

CTT Course System Architecting SARCH for management teams

by *Gerrit Muller* Embedded Systems Institute

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

`www.gaudisite.nl`

Abstract

Information about the course System Architecting for management teams.

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

logo
TBD

Course System Architecting for Management Teams

by *Gerrit Muller* Embedded Systems Institute

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

`www.gaudisite.nl`

Abstract

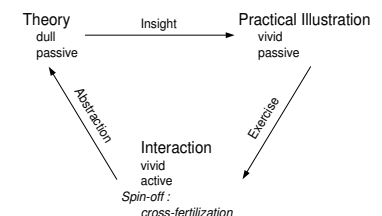
This article describes the condensed version the course System Architecture by the Center for Technical Training CTT. Trainer is the author of this article Gerrit Muller. At this moment this course is only accessible for Philips Employees.

The course is set up to make the art of system architecting more accessible. The course will address 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, skills, and psycho social factors.

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: 0.1

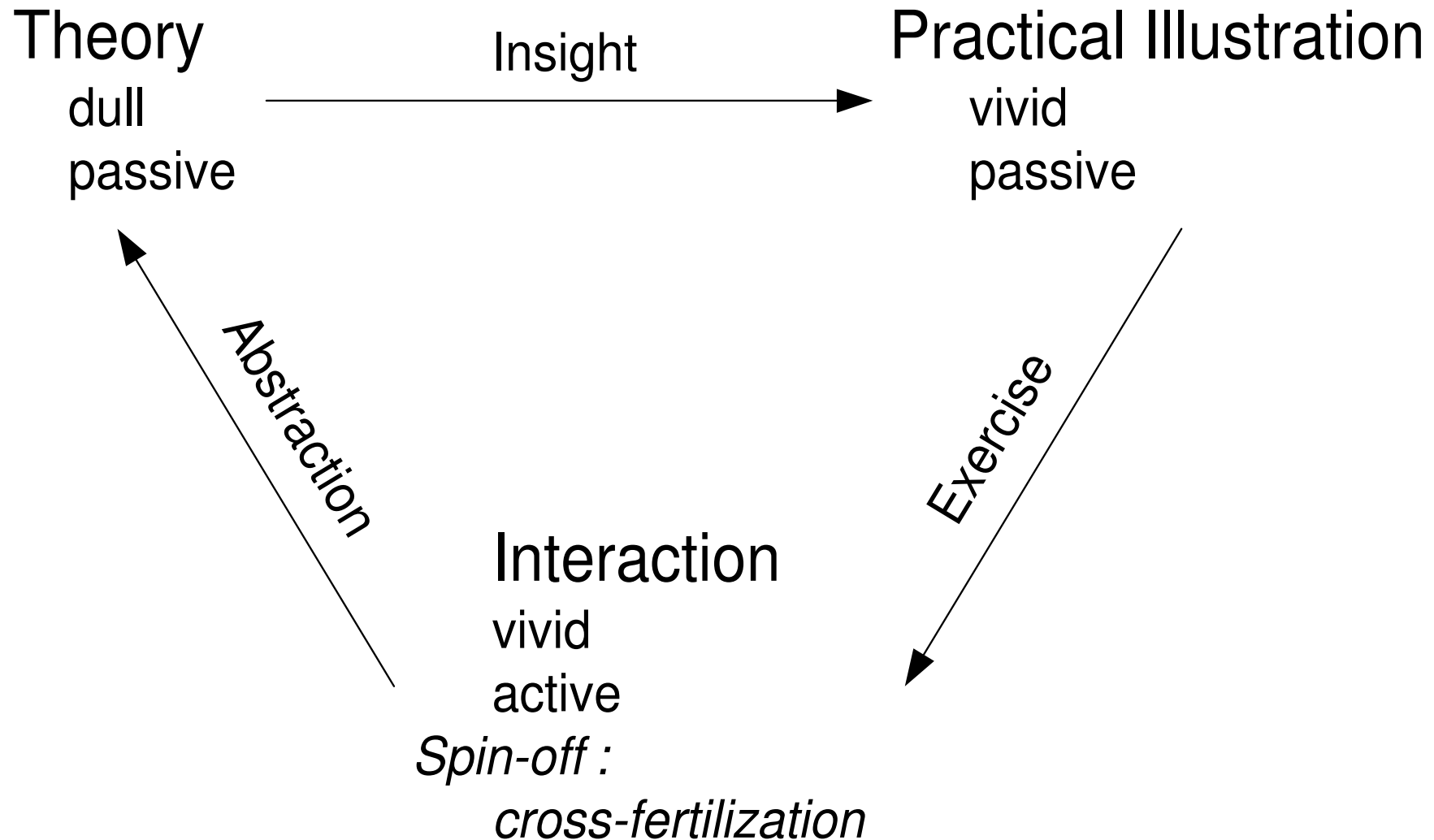


Program

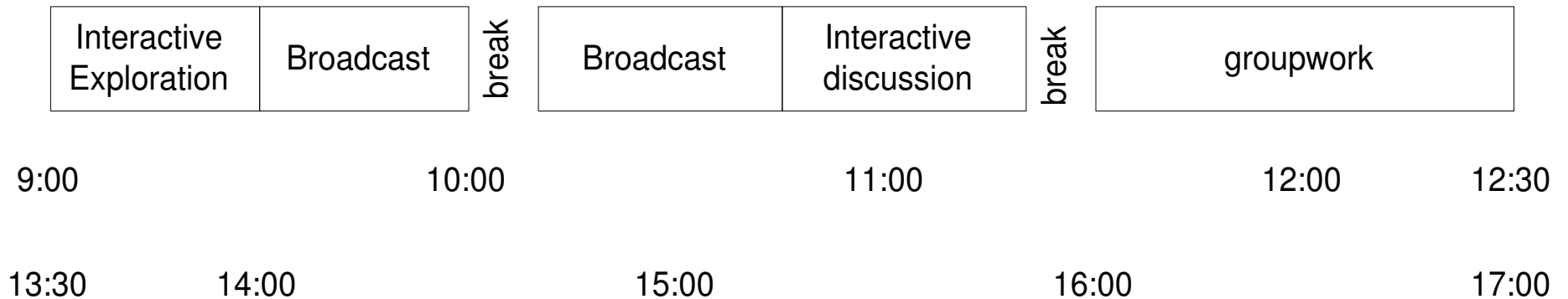
session	subject
day 1 morning	positioning the System Architecture Process Product Creation Process product families, generic developments
day 1 afternoon	role and task of the system architect profile of the system architect documentation, reviewing and other supportive processes
day 2 morning	requirements capturing, roadmapping
day 2 afternoon	HRM aspects; selection, appraisal, career path, etcetera wrap up, expectations, how to continue, evaluation

- 1 Make a map of the operational organization, from portfolio down to components, with specific products, names and roles; Discuss the relations in one of the core teams.
- 2 Role play, marketing manager + projectleader + system architect + observer; prepare initial product definition (=business relevance+specification+critical design issues+plan indication)
- 3 Determine Requirements and key drivers and show the relationship
- 4 Show the roadmap as far as known now

Structure



Timing Template of one subject



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 plurality.

- 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.

The Gaudí Project

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

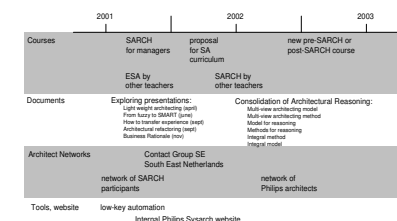
Abstract

The Gaudí project is described. The goals of the project, the way of working, and an outline for the period 2001 to 2003. The deliverables in terms of documents are positioned by means of a two-dimensional map. Courses based on the Gaudí material are described. The current status of the courses is given.

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.

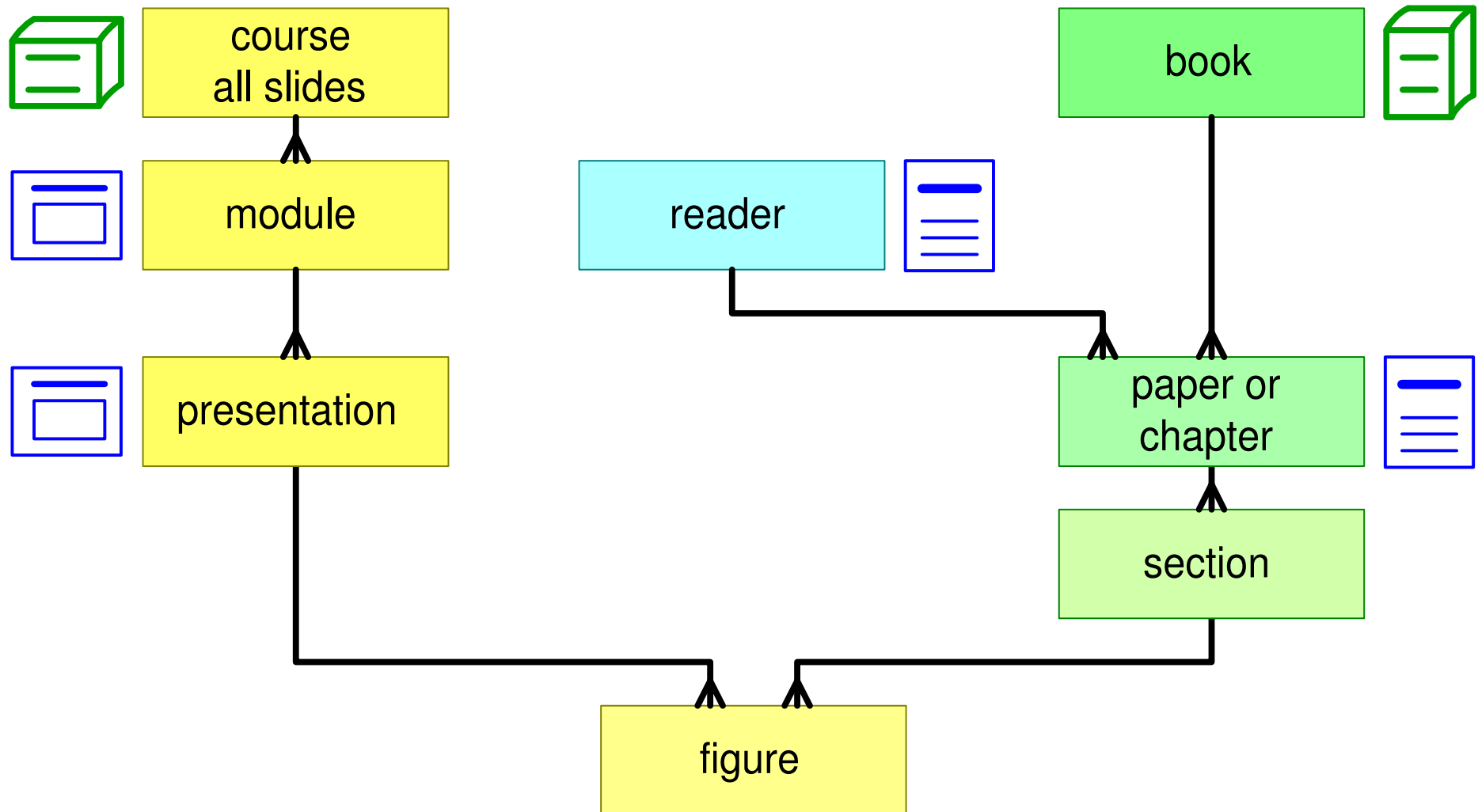
February 11, 2012
status: draft
version: 3.0



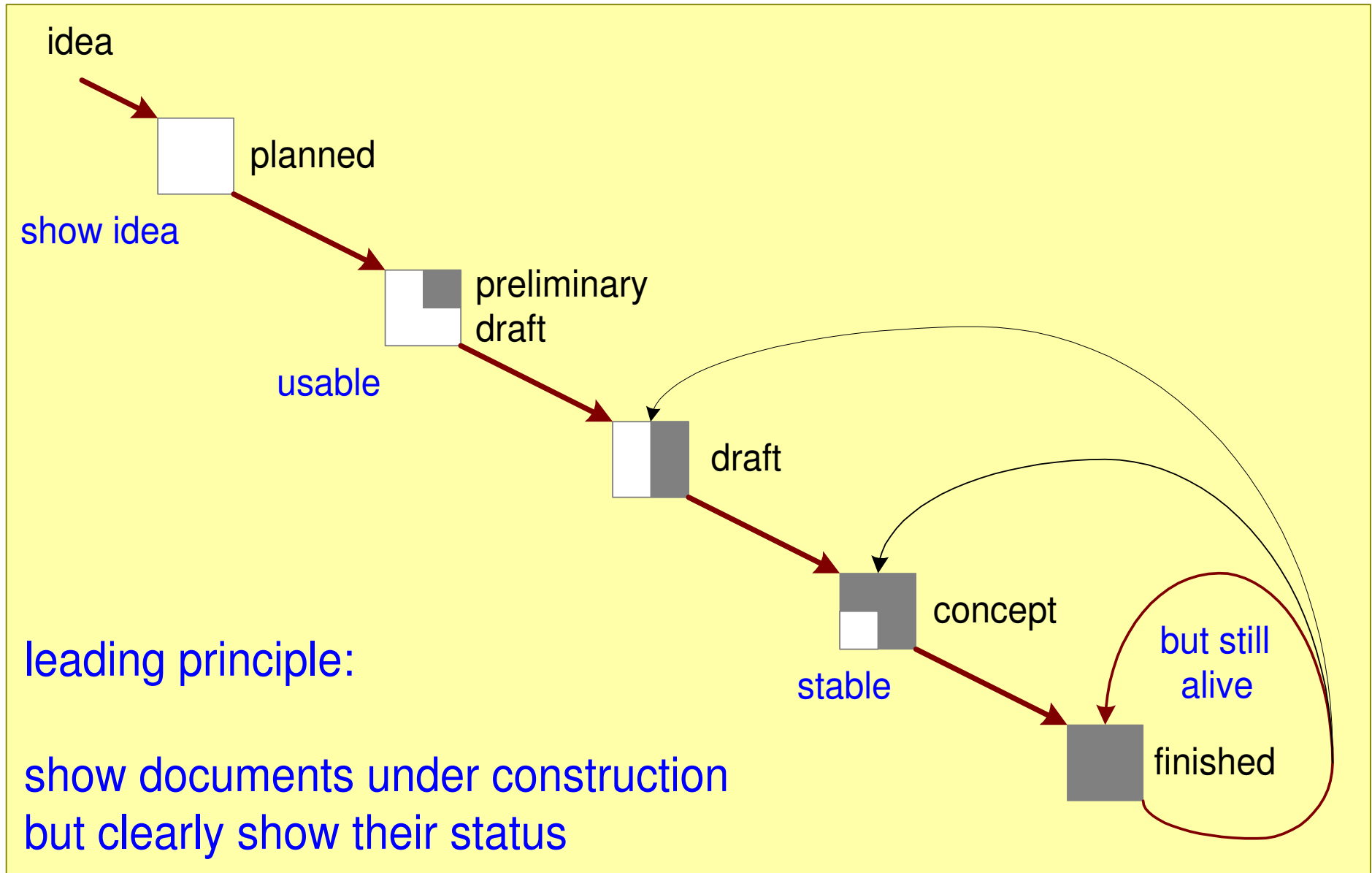
Goals of the Gaudí Project

- Consolidate existing Systems Architecting Methods
evaluate, reflect, generalize
- Make the Systems Architecting art more accessible
case descriptions
- Enable the education of (future) System Architects
curriculum, course material
- Research new or improved Systems Architecting Methods
industry as laboratory

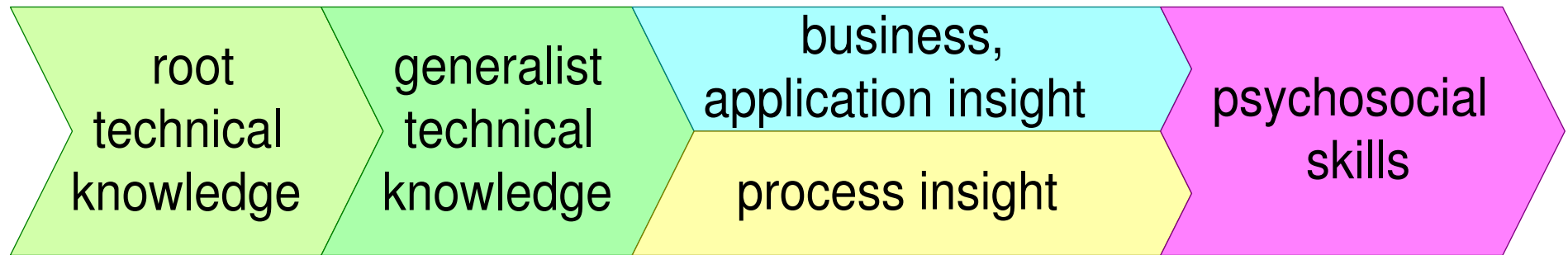
Modular approach



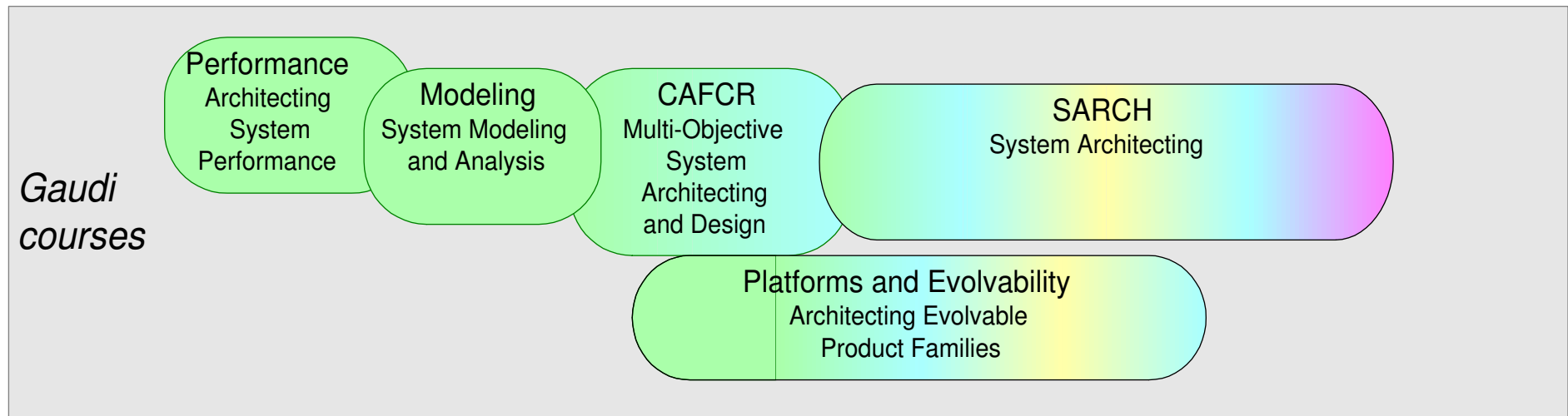
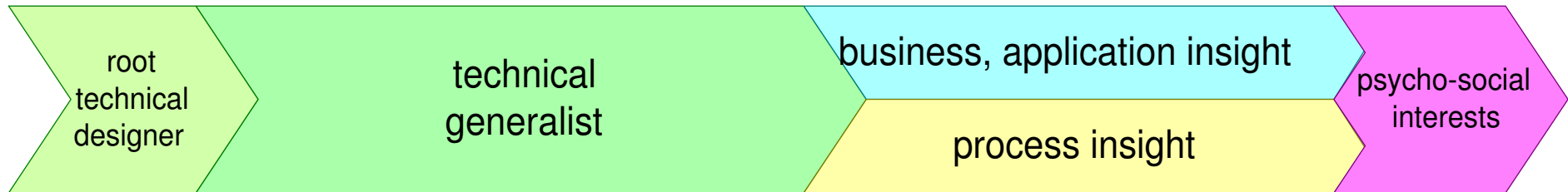
Show Early to Get Feedback



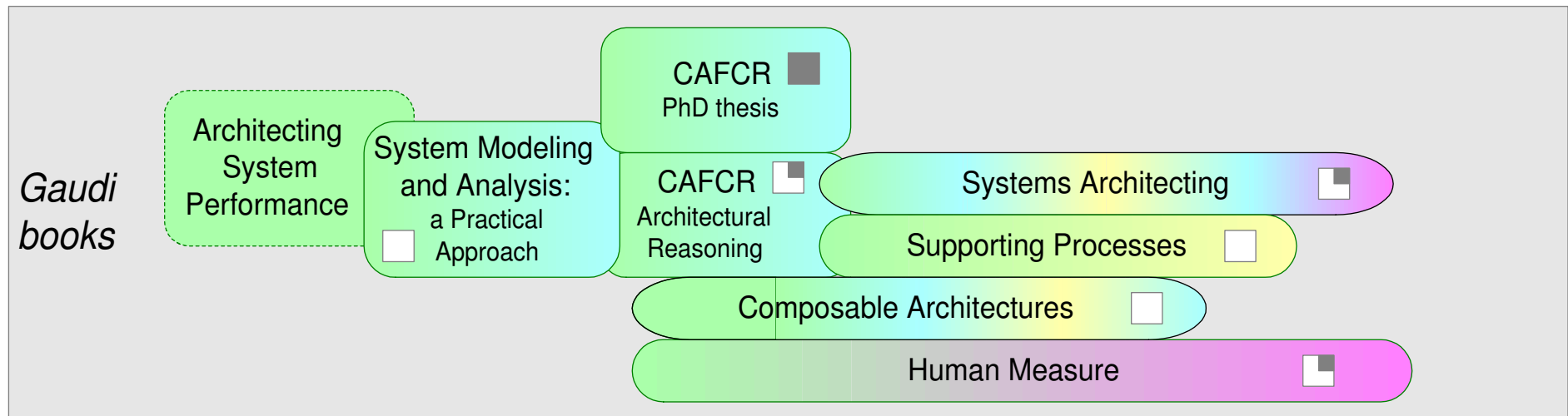
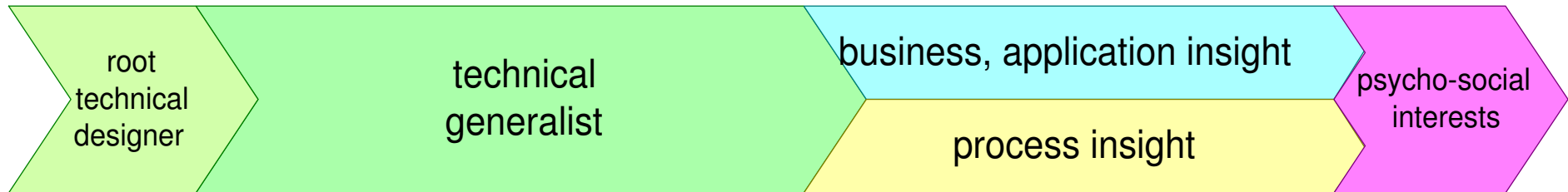
Growth of the System Architect



