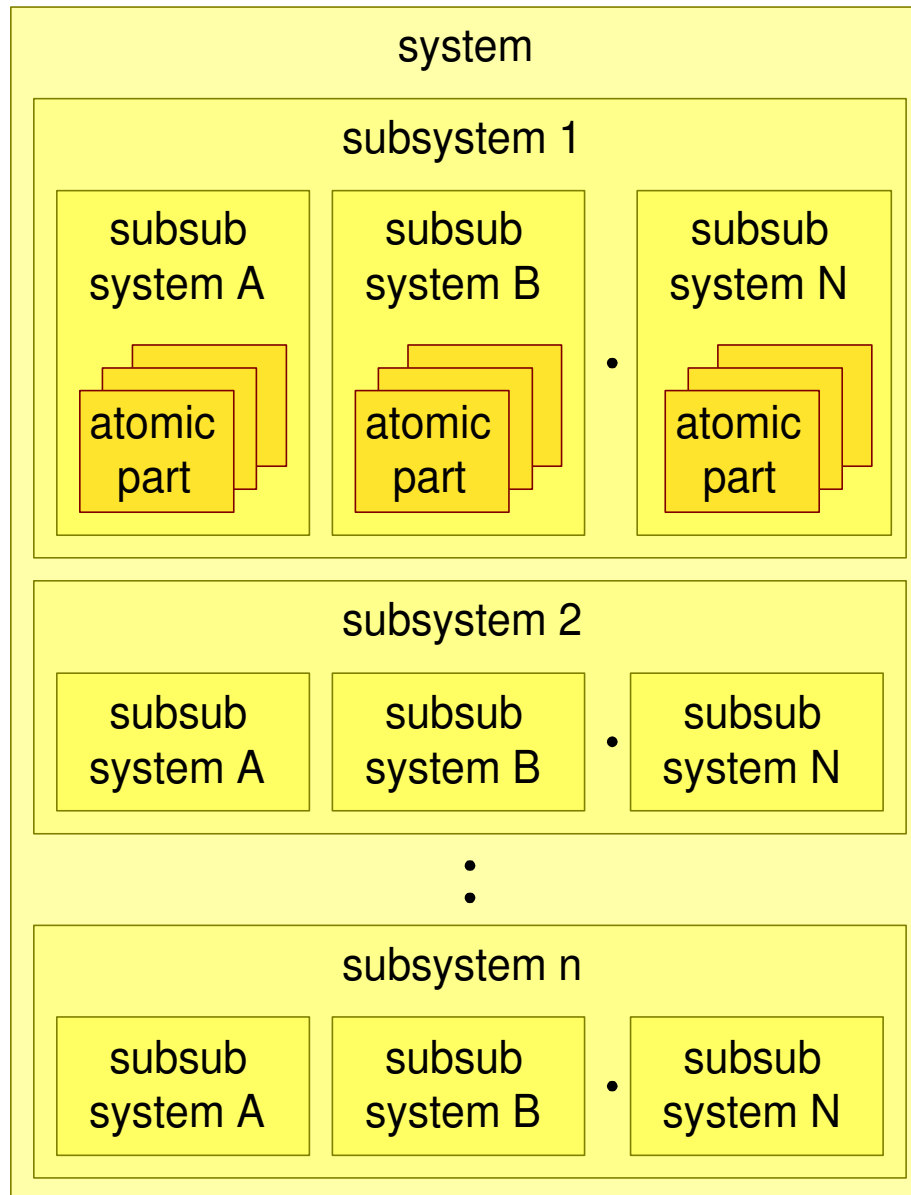
# Engineering



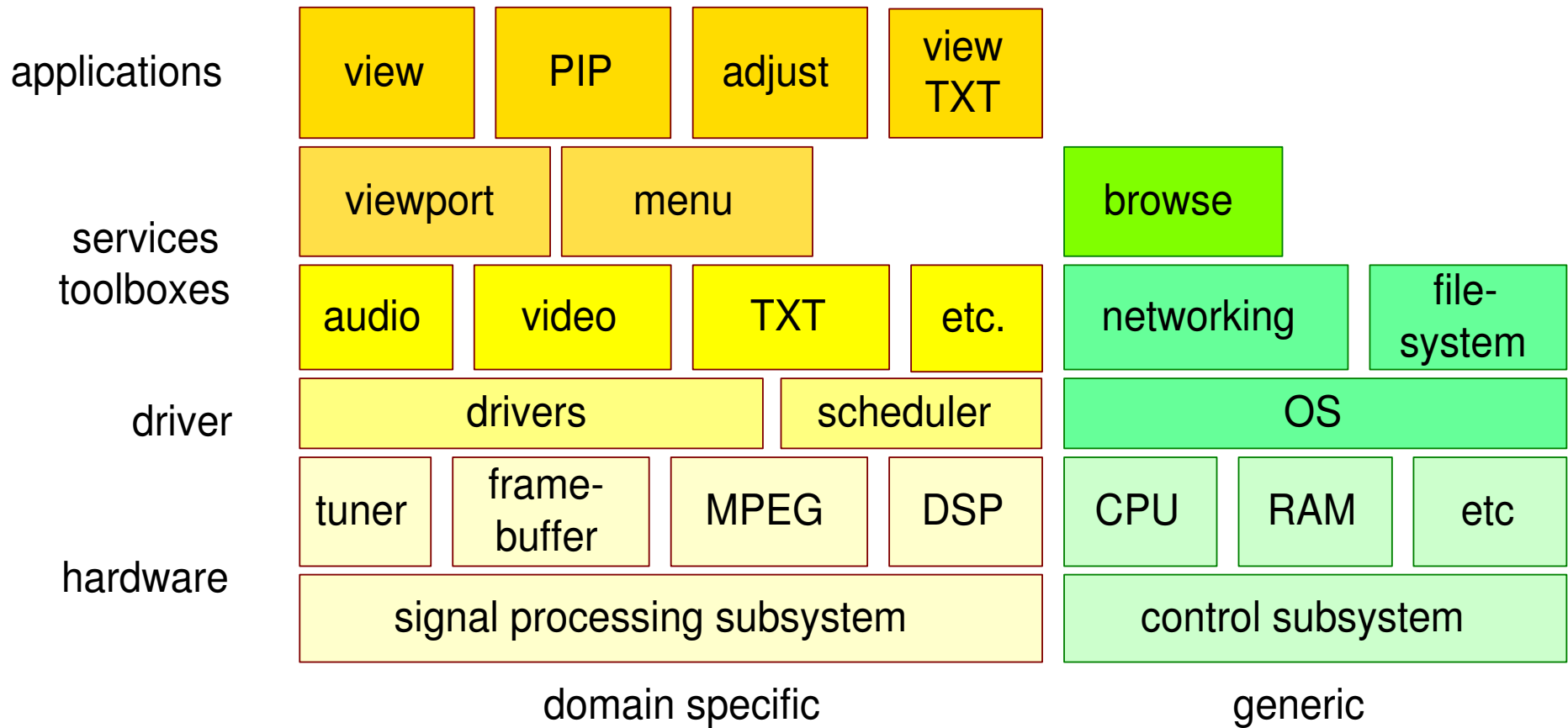
# Example Physical Decomposition



# Partitioning is Applied Recursively



# Software plus Hardware Decomposition



# Guidelines for Partitioning

---