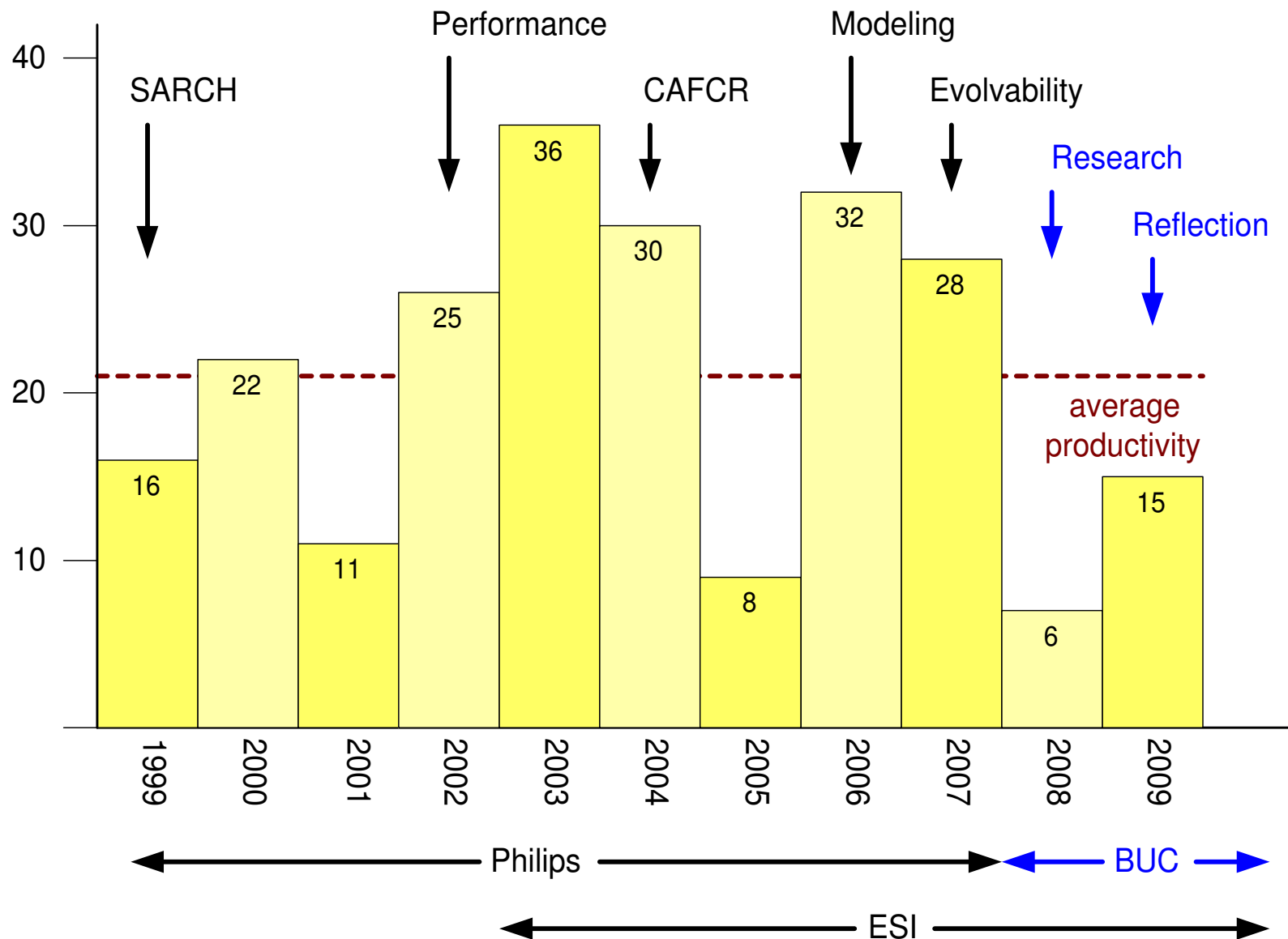
Positioning Courses



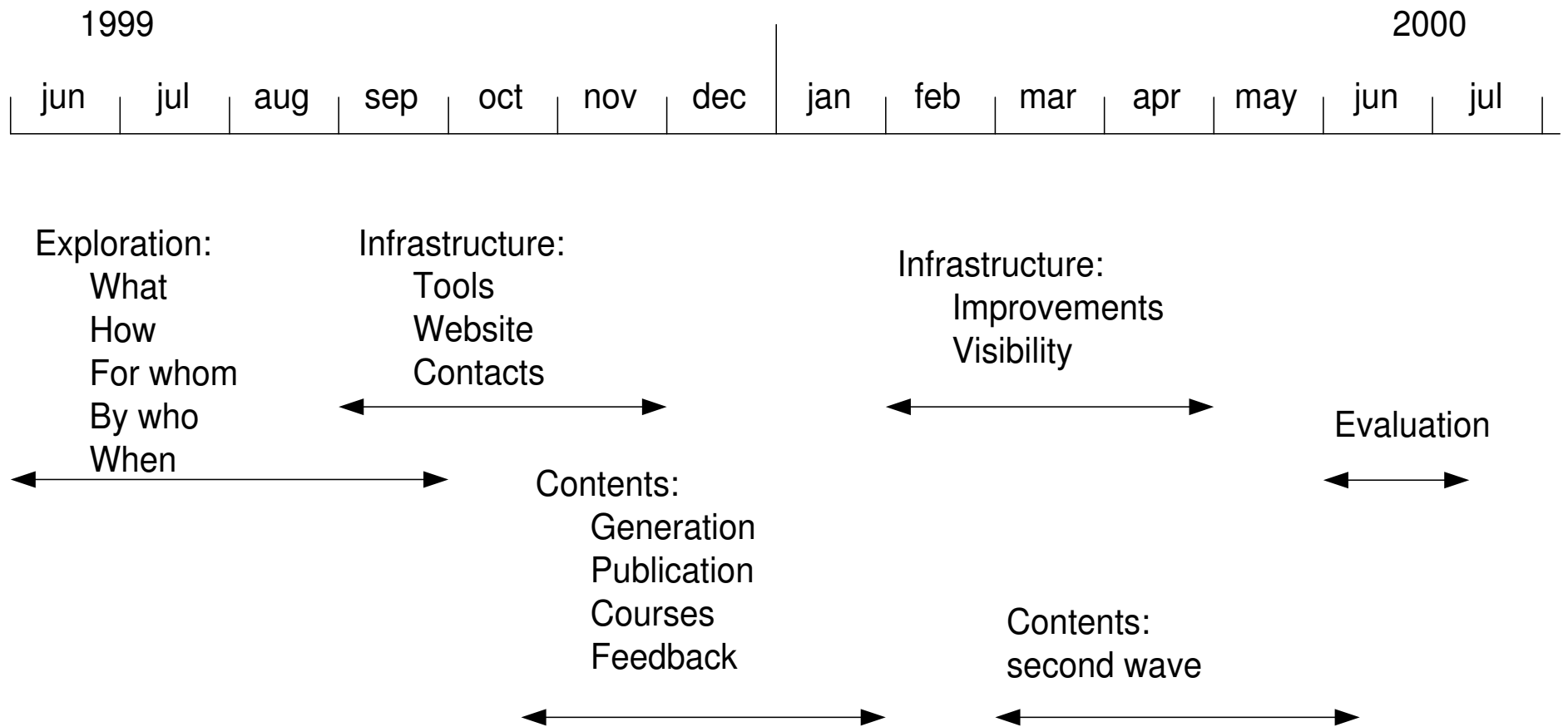
Positioning Books



Productivity: number of new entries



Concurrent Incremental Approach

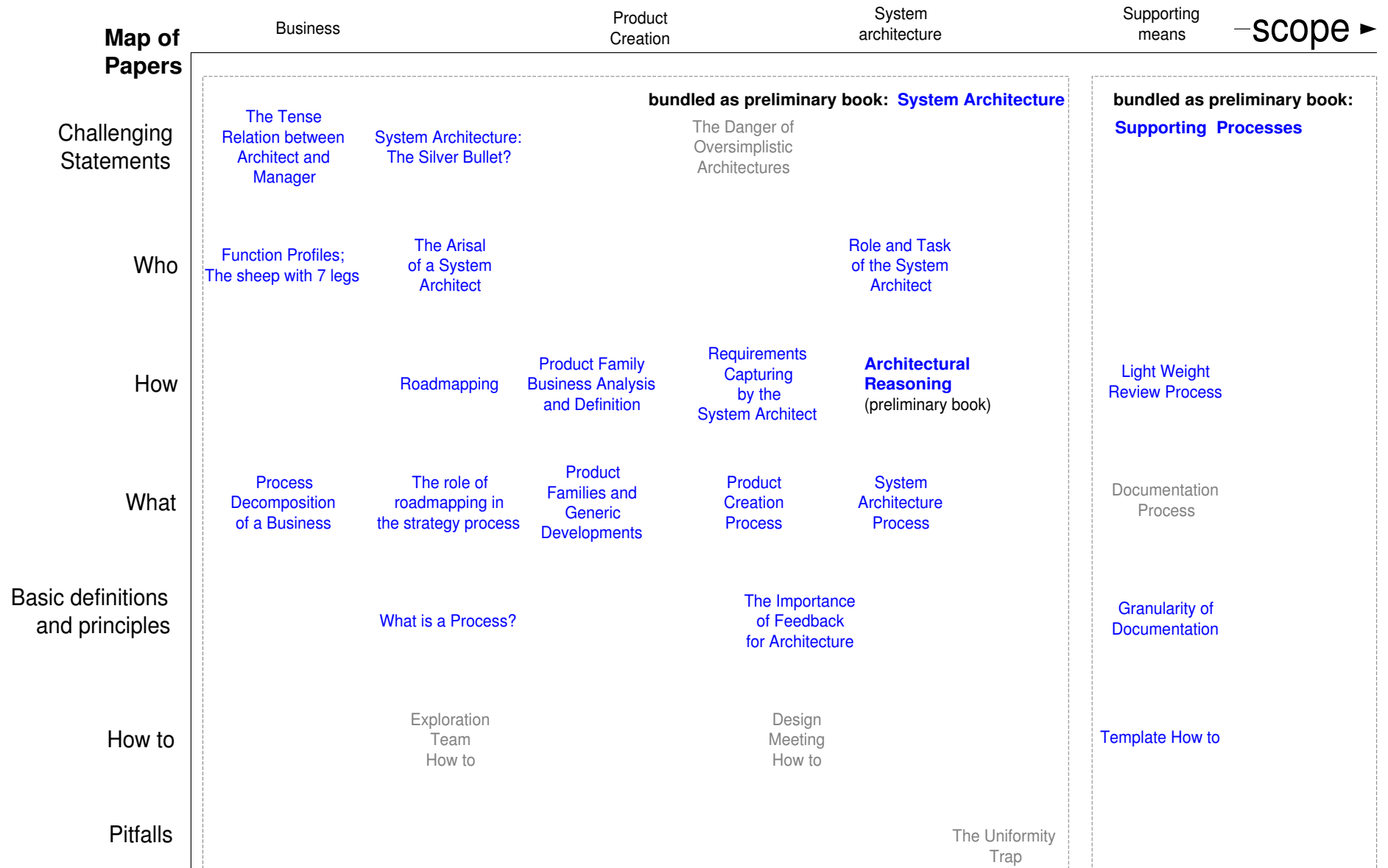


Outlook 2010-2012

	2010	2011	2012
Education	SE master program Reflective Practice yr 2 Master Project Modeling and Analysis Bachelor level	Reflective Practice yr 3 System Design Systems Engineering for other masters	SE PhD program
Research	Master Projects Methodology Research Agenda	staffing research model	PhD Projects Methodology broadening
SE Networks	local: SESG, BUC alumni, strategy&roadmapping, KSEE ESI Sr architecten global: architectingforum.org, SoSE network INCOSE academic forum, symposium, CSER, SEANET		
Tools, website	navigation and search	<i>ideas are welcome!</i>	
Book publication	Systems Architecting in Context	Multi-view Architecting and Modeling	

- frequent releases
- early accessibility (in infancy stage)
- encouragement of further distribution
- aimed at maximum feedback

Positioning Gaudí Documents



Courses based on Gaudí Material

Course	Abbreviation	Duration (in days)	Participants per course	Target audience
System Architecture	SARCH	5	16	architects stakeholders of architects
Management SARCH	MSARCH	2	16	management teams
Embedded Systems Architecting; Stakeholders	ESA	3	16	potential architects
Requirements Engineering as part of OOTI curriculum	OOTI	5	12-18	post-doctoral students
Embedded Systems context	EScontext	4	30	masters students
Execution Architecture (with Ton Kosteljik)	EXARCH ASP	4..5	16	SW designers architects
Multi-Objective System Architecting and Design	MOSAD	3..5	16	designers architects
System Modeling and Analysis	MA611	3..5	16	designers architects

Status of Courses

Course	Abbreviation	number of courses upto March2008	appr. total participants
System Architecture	SARCH	44	660
Management SARCH	MSARCH	7	72
Embedded Systems Architecting; Stakeholders	ESA	20	300
Requirements Engineering as part of OOTI curriculum	OOTI	7	125
Embedded Systems context	EScontext	3	90
Execution Architecture (with Ton Kostelijck)	EXARCH ASP	11	160
Multi-Objective System Architecting and Design	MOSAD	3	36
System Modeling and Analysis	MA611	2	16

Course Modules

No.	Content	MSARCH	SARCH	ESA
0	Course information (course-specific), Gaudí project	+	+	+
1	Positioning the system architecture process, product creation process	+	+	+
2	Role and task of the system architect	+	+	+
3	Requirements capturing	+	+	+
4	System architect toolkit		+	
5	Roadmapping	+	+	+
6	Product families, generic developments	+	+	+
7	Documentation, reviewing and other supporting processes		+	+
8	The role of software in complex products		+	
9	Psycho-social side		+	some
10	Wrap up, expectations, how to continue, evaluation	+	+	
11	Human Resource Management wrt architects	+		

Module System Architecture Context

by *Gerrit Muller* Buskerud University College and Embedded Systems
Institute

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

`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.

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: 1



Embedded Systems
INSTITUTE

Process Decomposition of a Business

by *Gerrit Muller* Embedded Systems Institute

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

`www.gaudisite.nl`

Abstract

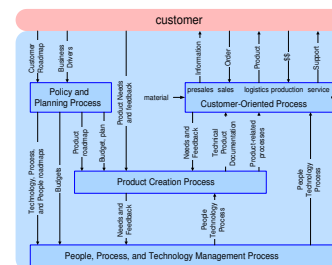
This article positions the system architecture process in a wider business scope. This positioning is intended to help understanding the processes in which the system architect (or team of system architects) is involved.

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.

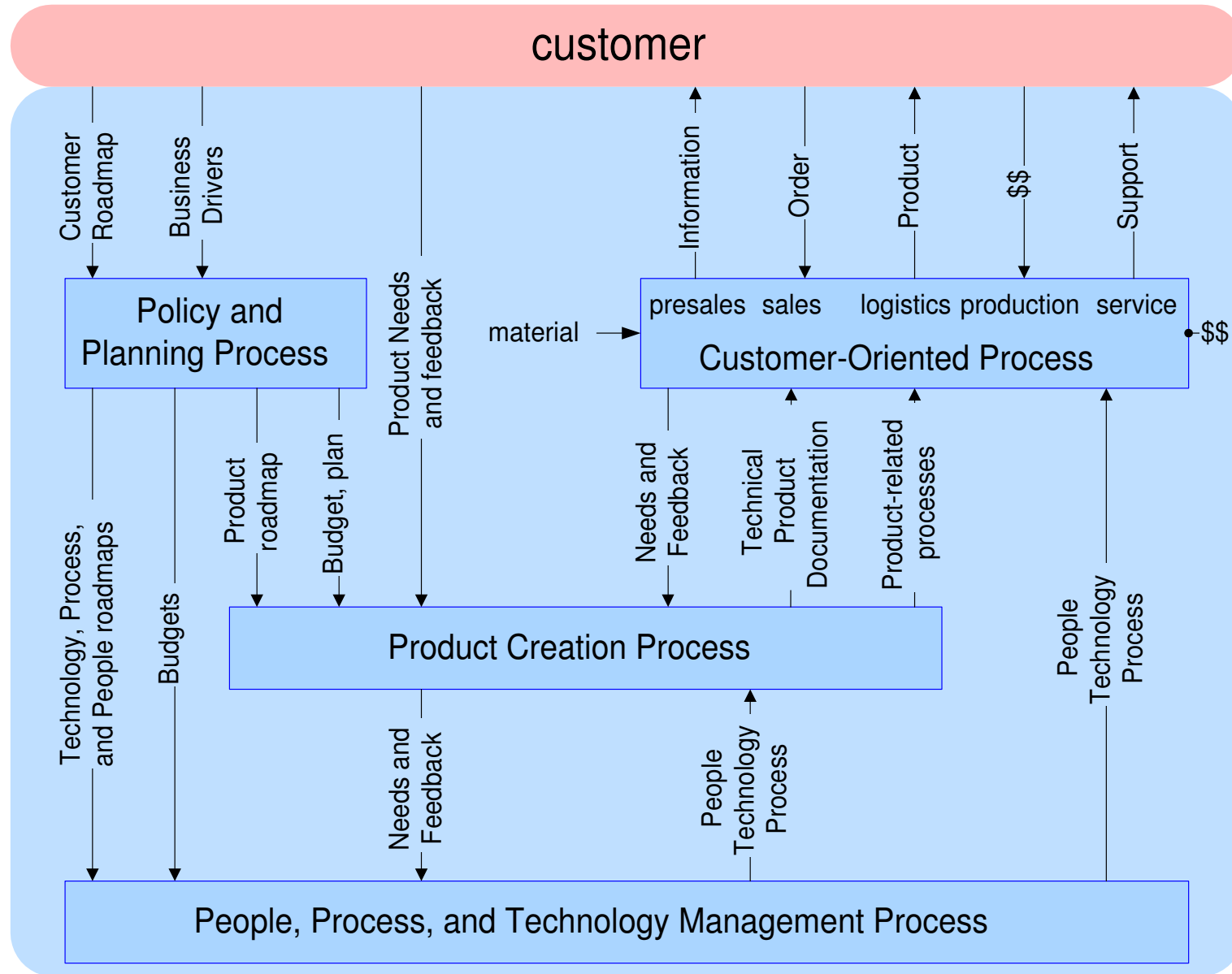
Distribution

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.

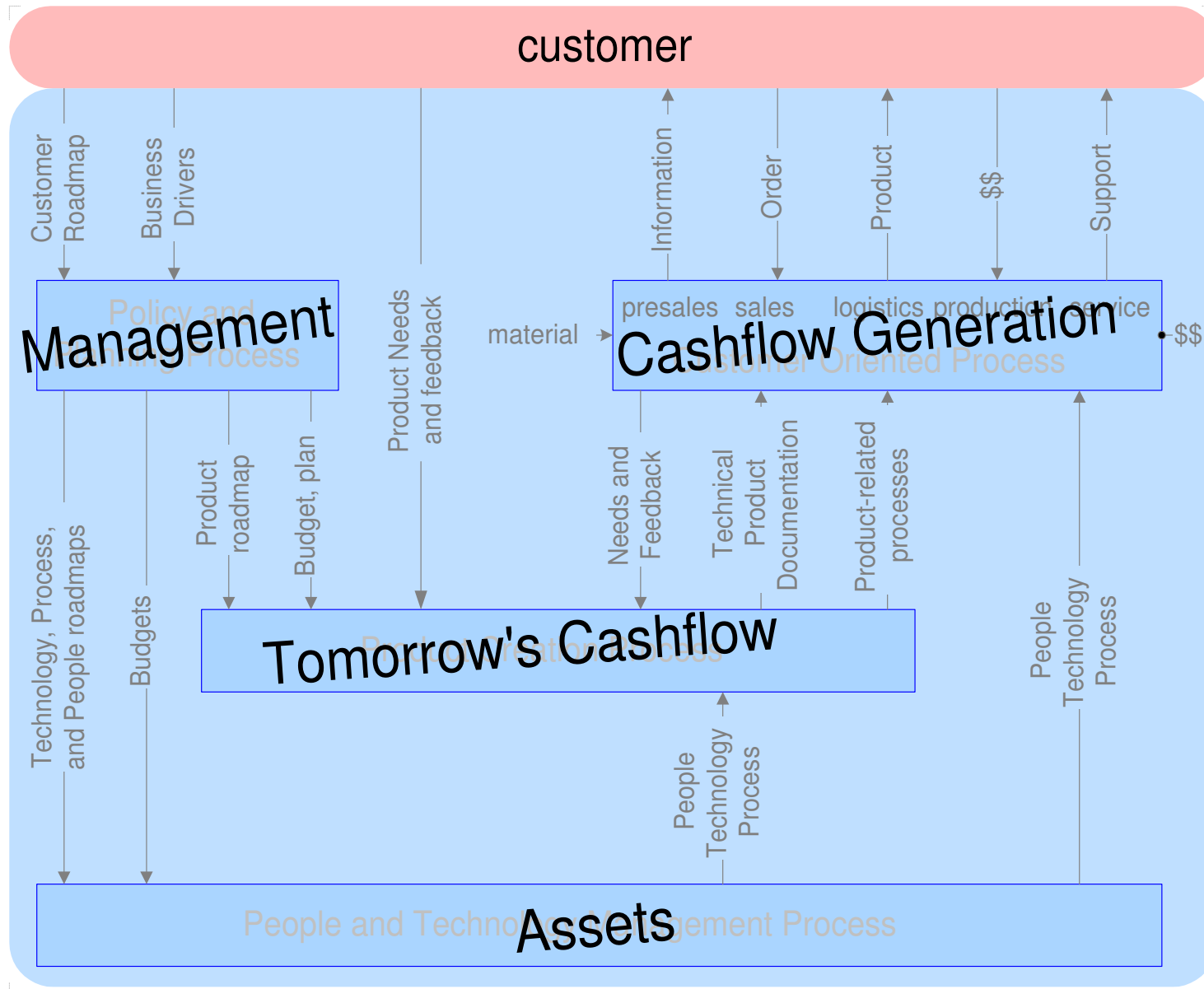
February 11, 2012
status: concept
version: 1.1



Simplified Decomposition of the Business



Financial Characterization of Decomposition



Multiple Instances per Process

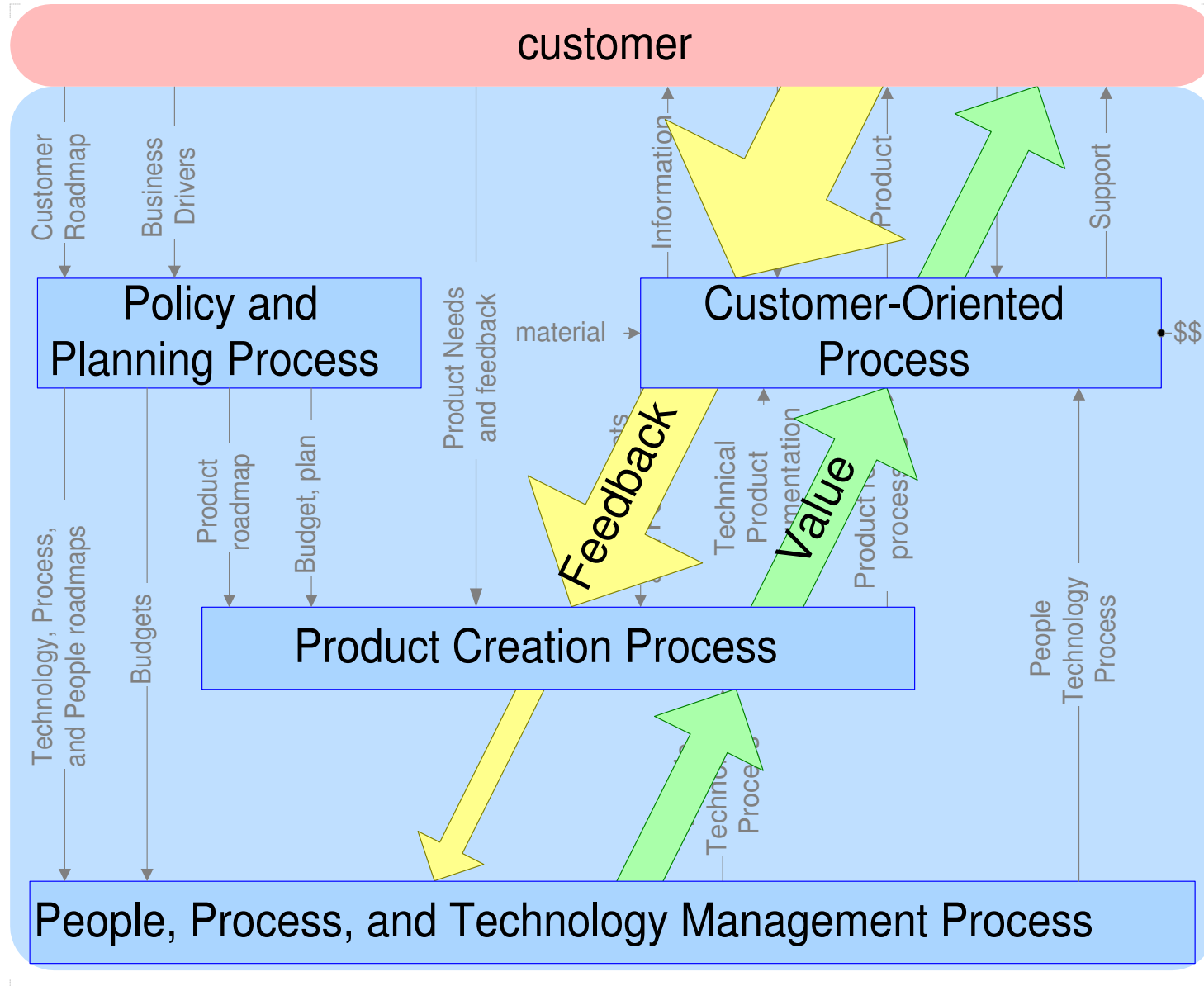
Customer Oriented Process: Depends on geography, customer base, and supply chain.

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

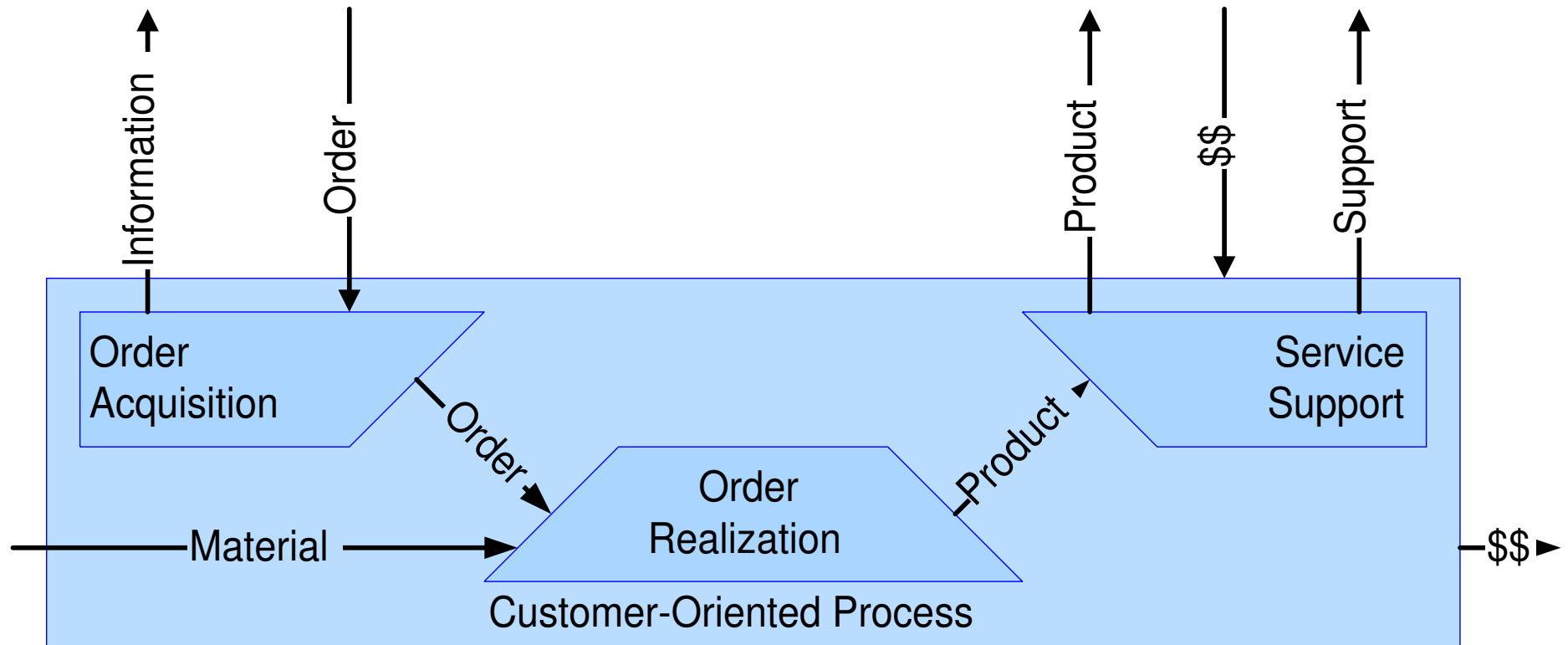
People and Technology Management Process: One per “competence”, where a competence is a cohesive set of technologies and methods.

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

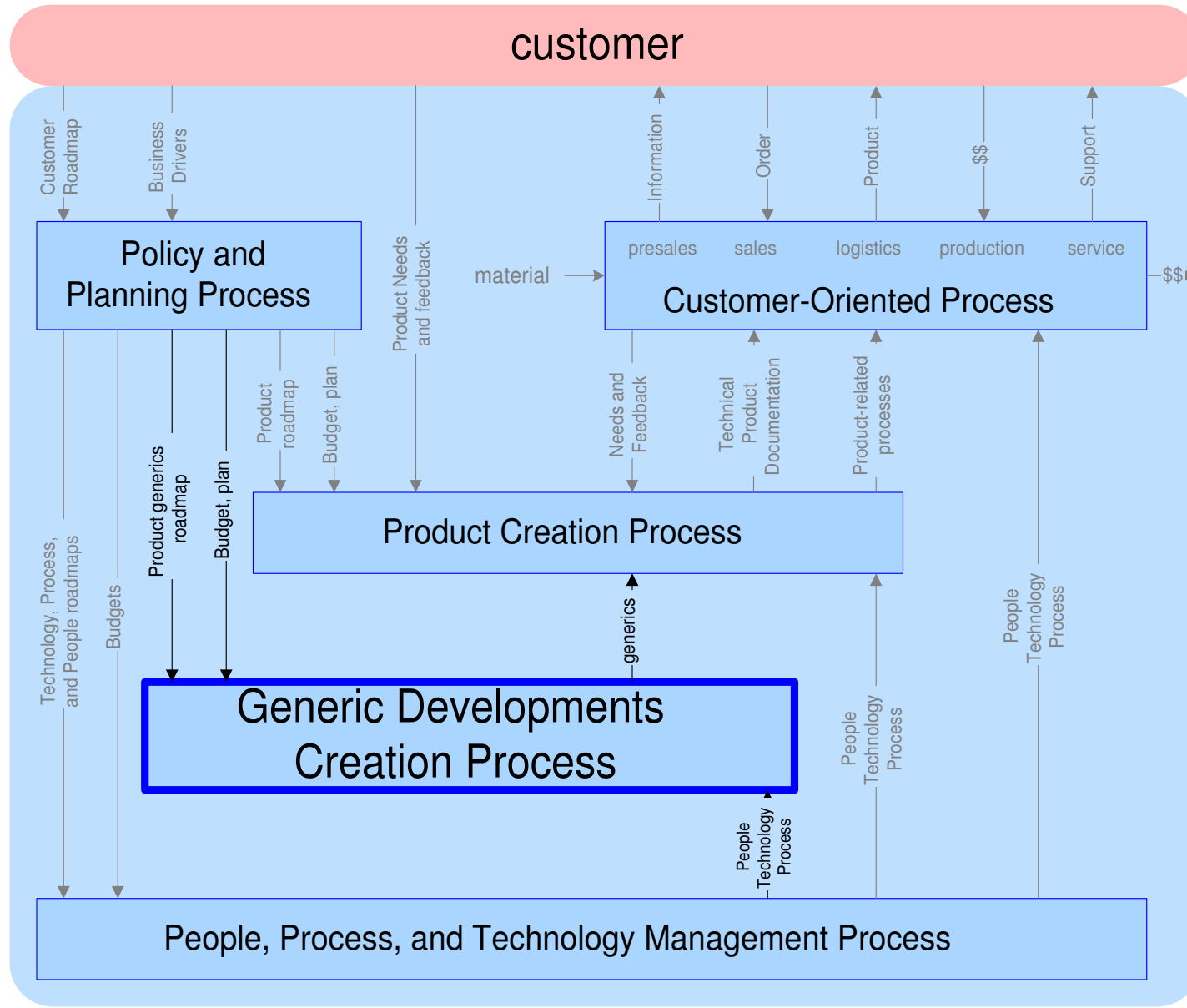
The Value Chain and the Opposite Feedback Flow



Decomposition of the Customer Oriented Process



Extended with Generic Developments



The Product Creation Process

by *Gerrit Muller* Buskerud University College

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

`www.gaudisite.nl`

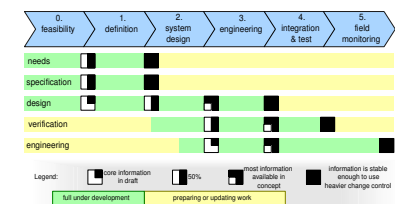
Abstract

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.

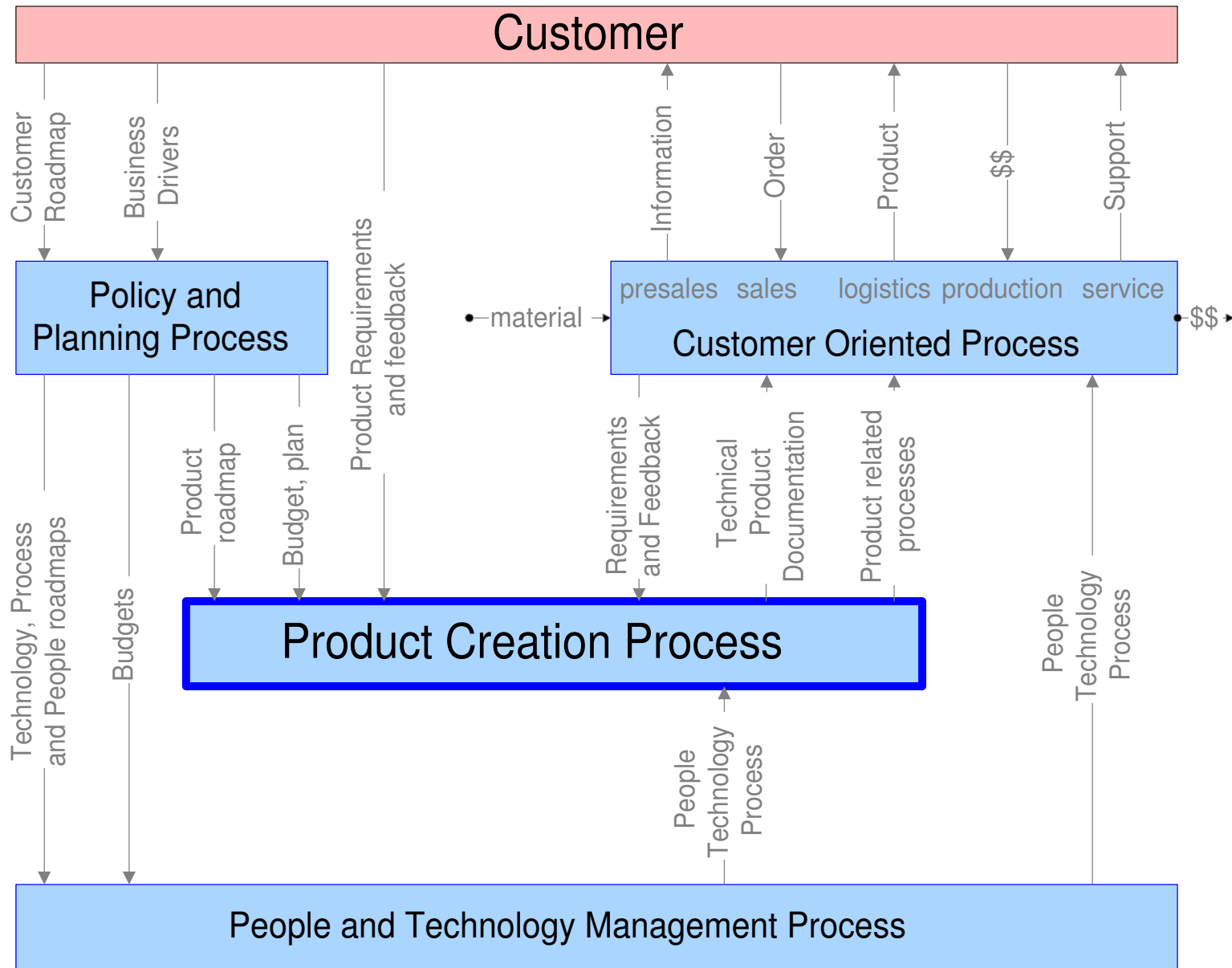
Distribution

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.

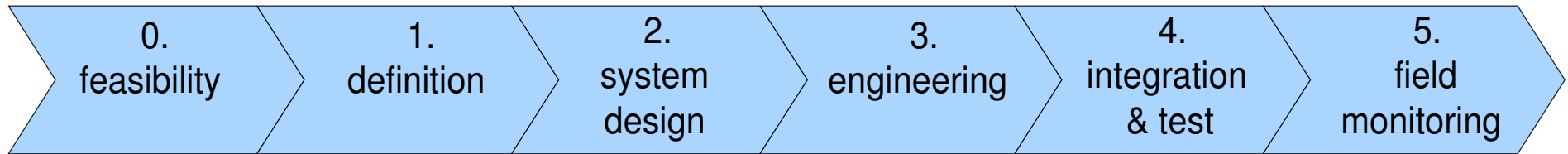
February 11, 2012
status: concept
version: 2.2



The Product Creation Process in Business Context



Phasing of the PCP at Business Level



sales

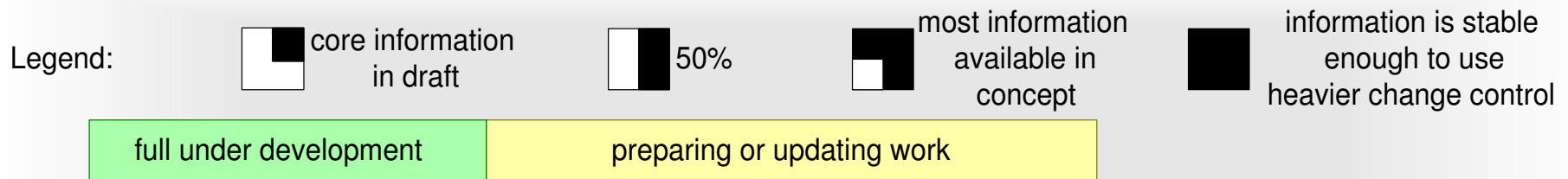
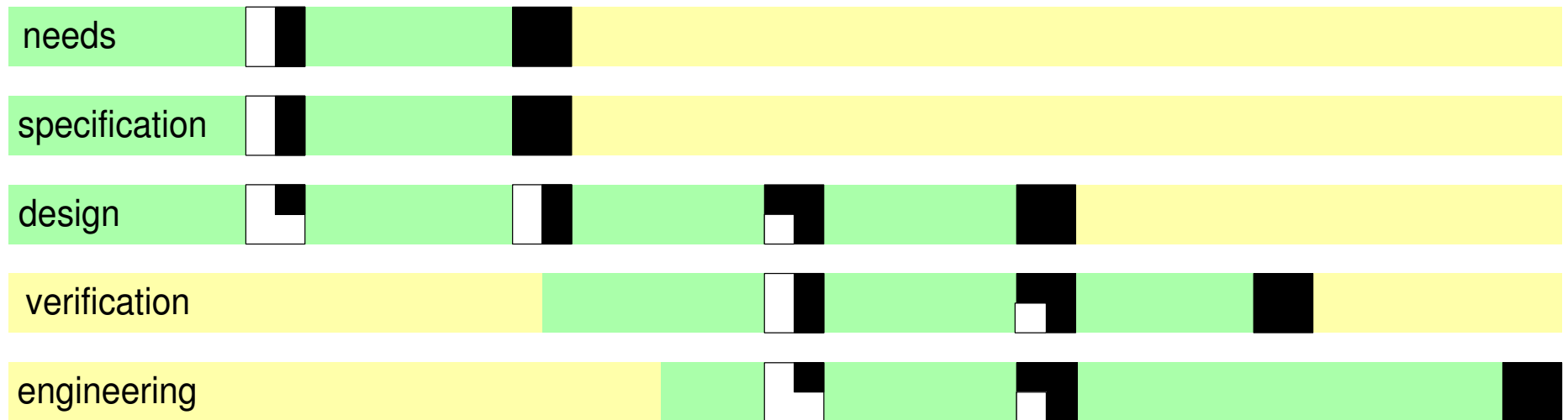
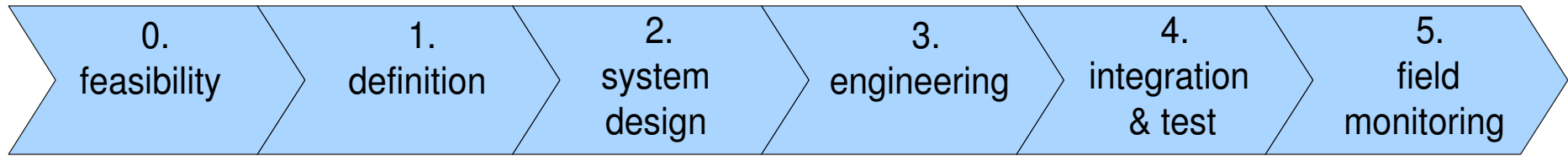
logistics