the part is cohesive

functionality and technology belongs together

the coupling with other parts is minimal

minimize interfaces

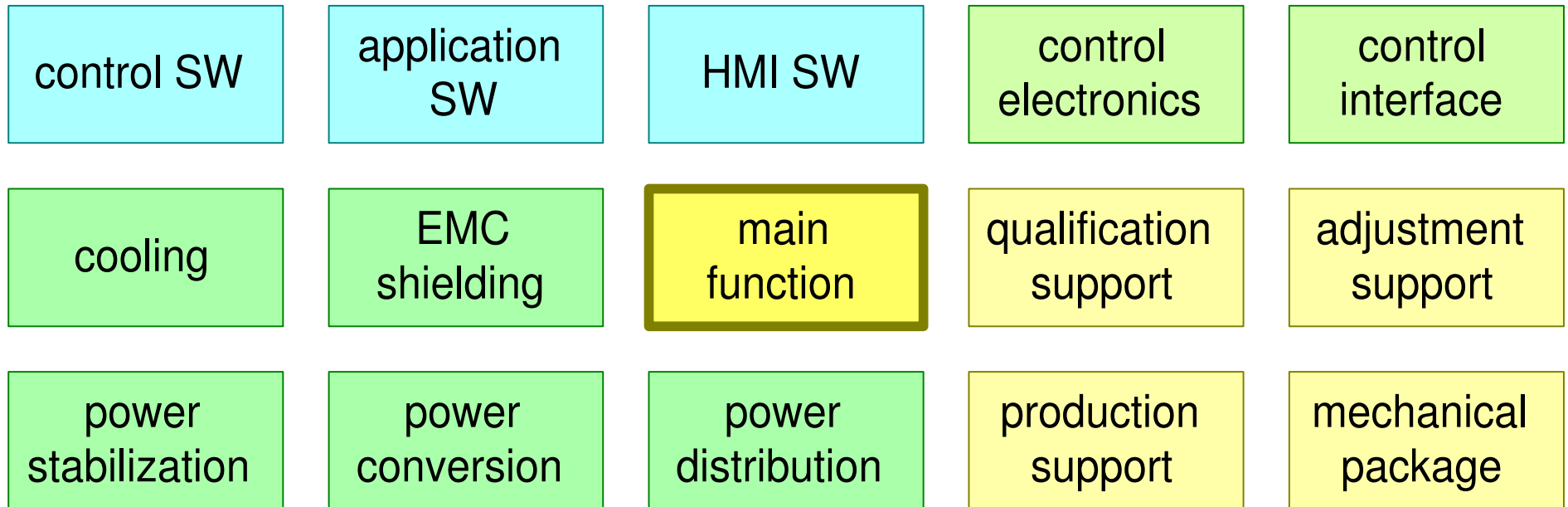
the part is selfsustained for production and qualification

can be in conflict with cost or space requirements

clear ownership of part

e.g. one department or supplier

# How much self-sustained?



How self sustained should a part be?