production

service

development & engineering: marketing, project management, design

Phasing the Design Control Process



Advantages and Disadvantages of a Phased Process

benefits

blueprint: how to work

reuse of experience

employees know *what* and *when*

reference for management

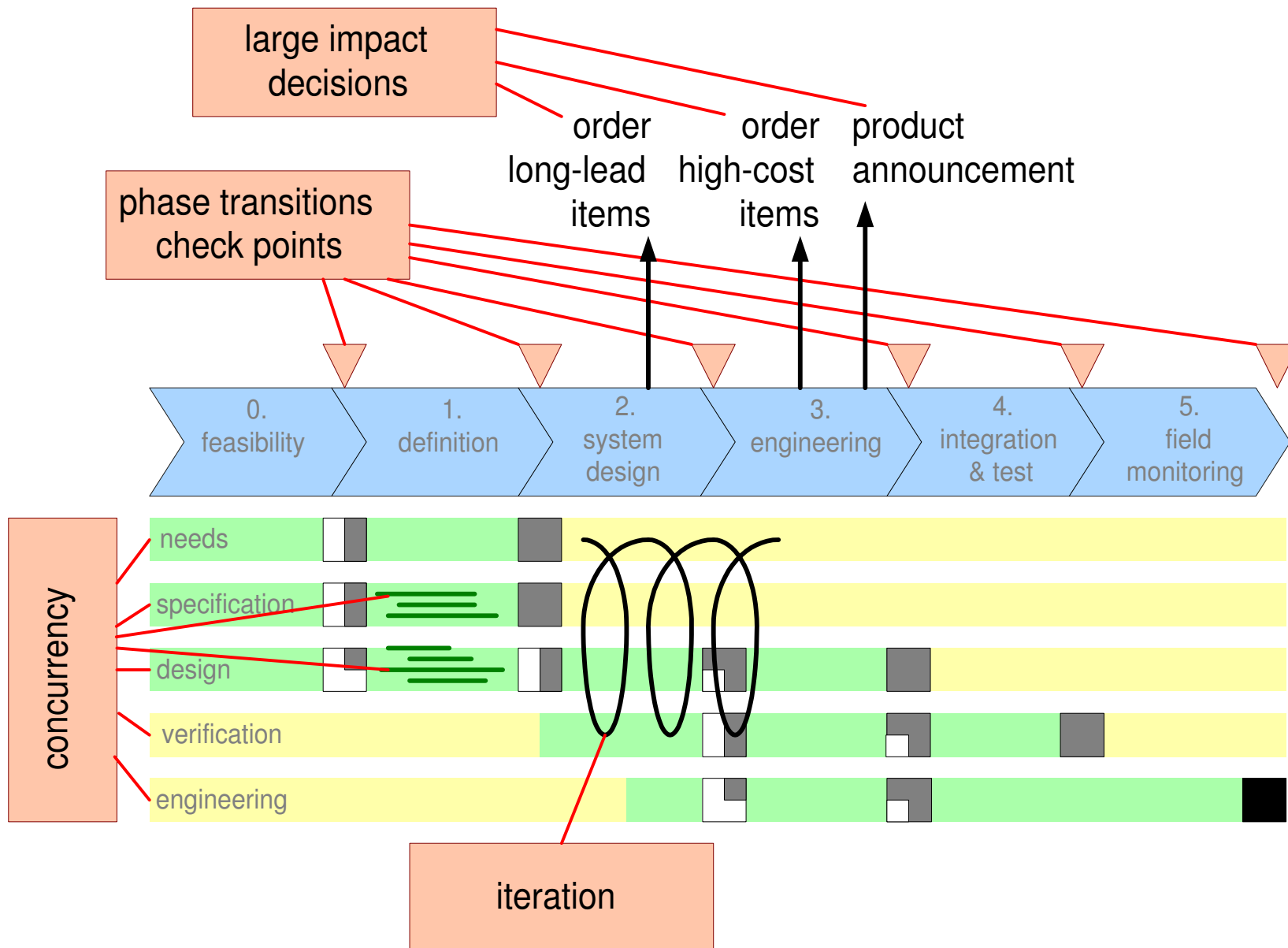
disadvantages

following blueprint blindly

too bureaucratic

transitions treated black and white

Characteristics of a Phase Model



Decisions and Phase Transitions

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

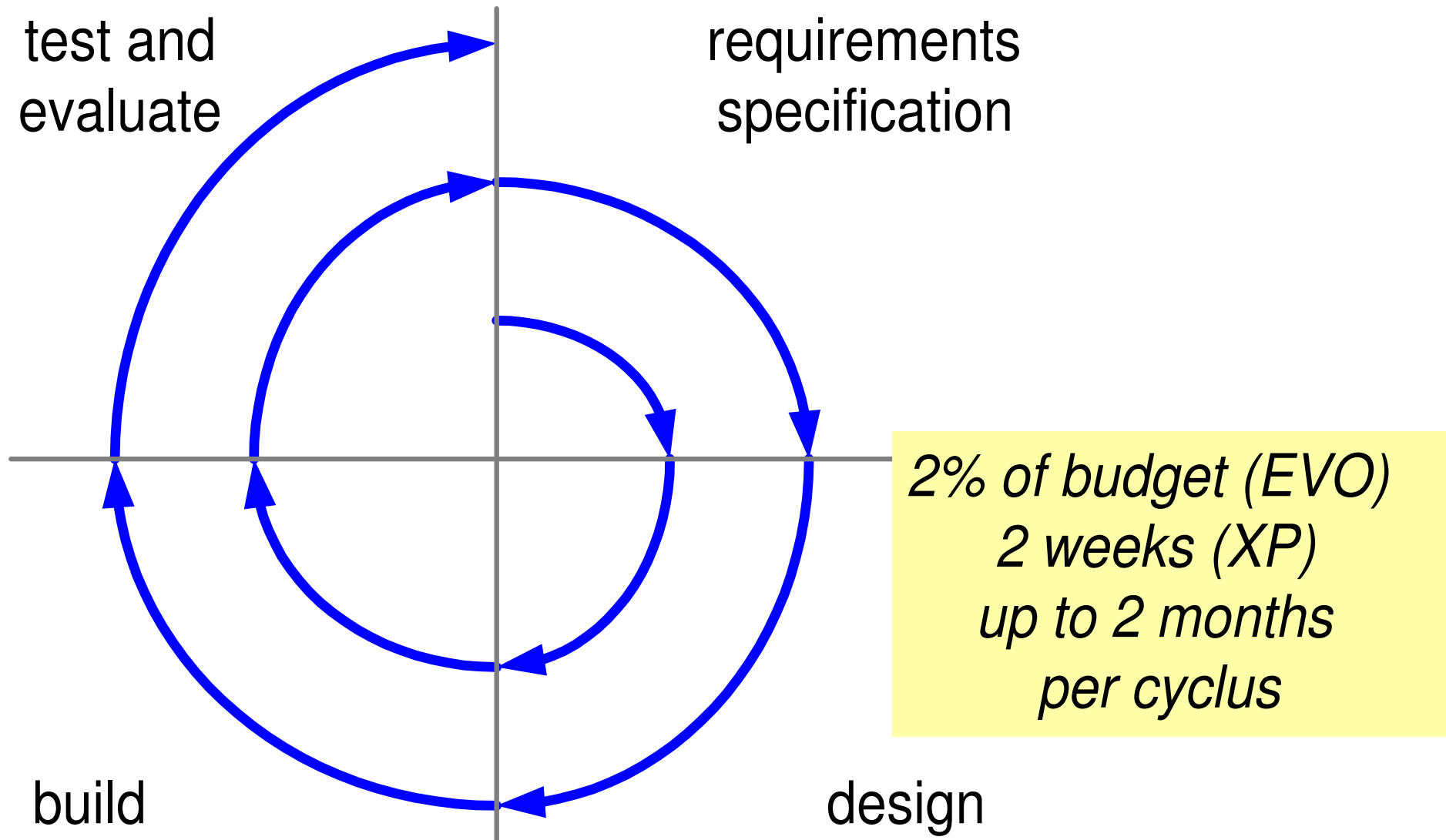
Define the mandatory and supporting information required for the decision.

Schedule a decision after the appropriate phase transition.

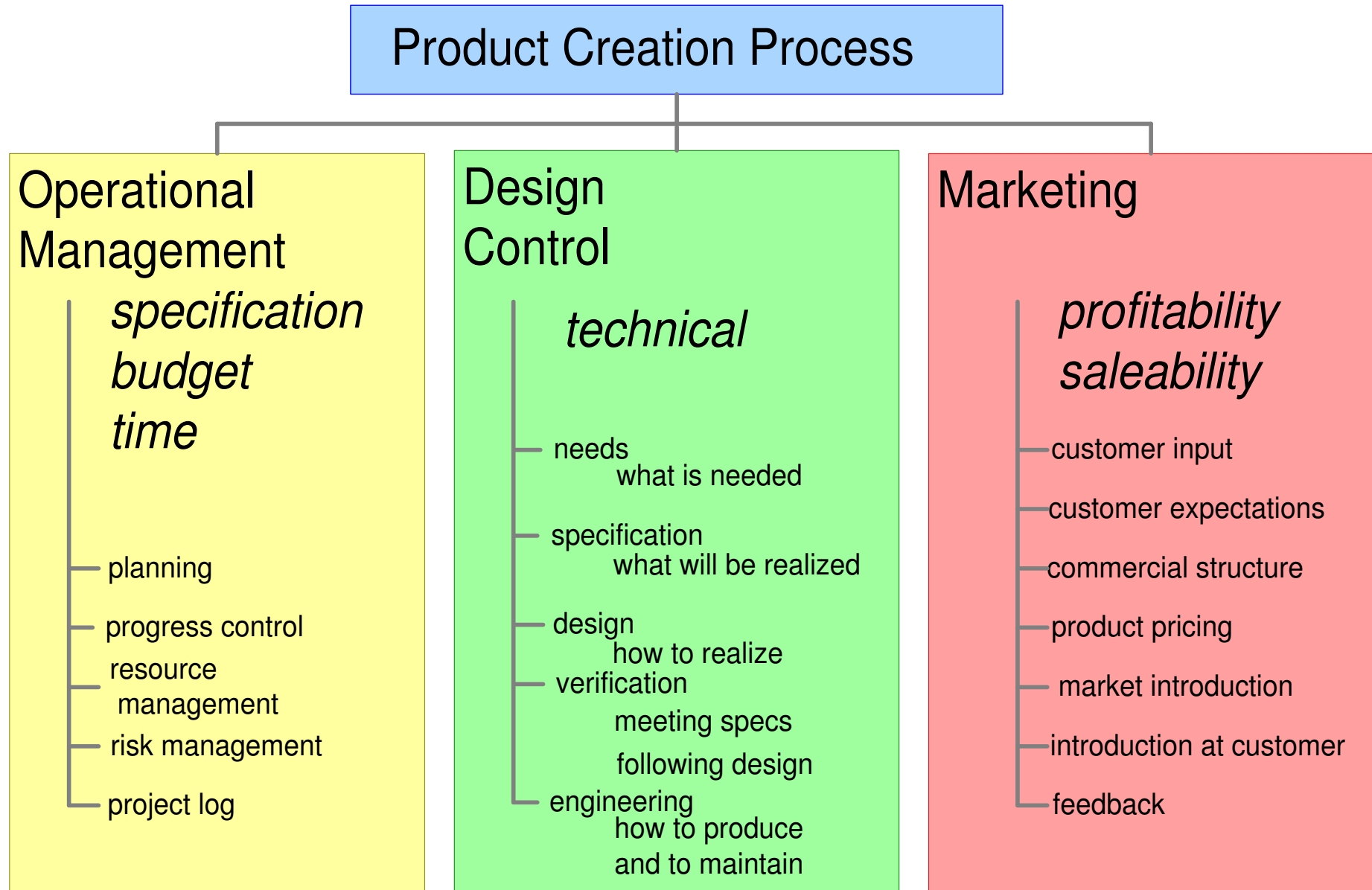
Decide explicitly.

Communicate the decision clearly and widely.

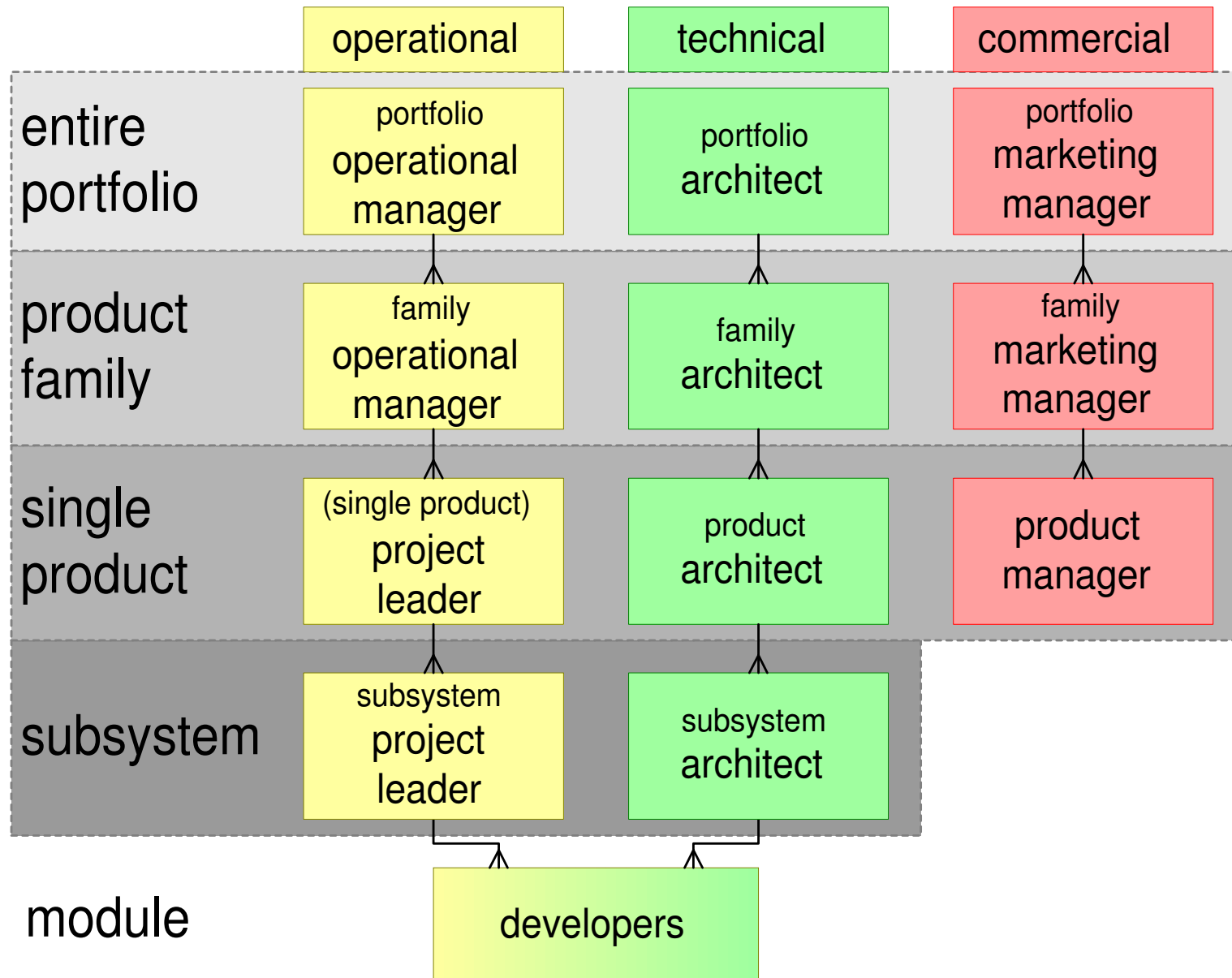
Evolutionary PCP model



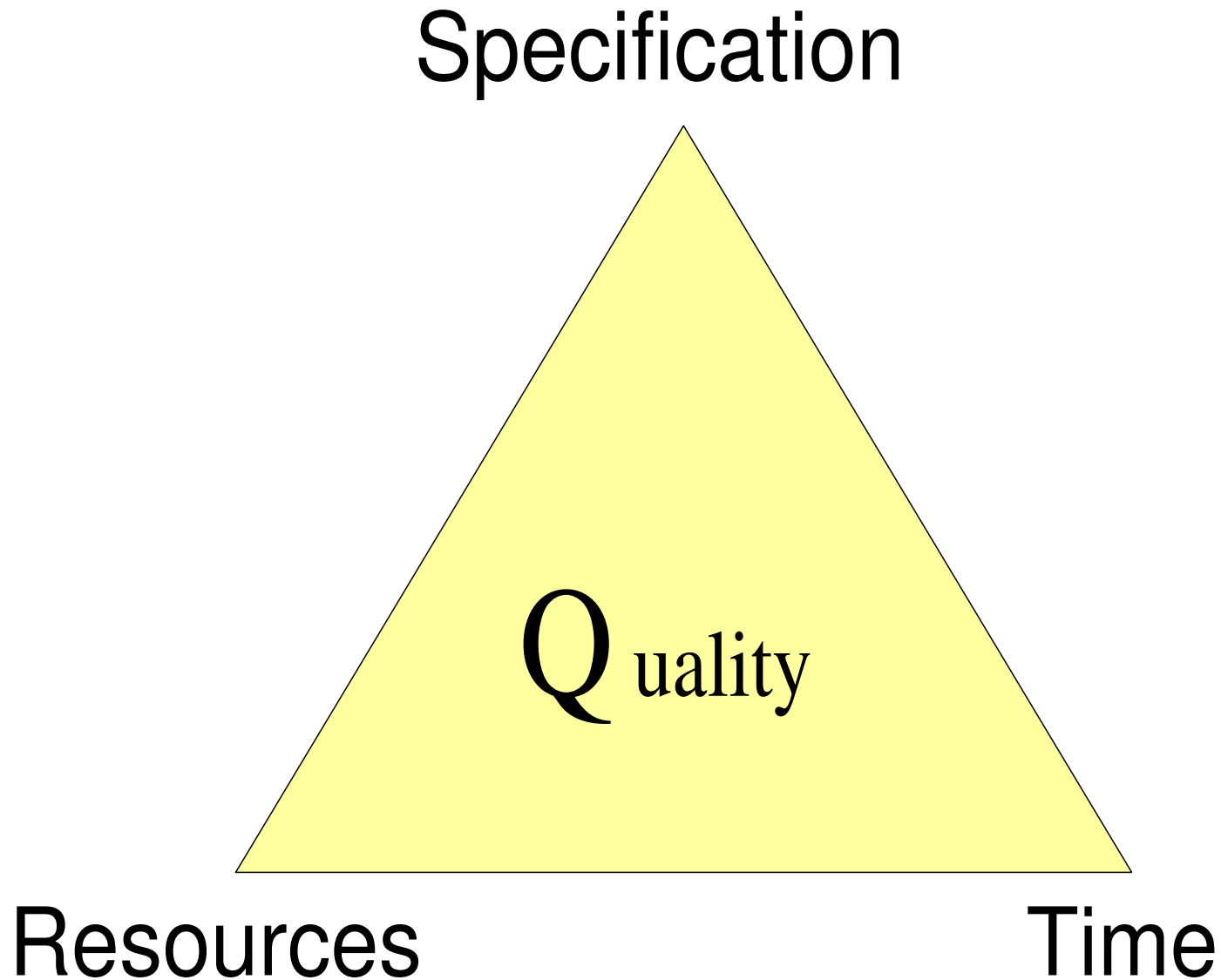
Decomposition of the Product Creation Process



Operational Organization of the PCP

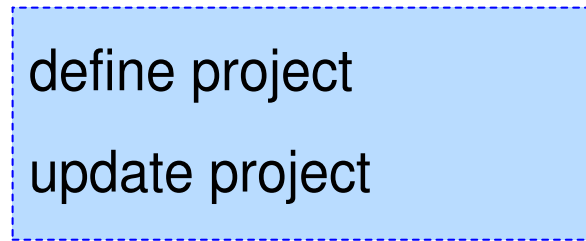


Prime Responsibilities of the Operational Leader

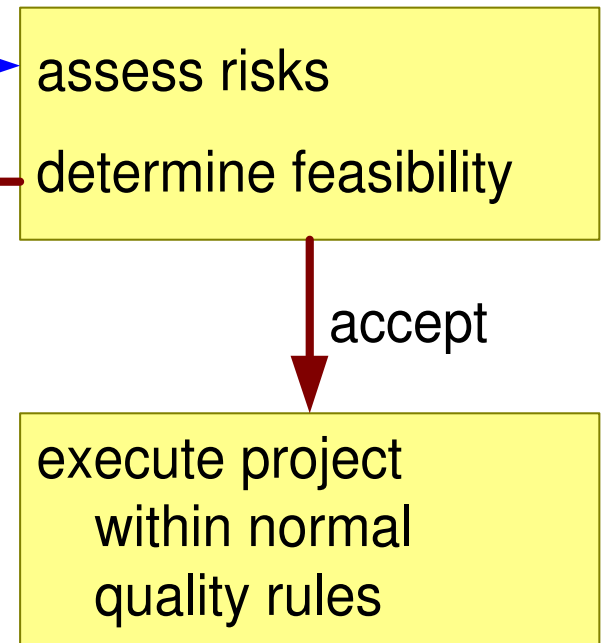


The Rules of the Operational Game

business management



project leader

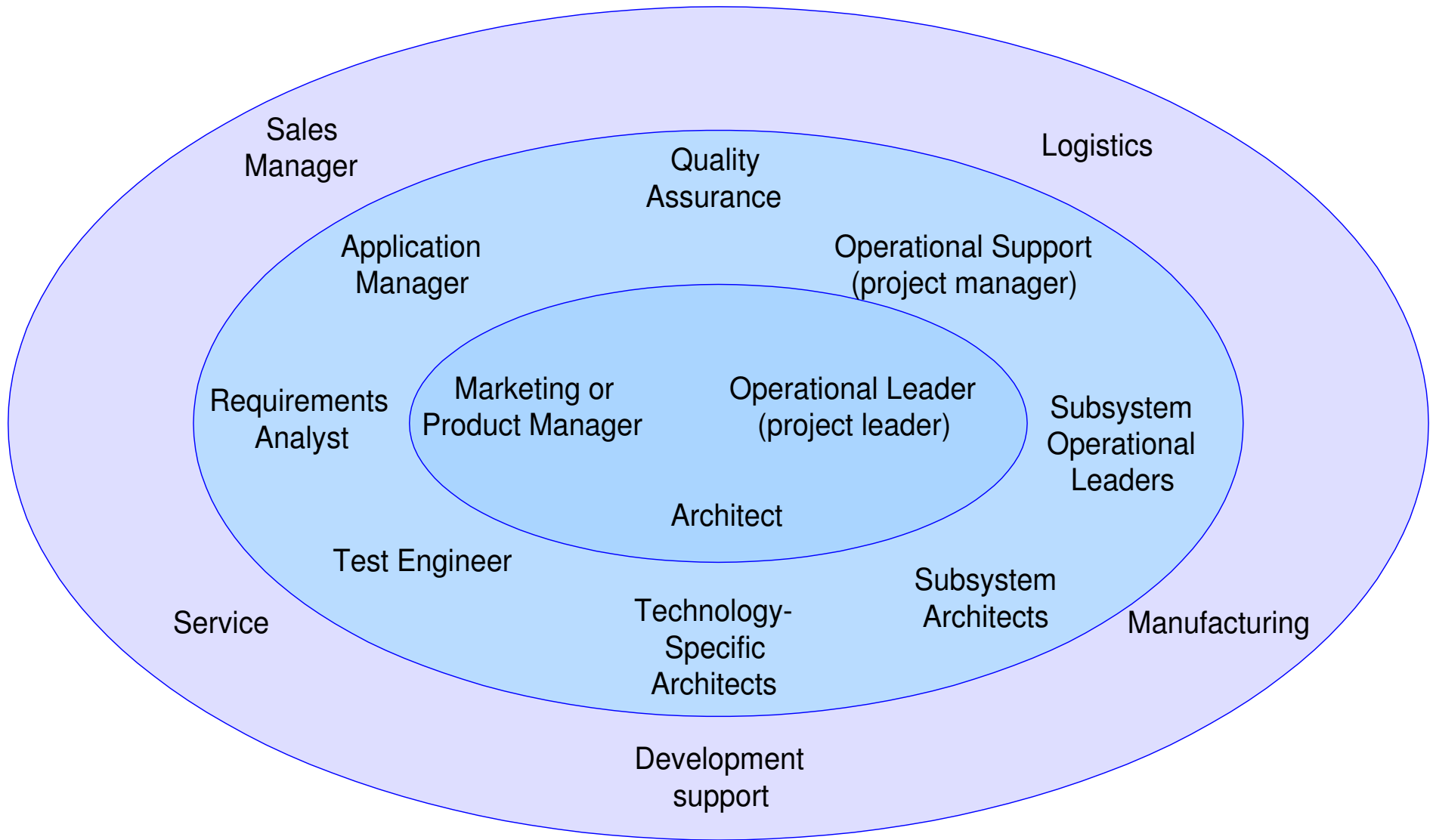


specification, resources, time

accept or reject

accept

Operational Teams



The System Architecture Process

by *Gerrit Muller* Buskerud University College

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

`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.

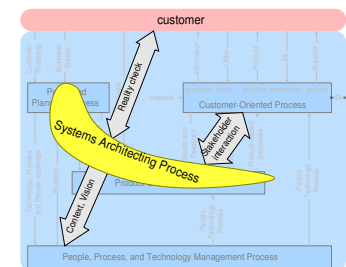
The purpose of the System Architecture Process is to provide the Integral Technical overview and consistency, and to maintain the integrity over time. Subjective characteristics as elegance and simplicity are key elements of a good architecture.

The scope of the system architecture process is illustrated by showing 5 views used in a reference architecture, ranging from Customer Business to Realization.

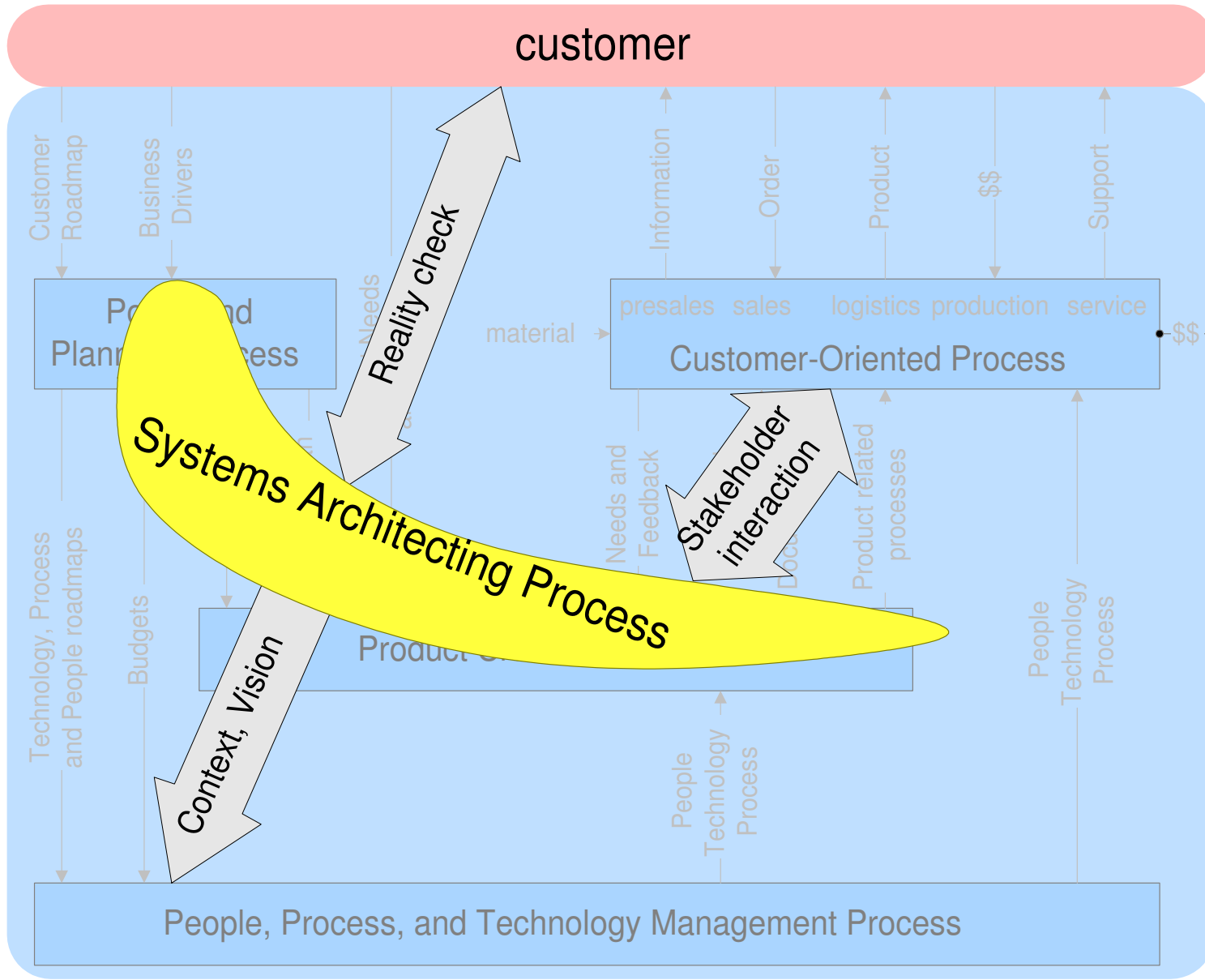
Distribution

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.

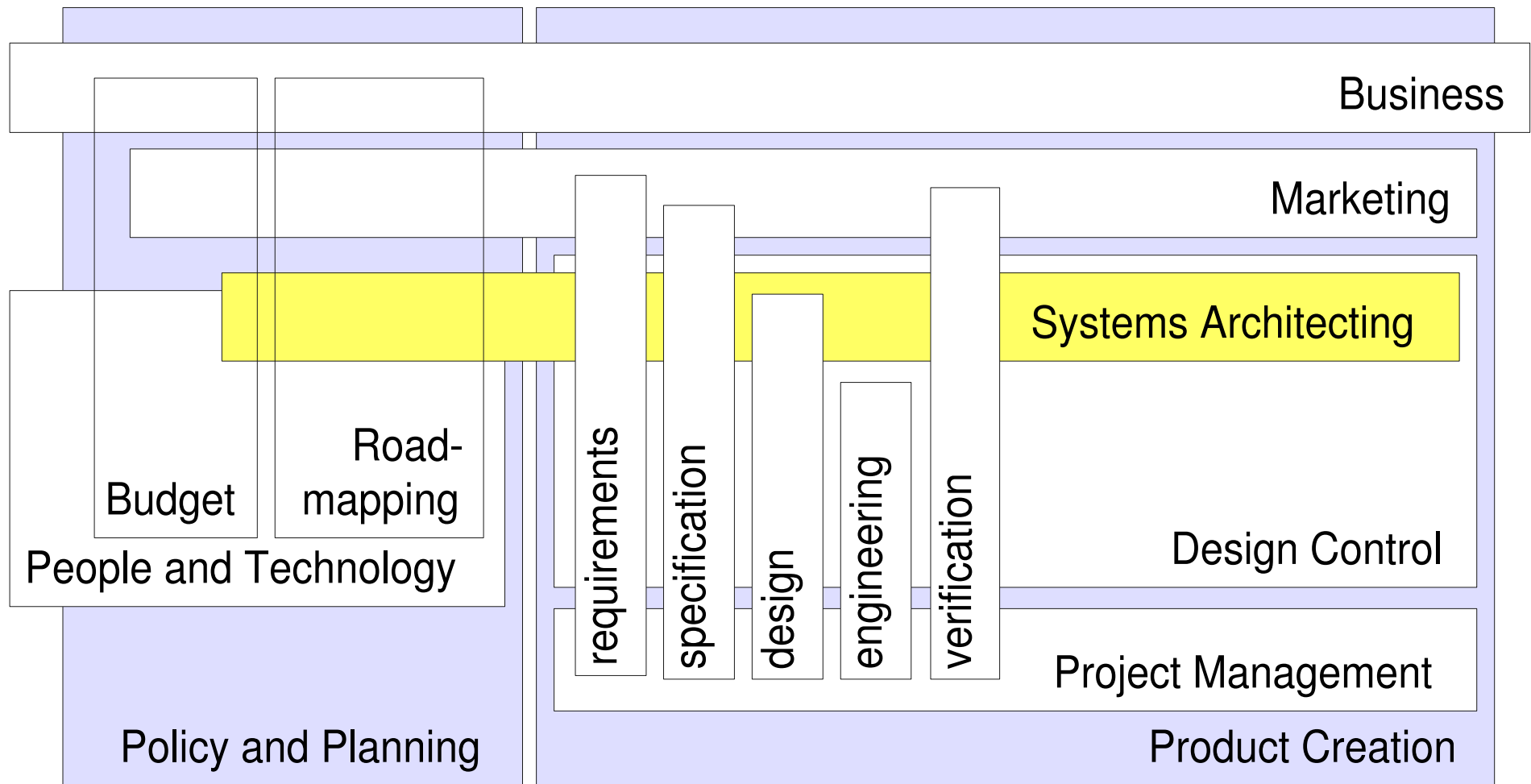
February 11, 2012
status: concept
version: 2.3



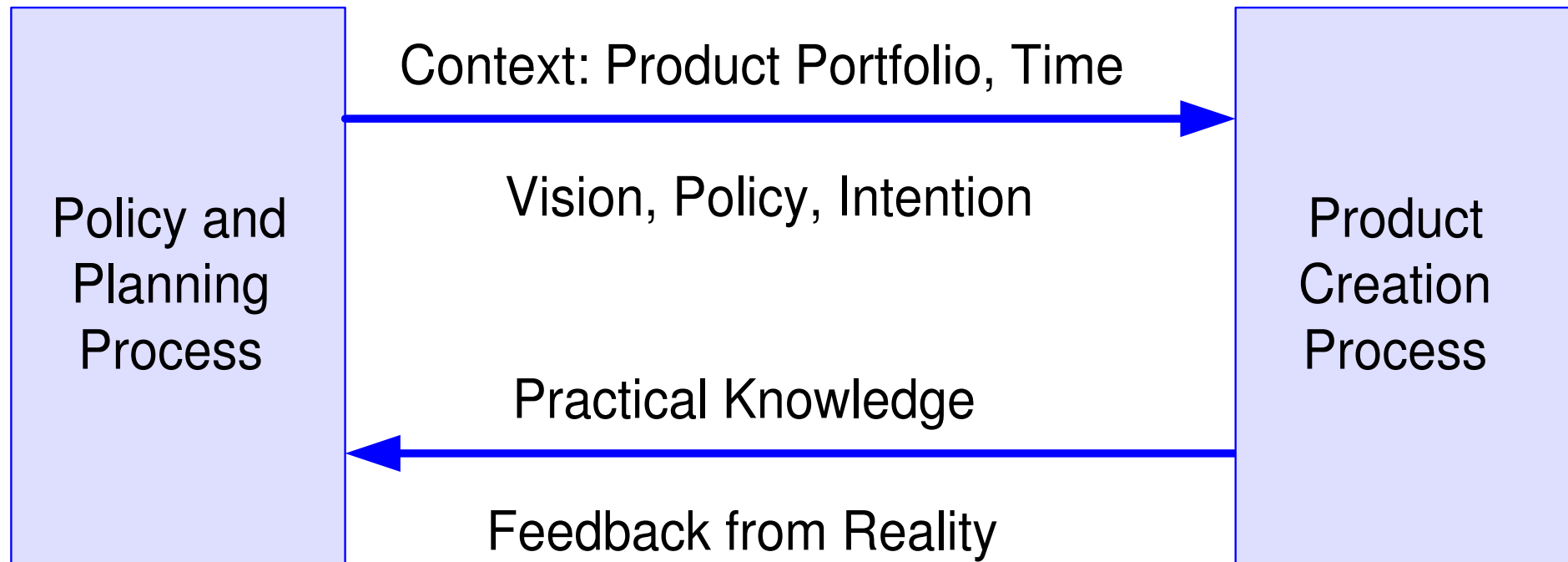
System Architecting Process in Business Context



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

performance



qualities



functionality



synergy



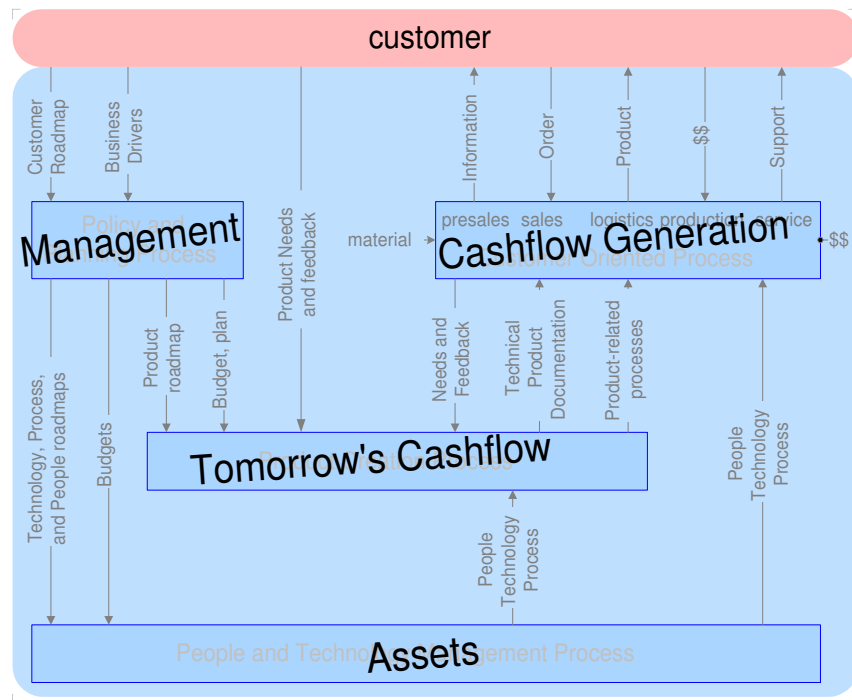
specific solution

Exercise Product Creation Process

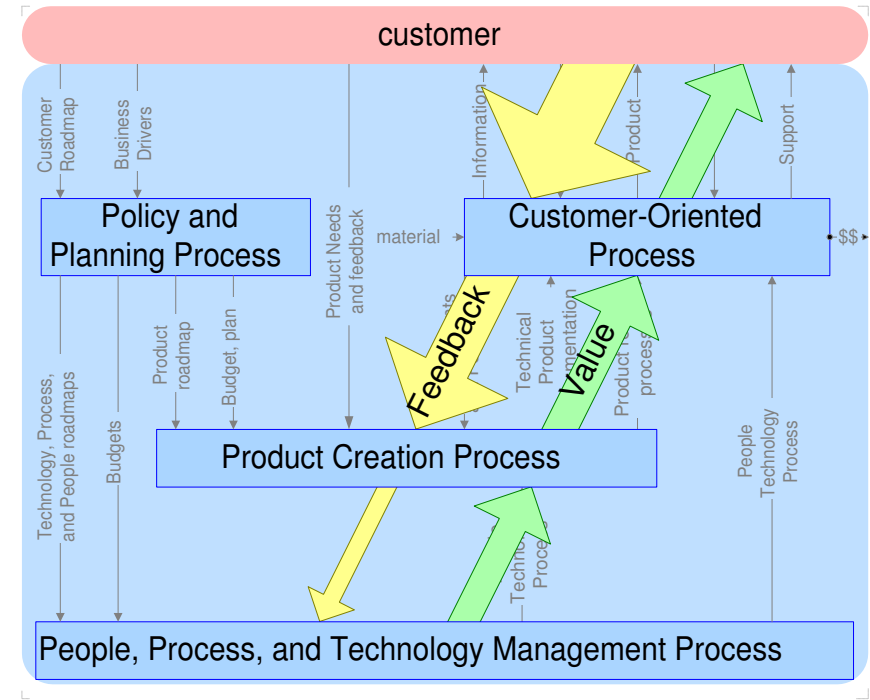
1. Map operational organization.
2. Report on one flip the best case.
3. Identify the relationships of the core team: geographical, organizational, psychological, et cetera.
4. Report the result of 3 on one flip.

Process Decomposition of a Business

Importance in Financial terms

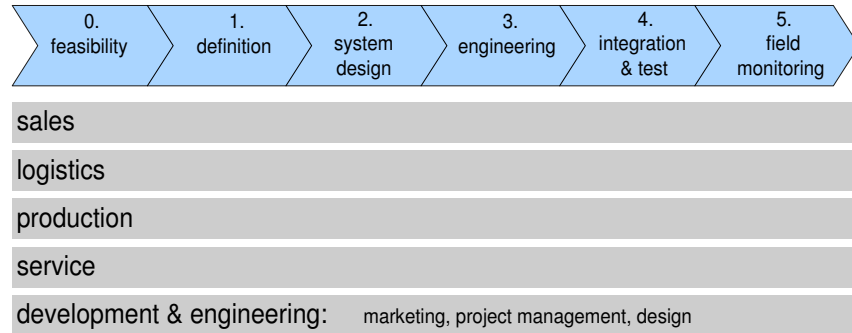


Value Chain and Feedback Flow

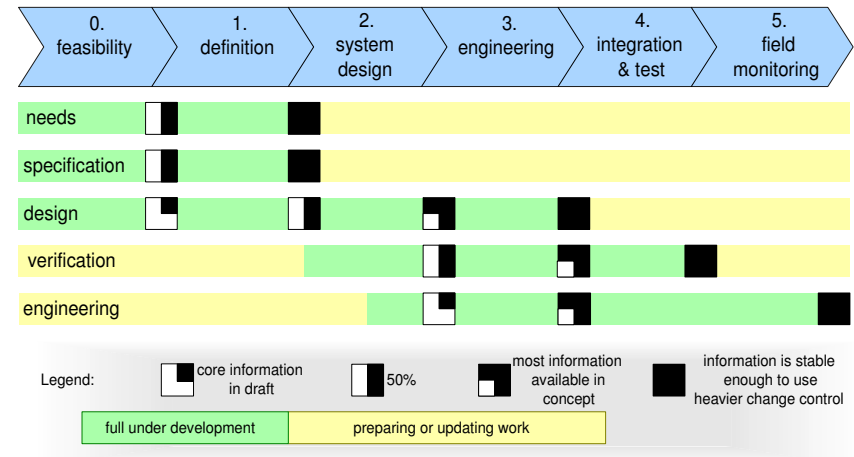


Product Creation Process

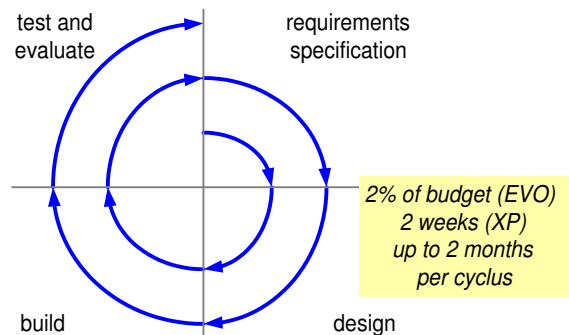
PCP involves **all** disciplines, much more than D&E