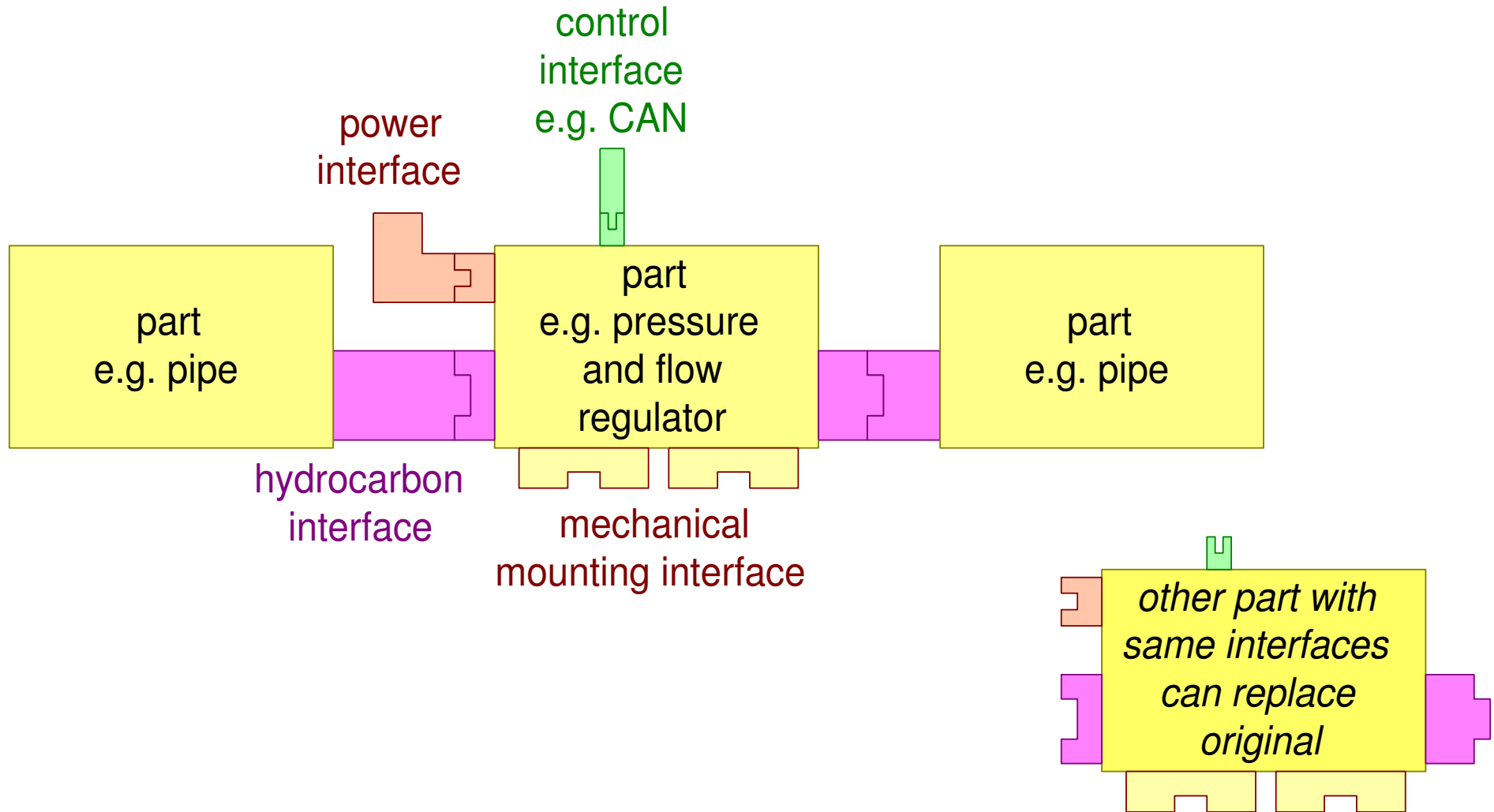
trade-off:

cost/speed/space  
optimization



logistics/lifecycle/production  
flexibility  
clarity

# Decoupling via Interfaces



# The Ideal Modularity

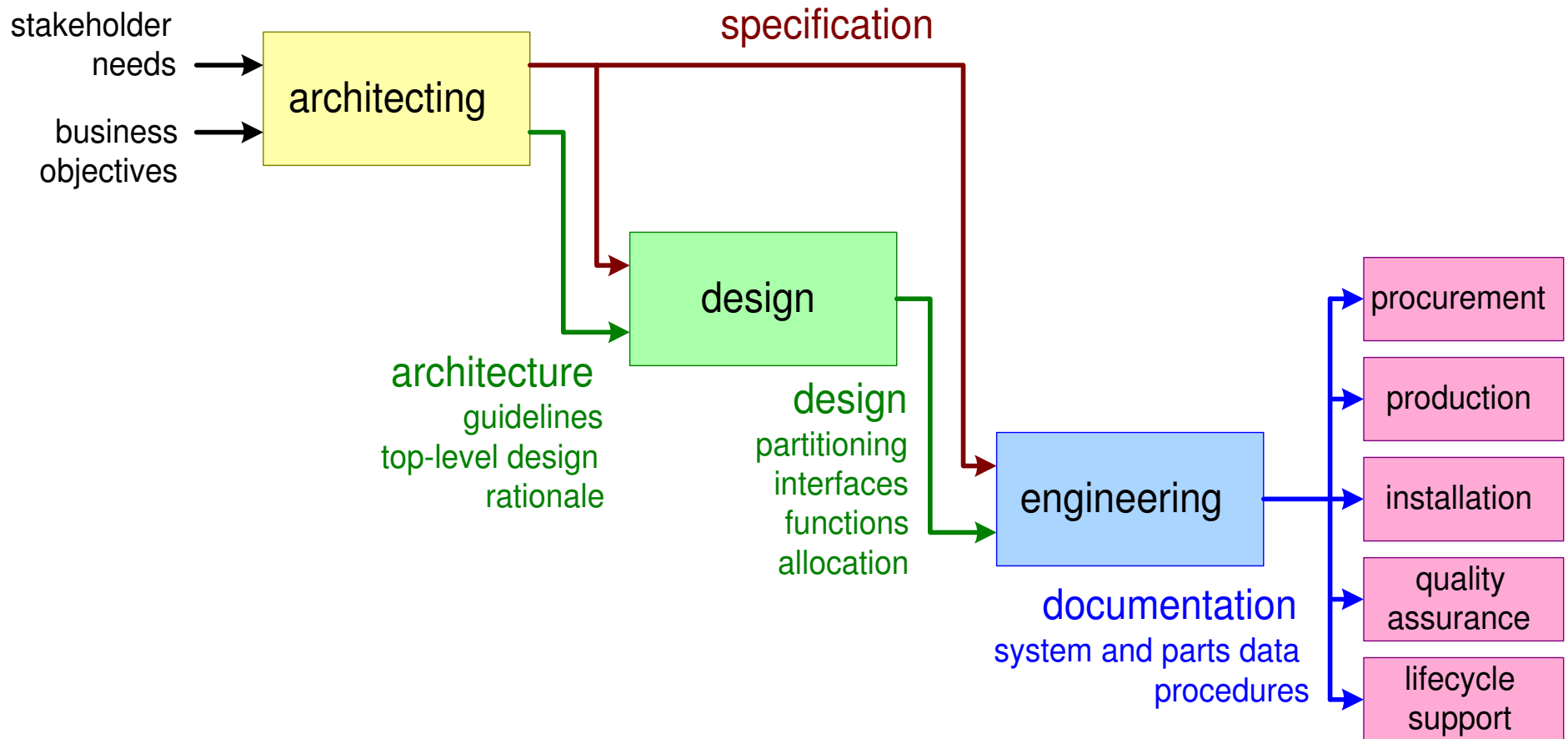
---

System is composed

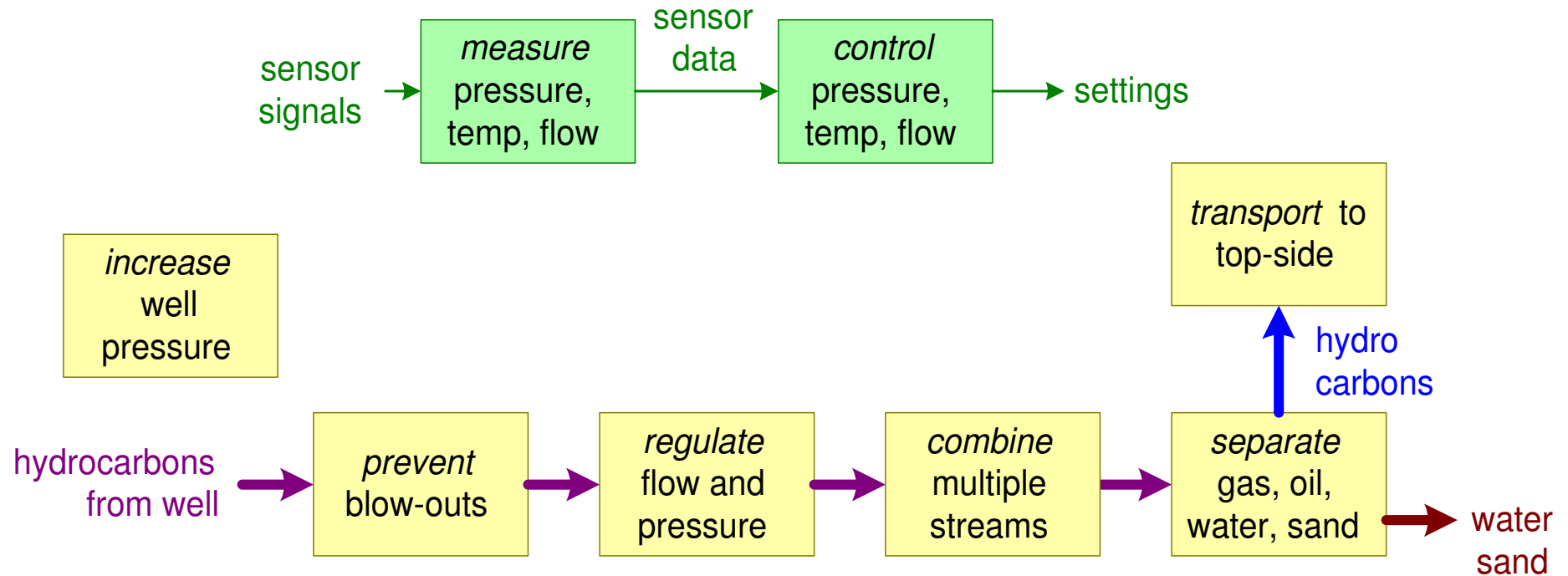
by using standard interfaces

limited catalogue of variants (e.g. cost performance points)

# System Creation



# Simplistic Functional SubSea Example



# Functional Decomposition

---

How does the system work and operate?

Functions describe *what* rather than *how*.

Functions are *verbs*.

Input-Process-Output paradigm.

Multiple kinds of flows:

- physical (e.g. hydrocarbons)

- information (e.g. measurements)