Phased Process

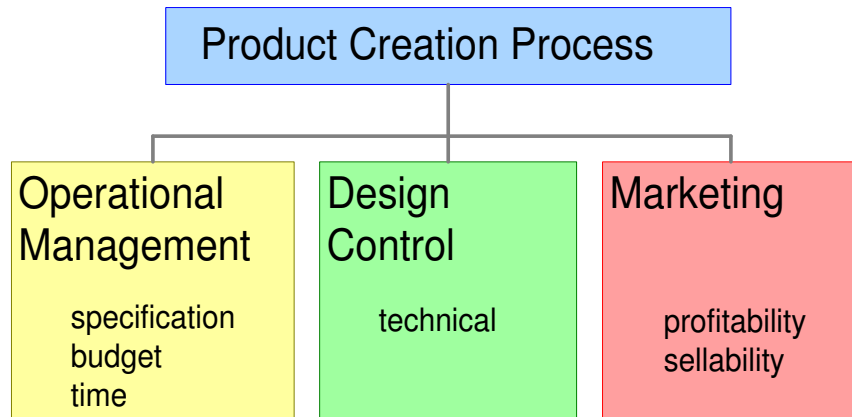


Incremental Development

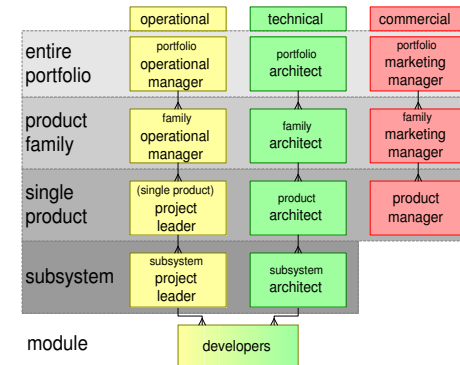


PCP Decomposition and Operational Management

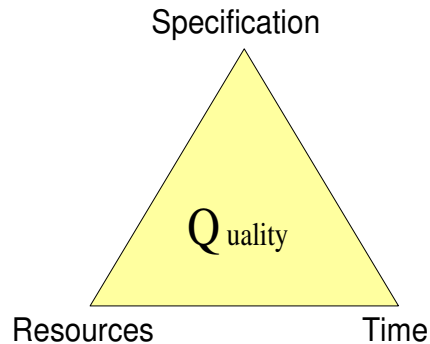
PCP decomposition



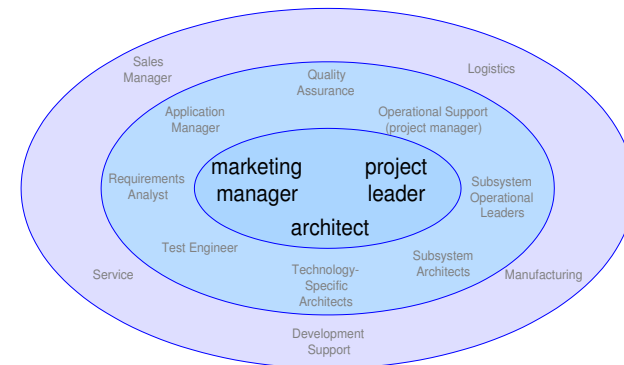
Architecture at all levels; From portfolio to subsystem



Operational Commitment

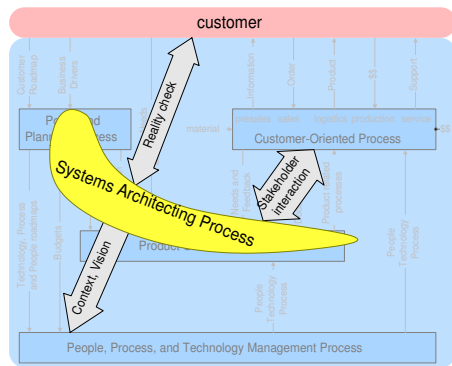


Core: Operational + Technical + Commercial



System Architecture Process

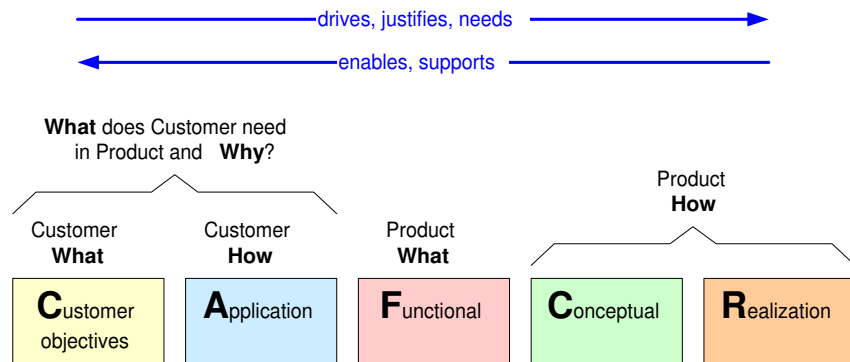
In Business Context



Key Issues

<p><i>key words</i></p> <p>balance</p> <p>consistency</p> <p>integrity</p> <p>simplicity</p> <p>elegance</p> <p>stakeholder satisfaction</p>	<p><i>balancing acts</i></p> <p>External ↔ internal requirements</p> <p>Short term needs ↔ long term interests</p> <p>Efforts ↔ risks from requirements to verification</p> <p>Mutual influence of detailed designs</p> <p>Value ↔ costs</p>
	<p><i>example trade-offs</i></p> <p>performance ↔ synergy</p> <p>qualities ↔ functionality ↔ specific solution</p>

5 Views



Module Product Families and Generic Developments

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

Abstract

This module addresses product families and generic developments.

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: 1.1



Product Families and Generic Aspects

by *Gerrit Muller* Buskerud University College

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

`www.gaudisite.nl`

Abstract

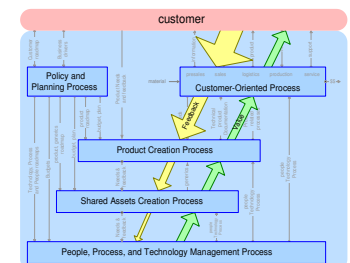
Most products fit in a larger family of products. The members of such a product family share a lot of functionality and features. It is attractive to share implementations, designs et cetera between those members to increase the efficiency of the entire company.

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.

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: 2.3



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	maturing realization
Improved reliability	
Improved predictability	
Easier diversity management	modularity
Increases uniformity	less learning
Employees only have to understand one base system	
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

longer time to market
high investments
lots of maintenance
poor quality
poor reliability
diversity is opposed
lot of know how required
predictable too late
dependability
knowledge dilution
lack of market focus
interference
but integration required

good

reduced time to market
reduced investment
reduced (shared) maintenance cost
improved quality
improved reliability
easier diversity management
understanding of one base system
improved predictability
larger purchasing power
means to consolidate knowledge
increase added value
enables parallel developments
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

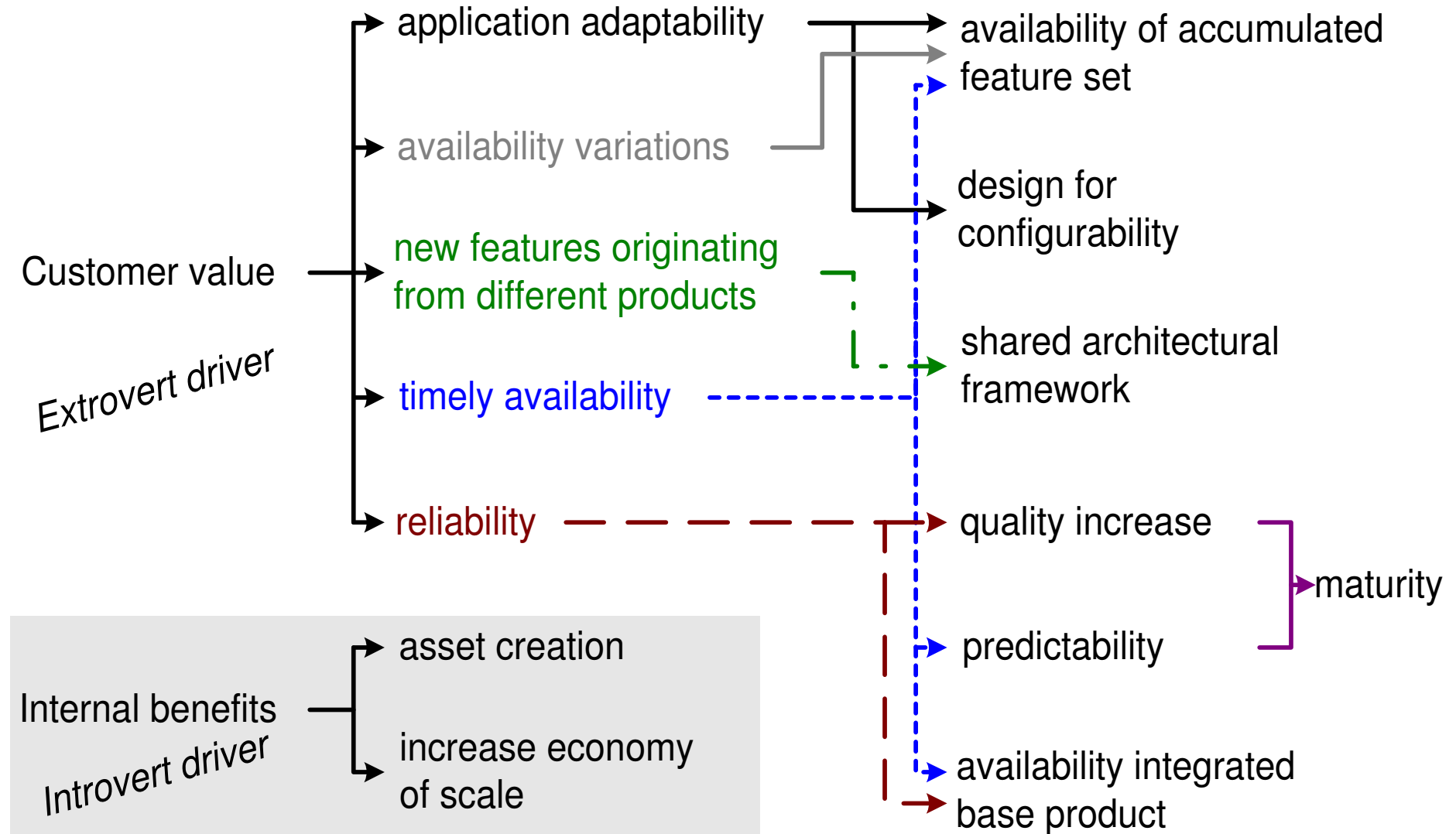
TV: LCD screens
cath lab: image based acquisition control

software maintenance, configurations, integration, release

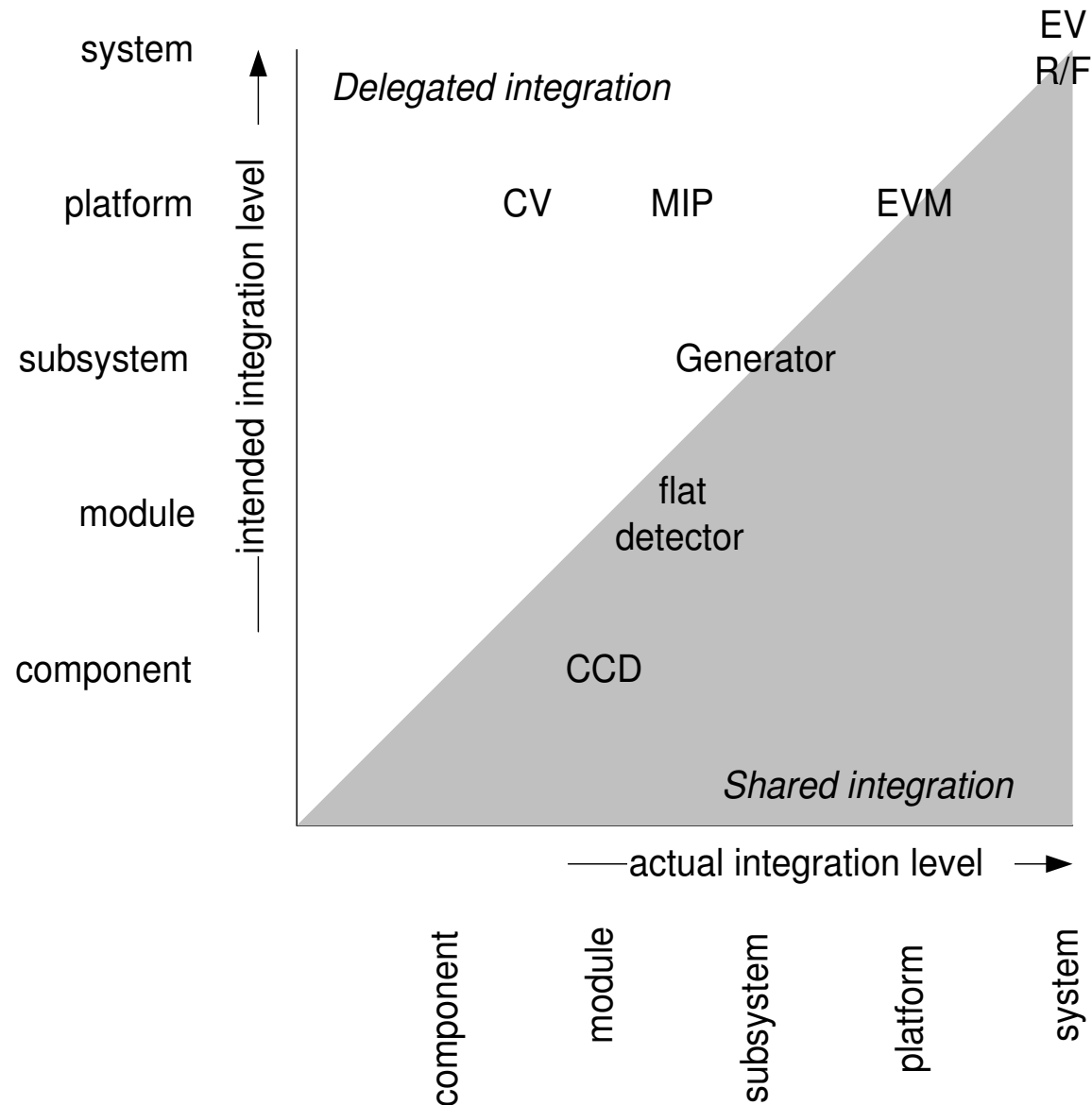
MRI: integration and test
wafersteppers: number of configurations

how to innovate?

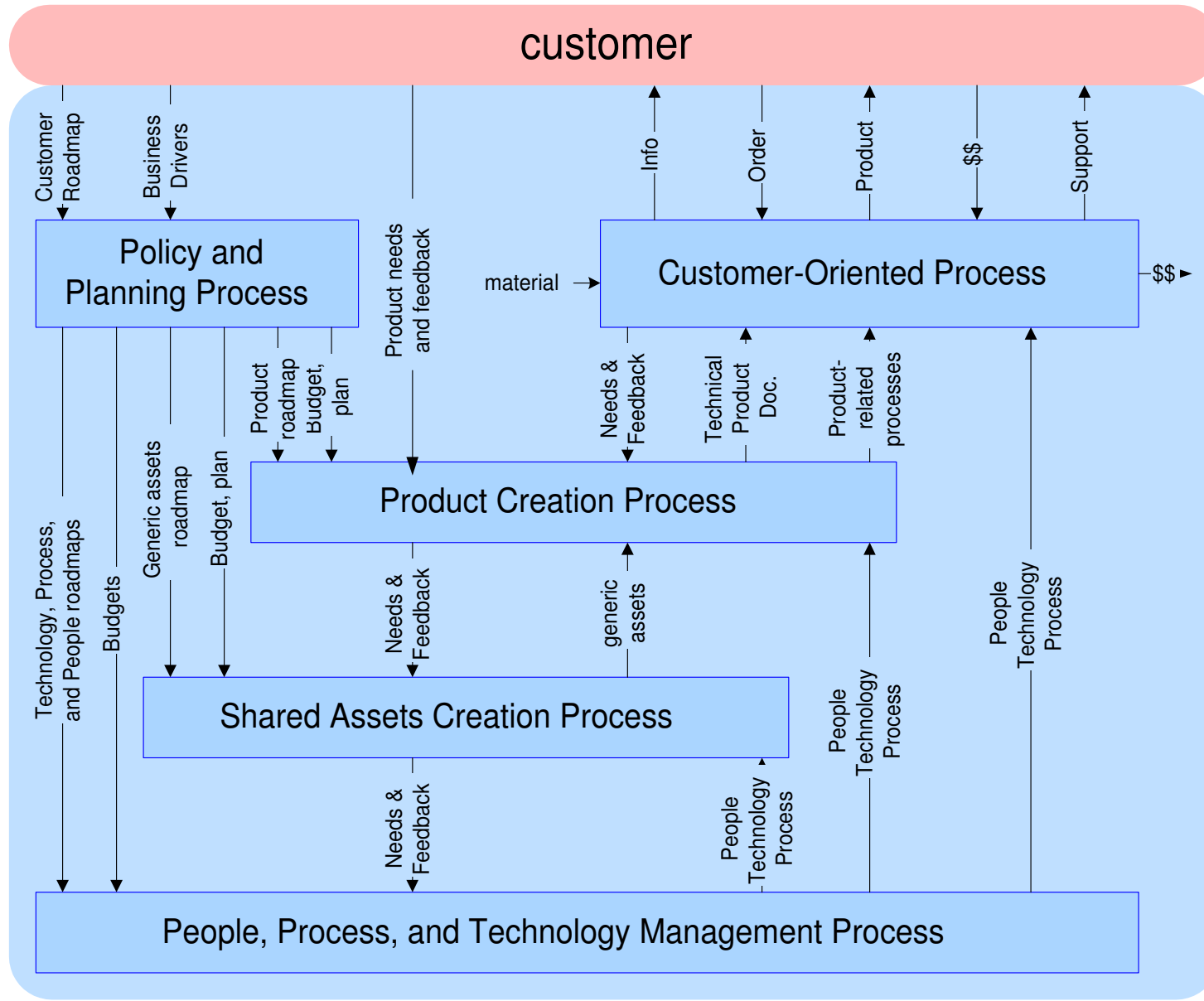
Drivers for Generic Developments



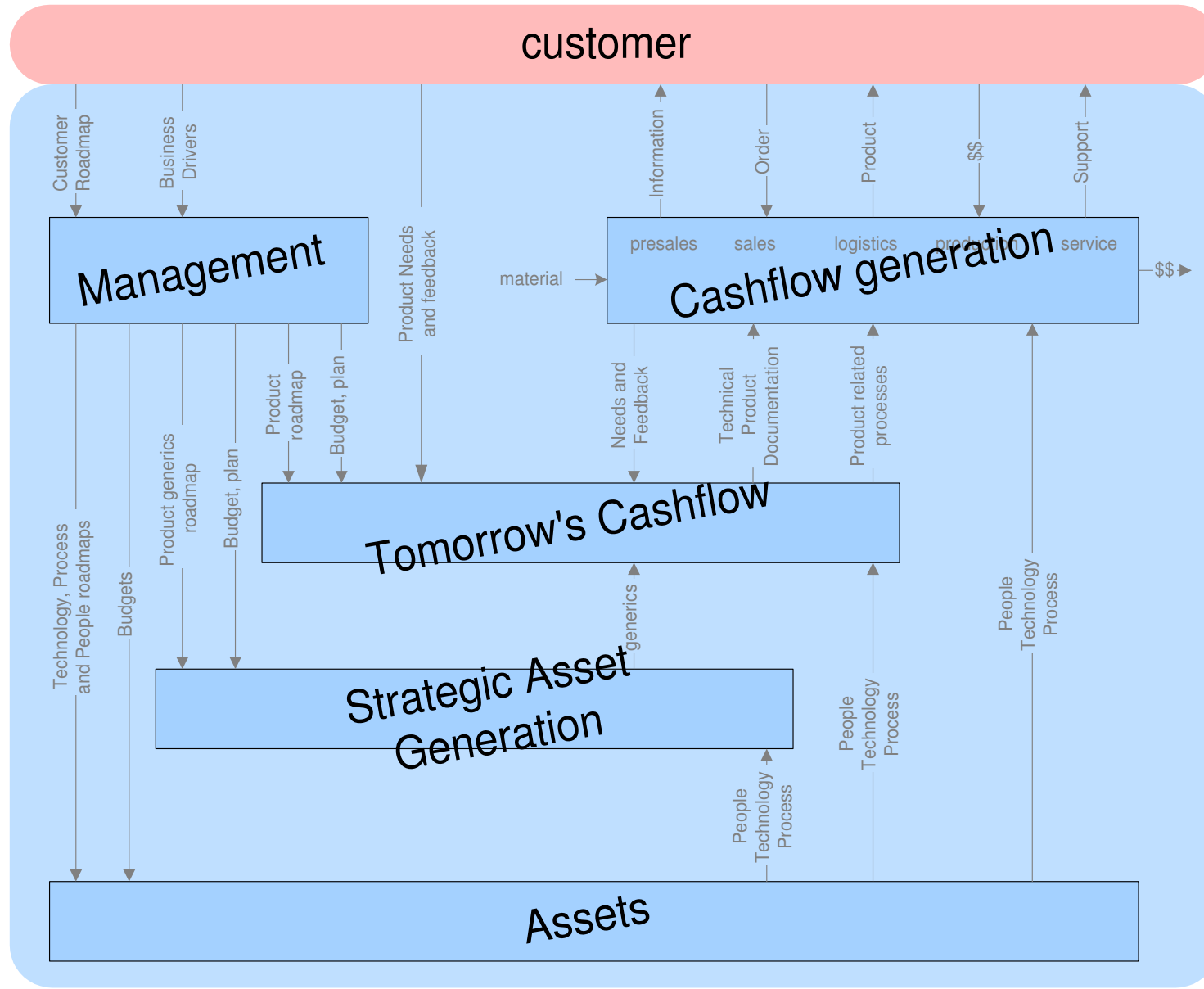
Granularity of generic developments shown in 2 dimensions



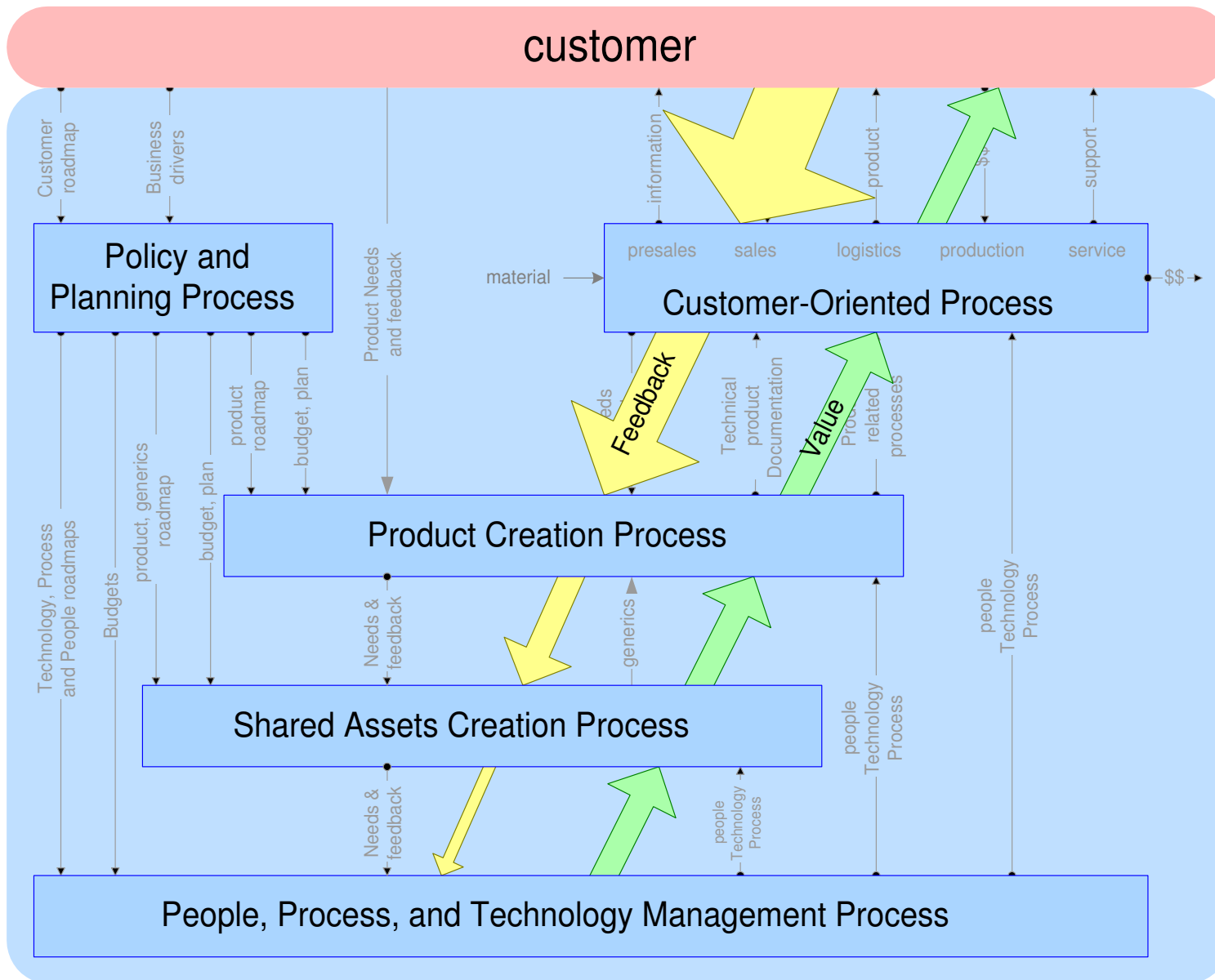
Modified Process Decomposition



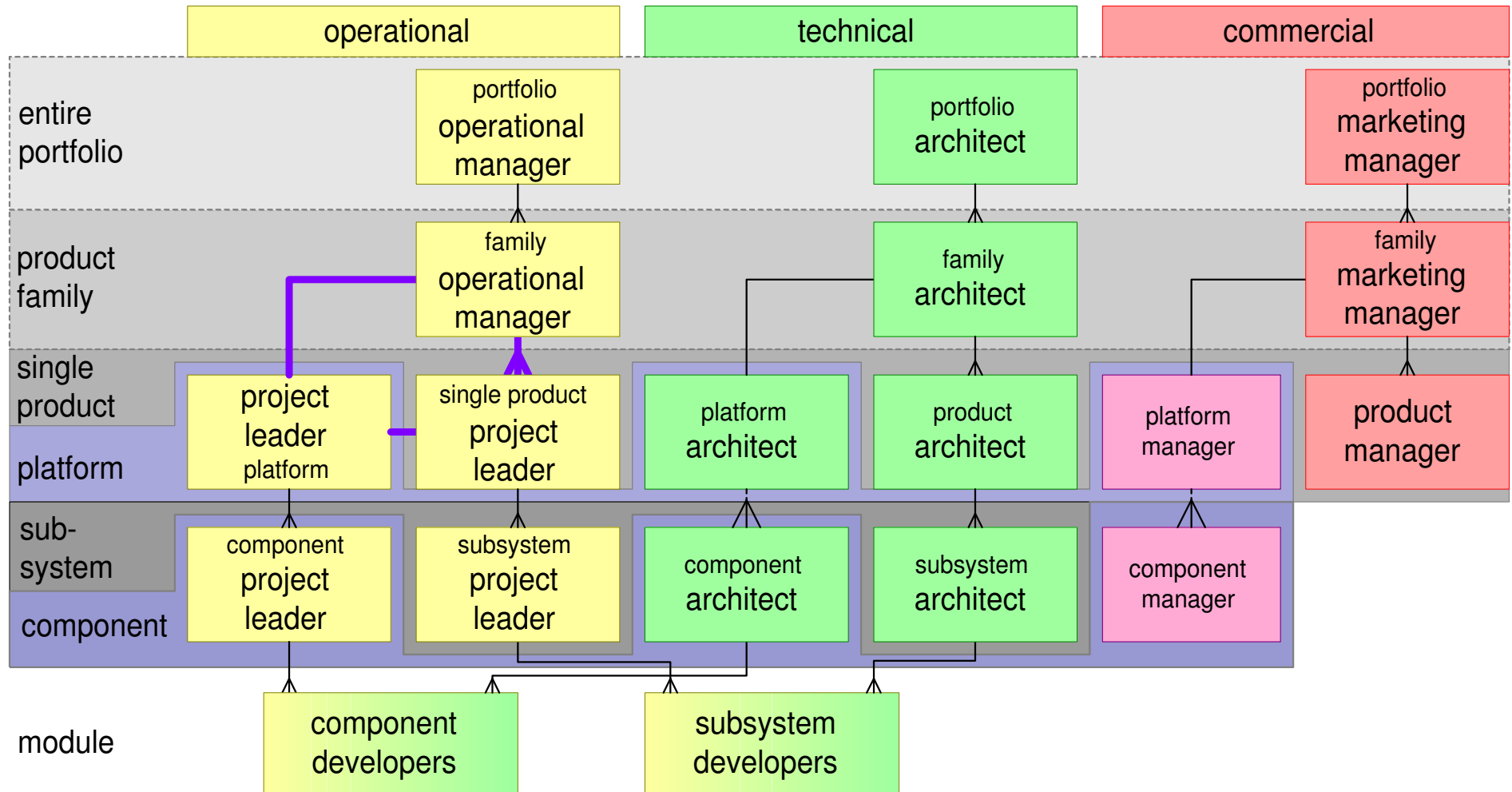
Financial Viewpoint on Process Decomposition



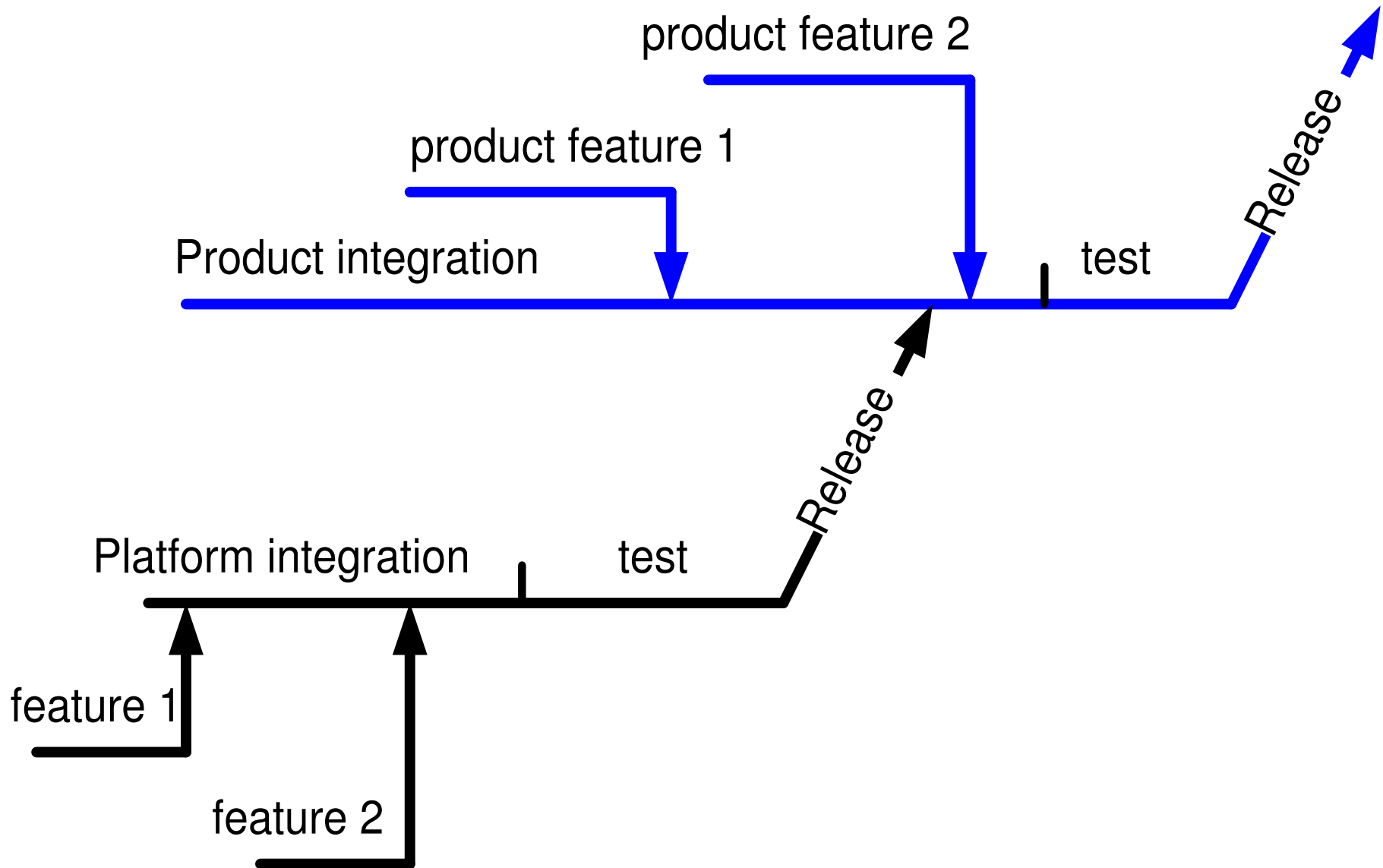
Value and Feedback Flow



Modified Operational Organization PCP



Propagation Delay Platform Feature to Market



Sources of Failure in Generic Developments

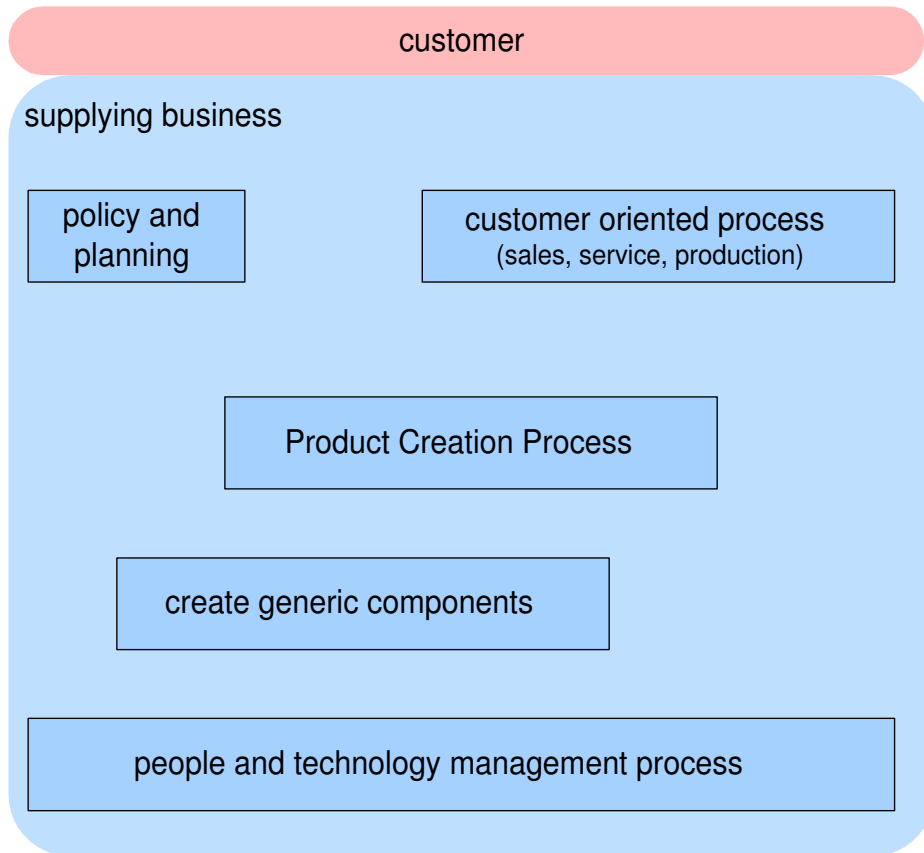
Technical

- 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?

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`

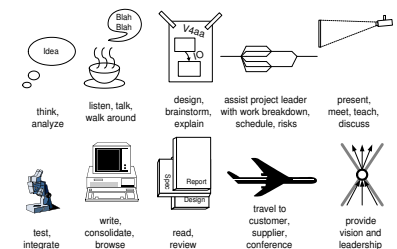
Abstract

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.

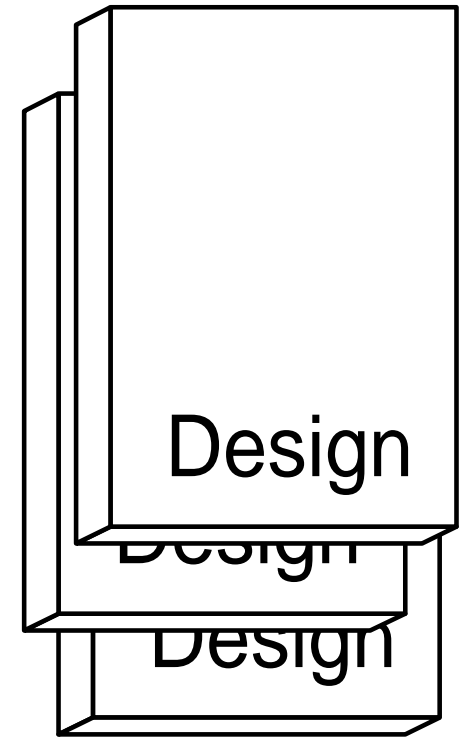
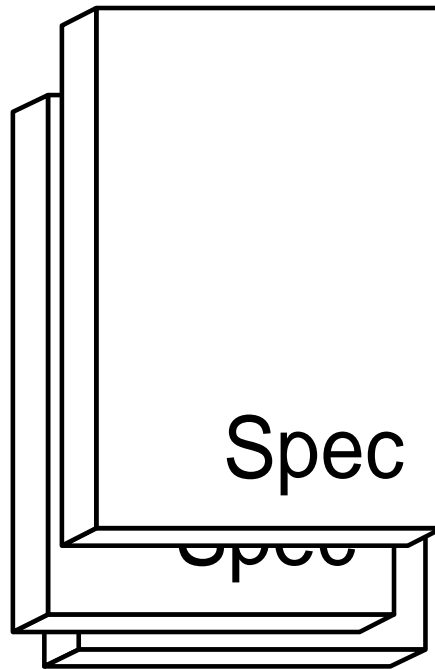
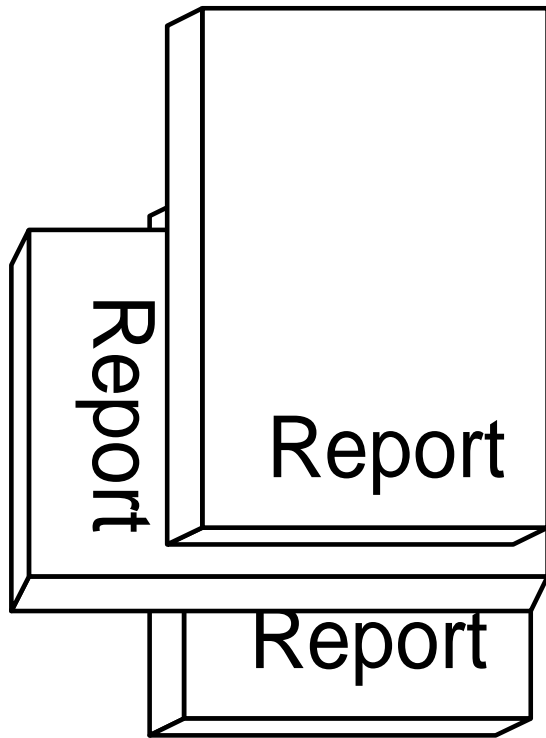
Distribution

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)*

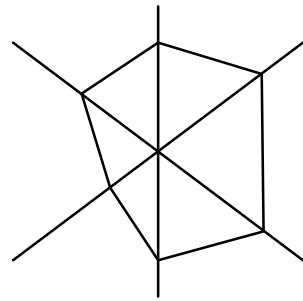
Verification Specification *(how the system will be verified)*

Verification Report *(the result of the verification)*

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

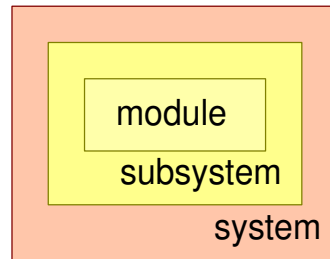
Roadmap

Responsibilities of the System Architect



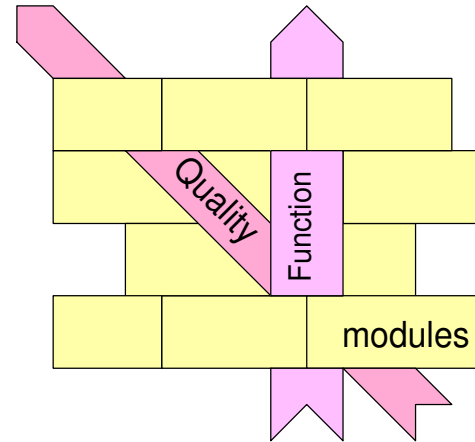
Balance

Requirement
Spec
Design
Realization



Consistency

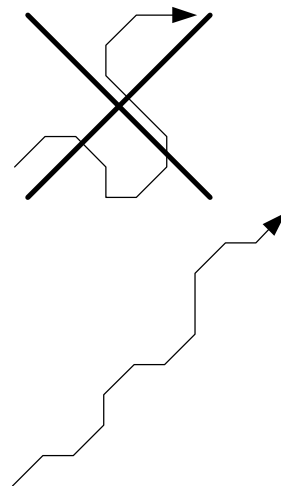
Decomposition
Integration



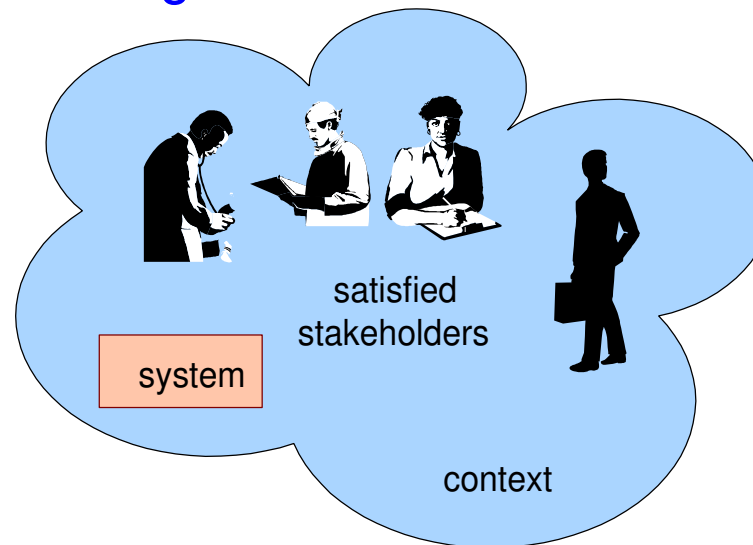
Overview

KISS

Elegance
Simple



Integrity

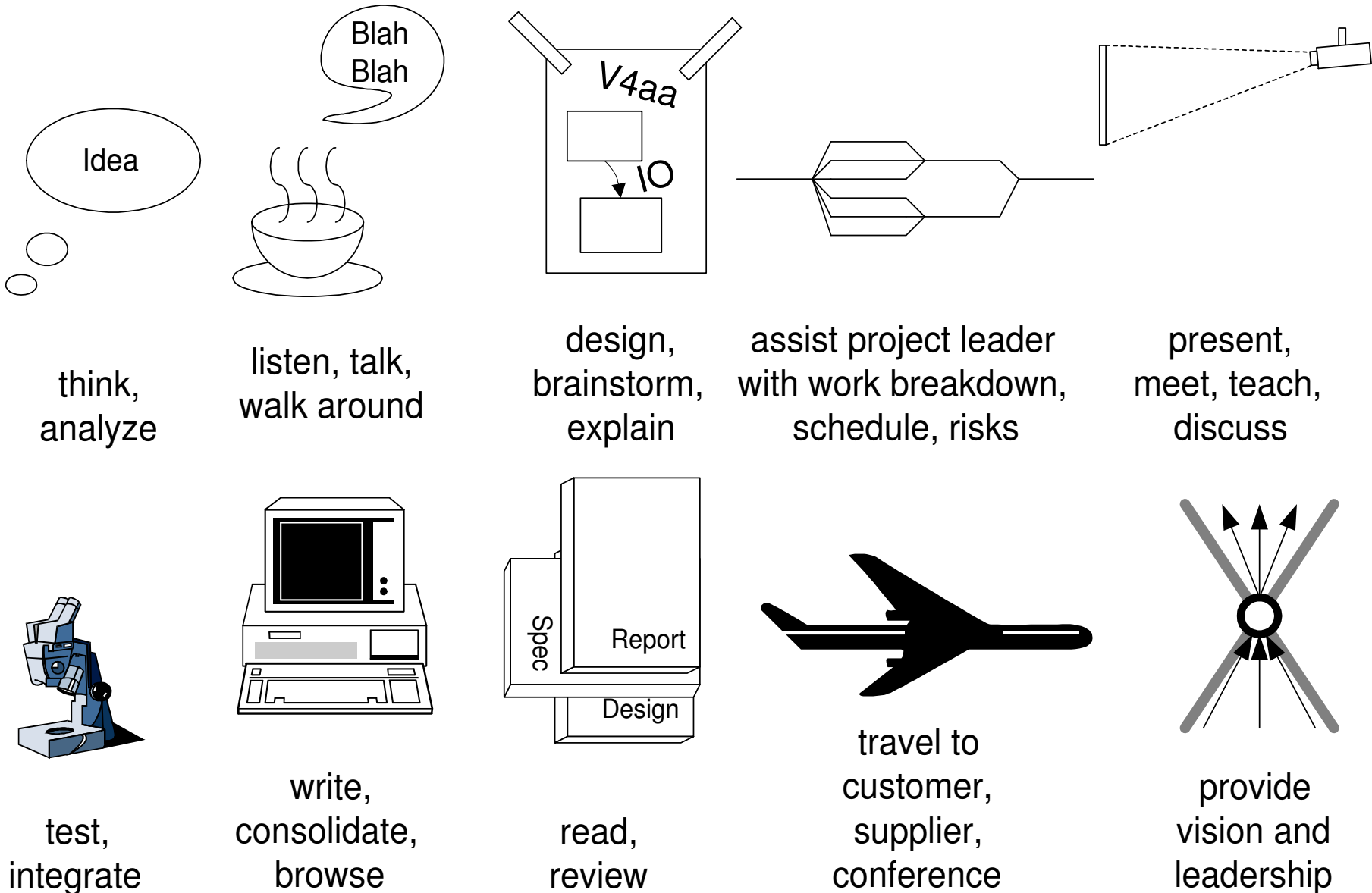


Fitting

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?