- control

At lower level one part  $\approx$  one function

- pump pumps, compressor compresses, controller controls

At higher level functions are complex interplay of physical parts

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

# Quantification

Size 2.4m \* 0.7m \* 1.3m

Weight 1450 Kg

Cost 30000 NoK

Reliability MTBF 4000 hr

Throughput 3000 l/hr

Response time 0.1 s

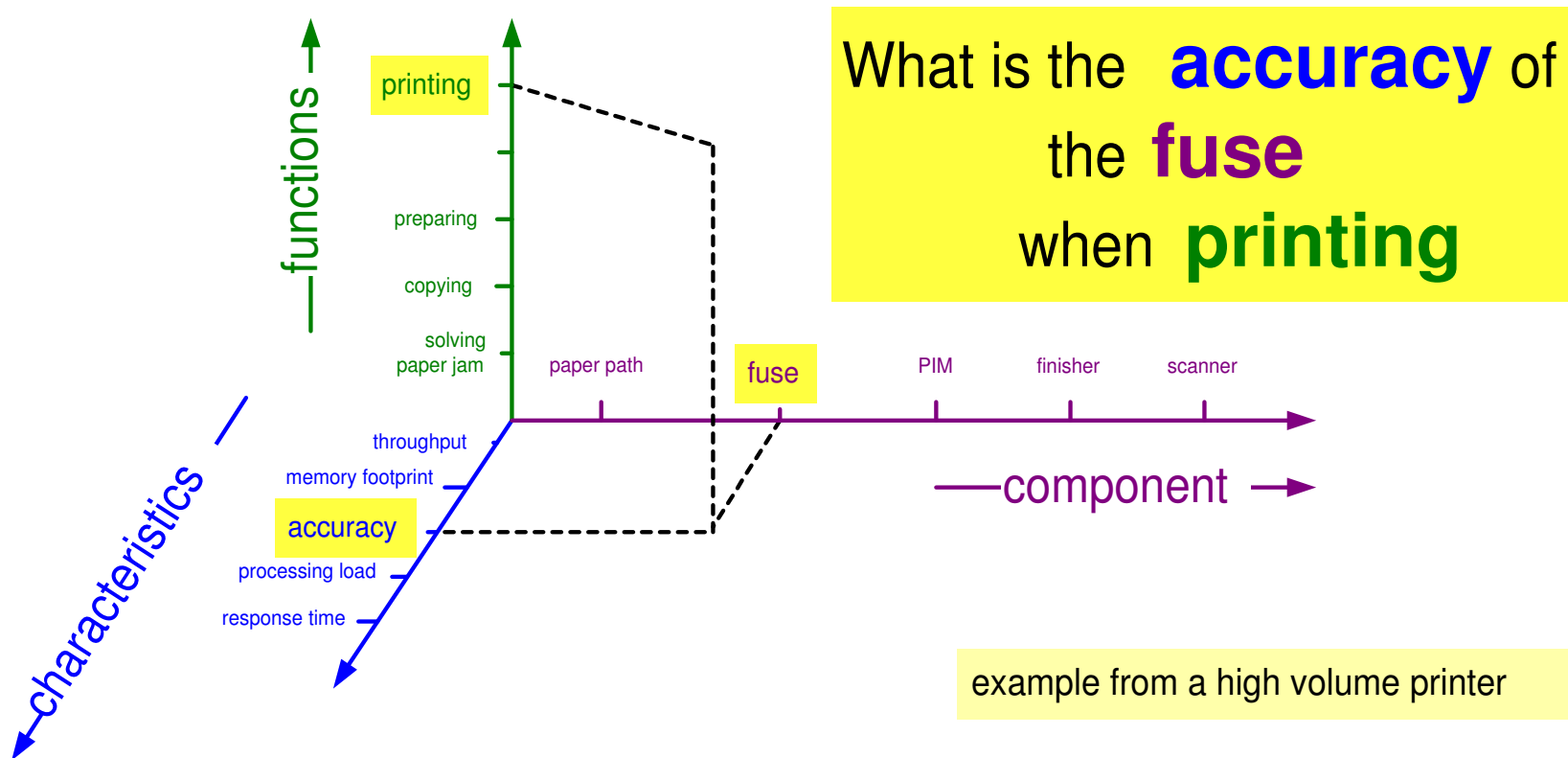
Accuracy +/- 0.1%

*many characteristics  
of a system, function or part  
can be quantified*

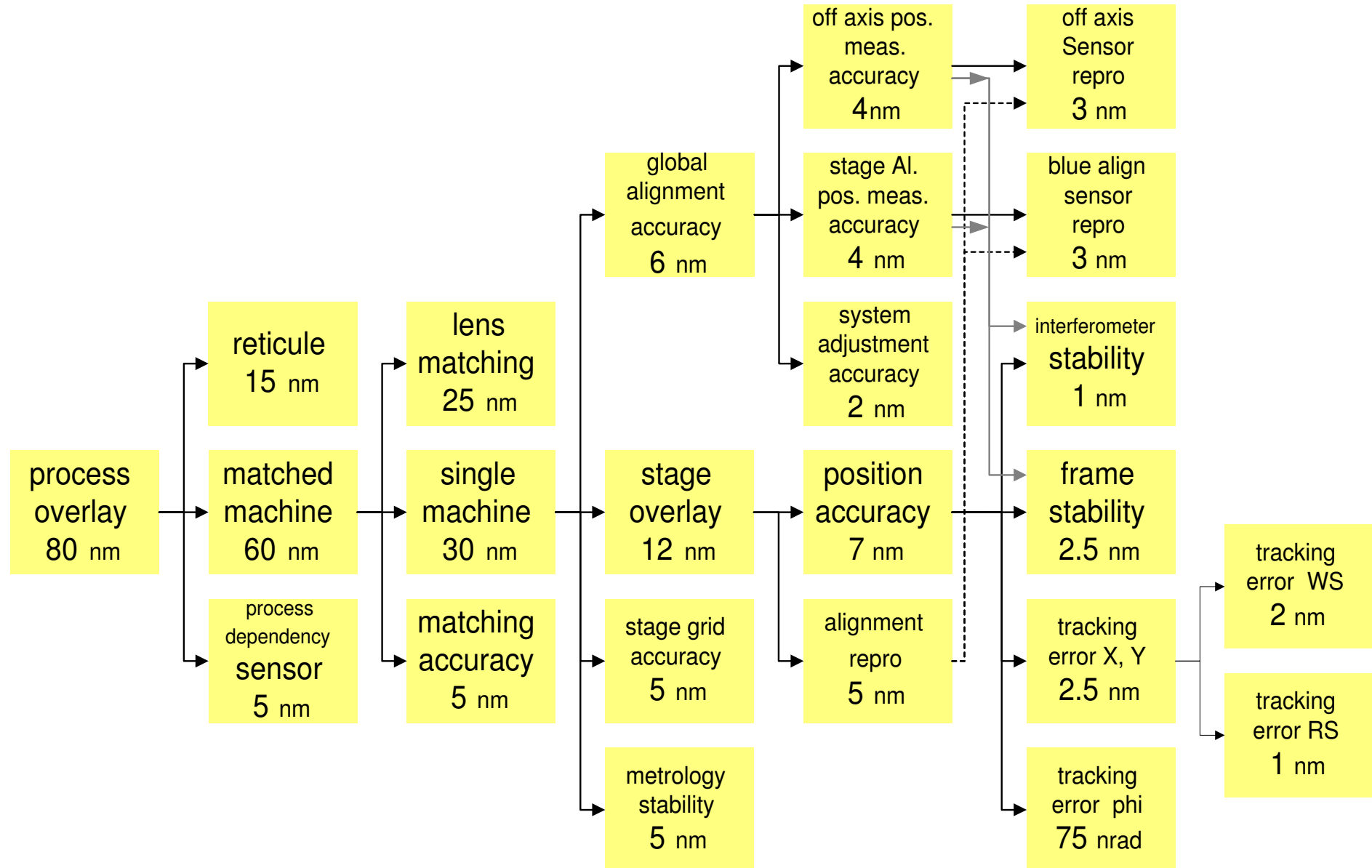
*Note that quantities  
have **unit***

# Question Generator

How about the **<characteristic>**  
of the **<component>**  
when performing **<function>**?



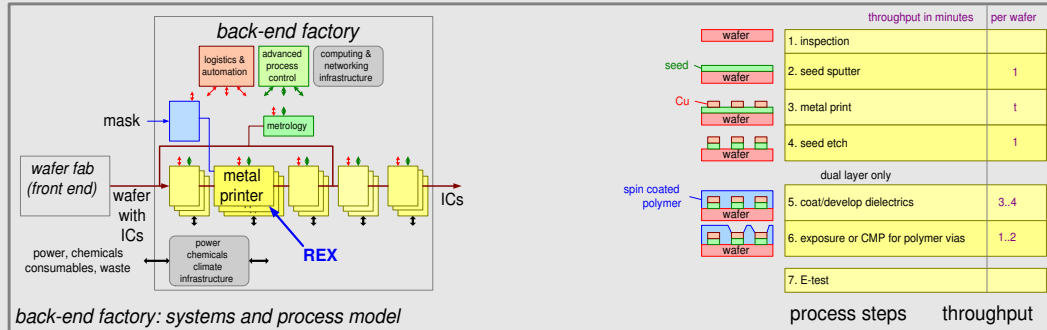
# Example Technical Budget



# Example of A3 overview

## A3 architecture overview of the Metal Printer

(all numbers have been removed for competitive sensitivity)



author	Gerrit Muller	scope	system and supersystem
version	0.1	status	preliminary draft
date last update	August 3, 2010		

*Document meta-information*