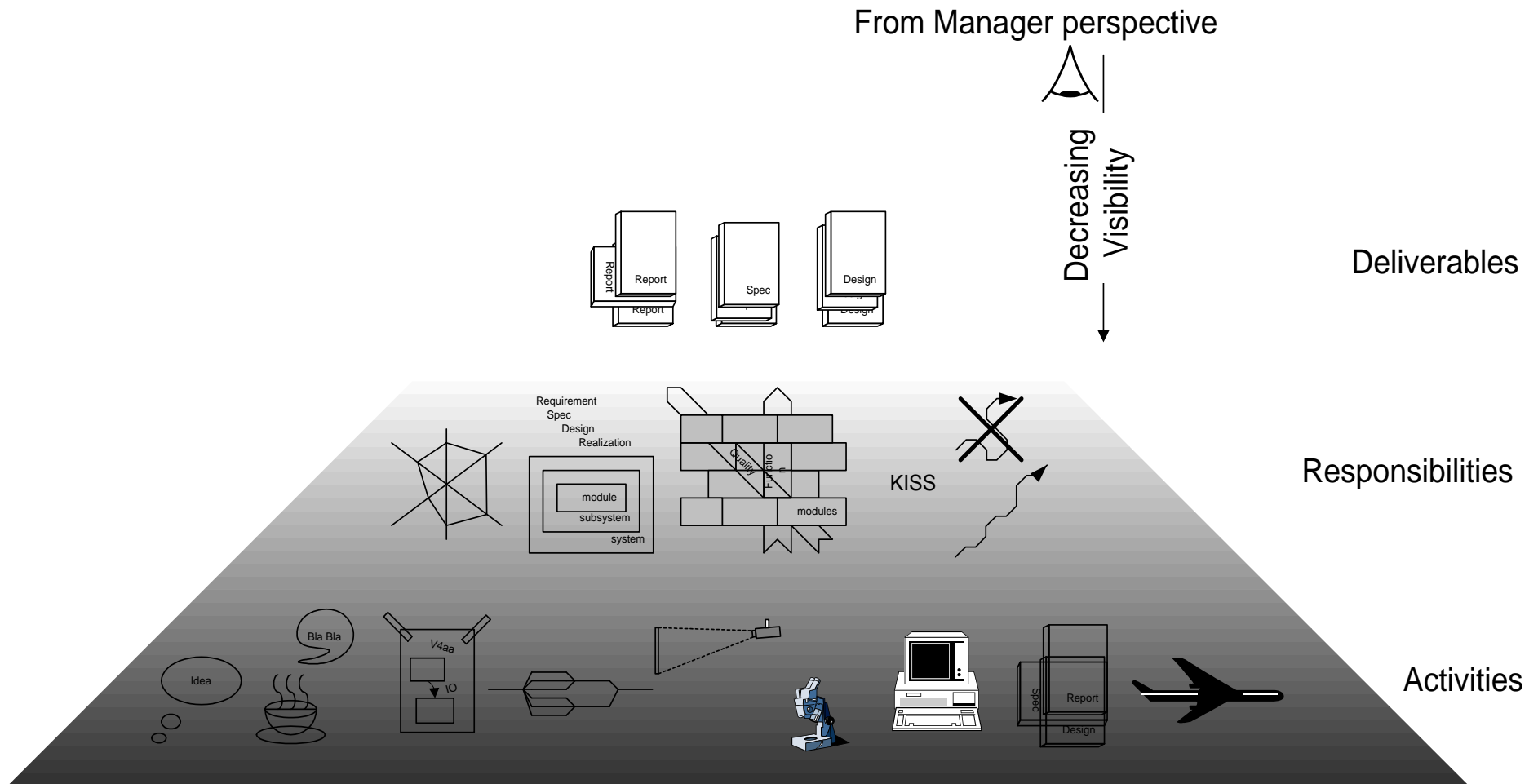


From Detail to Overview

	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	

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.

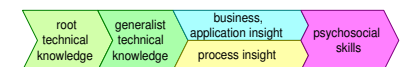
Distribution

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.

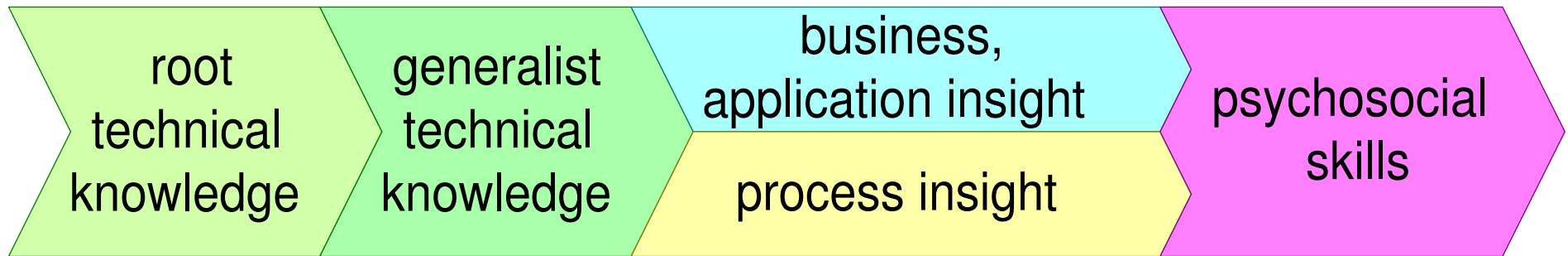
February 11, 2012

status: concept

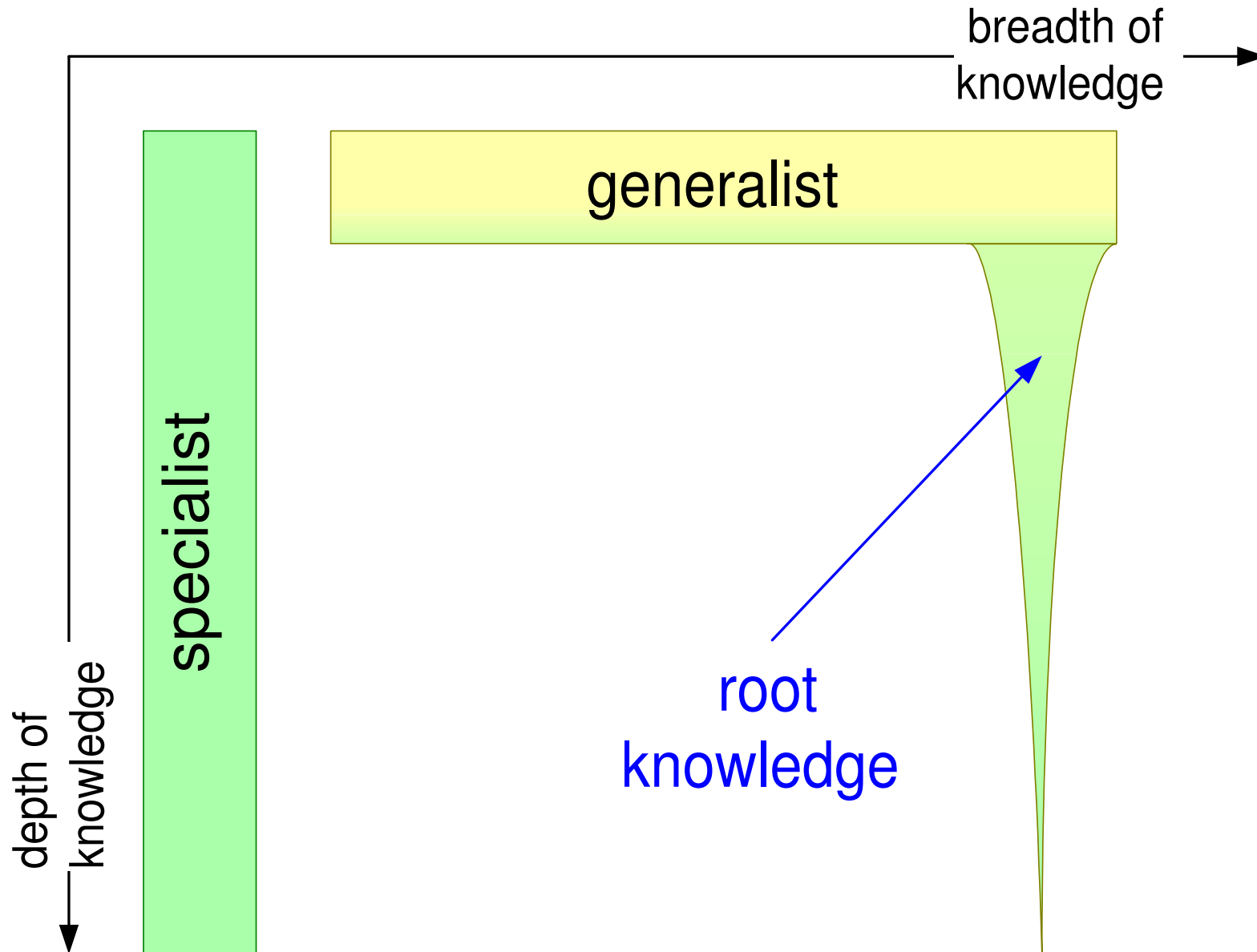
version: 1.1



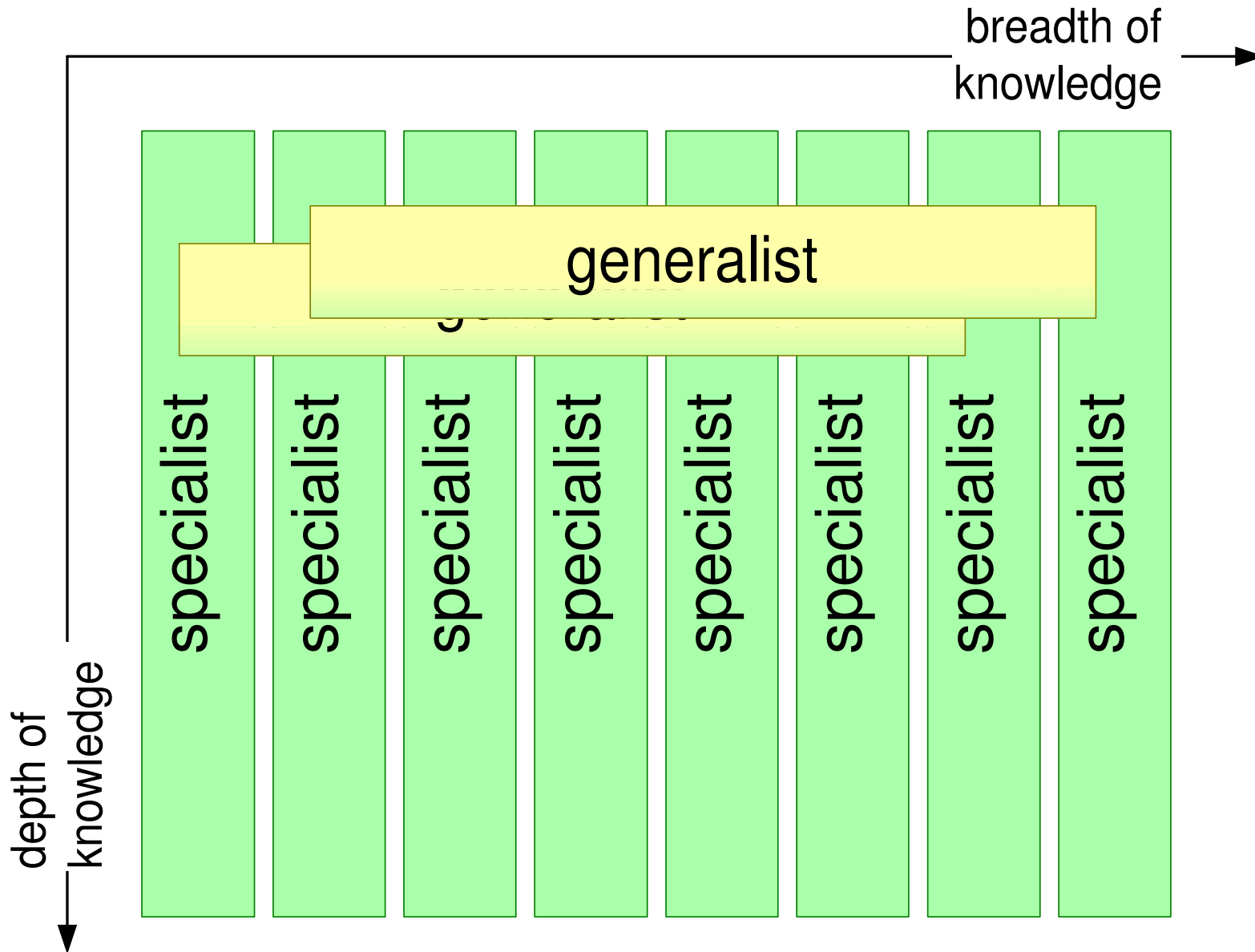
Typical Growth of a System Architect



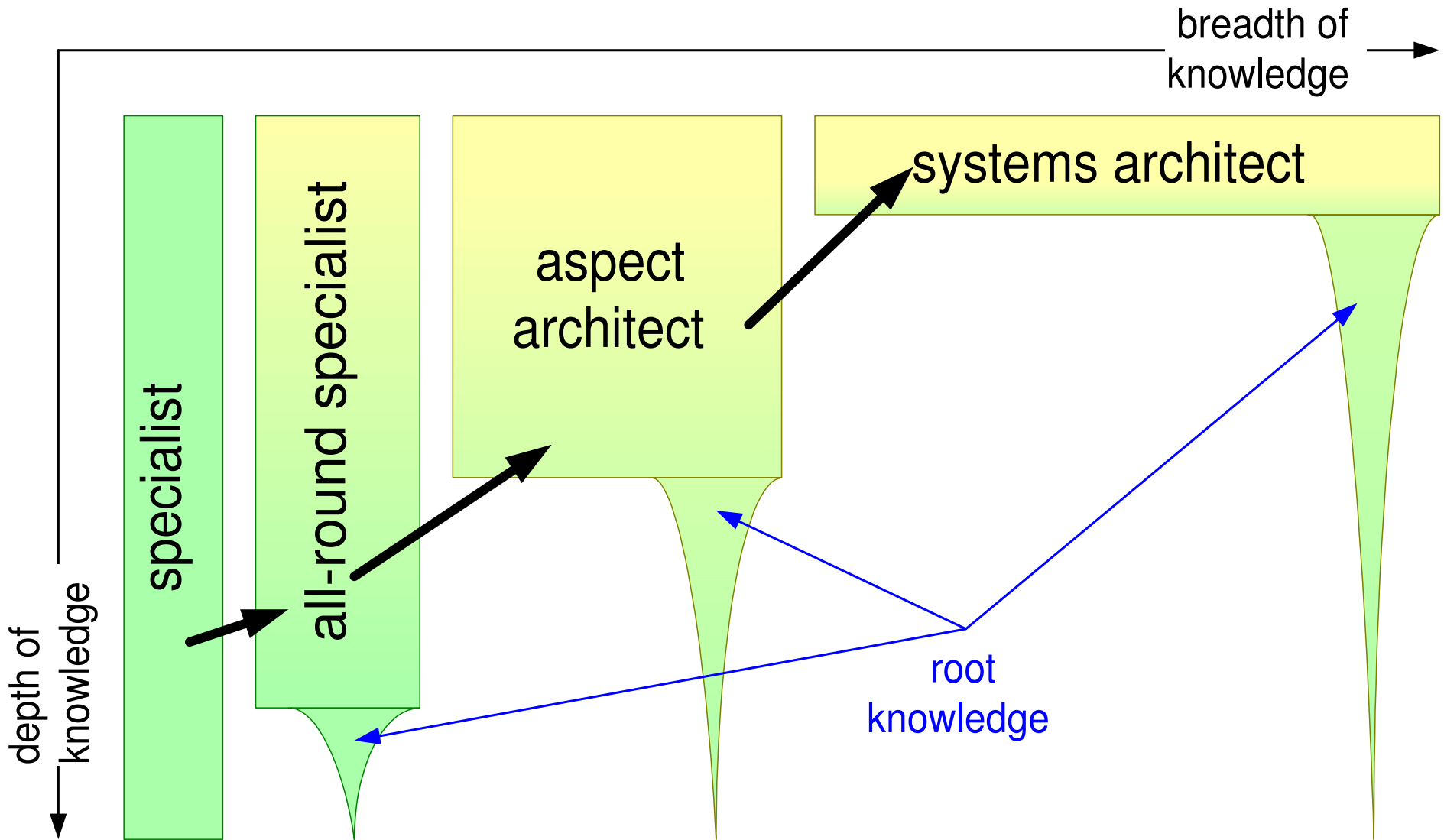
Generalist versus Specialist



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:

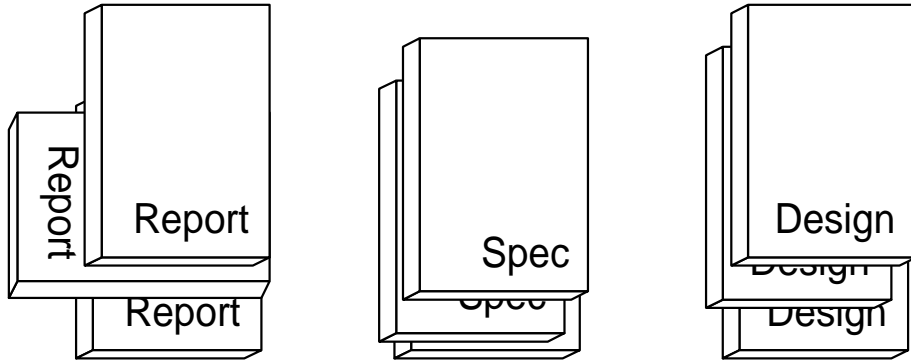
- 1 operational leader (project leader)
- 1 system architect
- 1 marketing manager
- 1 observer (optional)

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

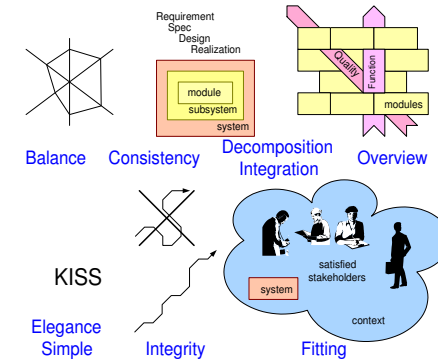
Present (max. 2 flips) the result and the process (the relation and interaction of the three roles).

Role and Task of a System Architect

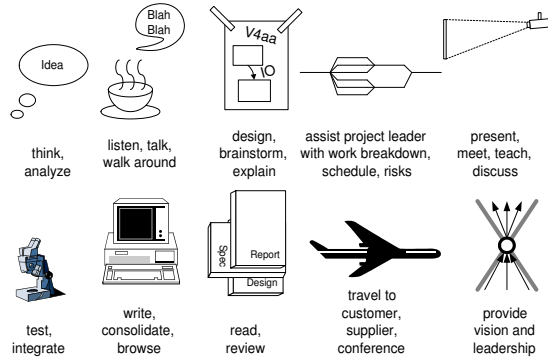
Deliverables



Responsibilities



Daily Activities

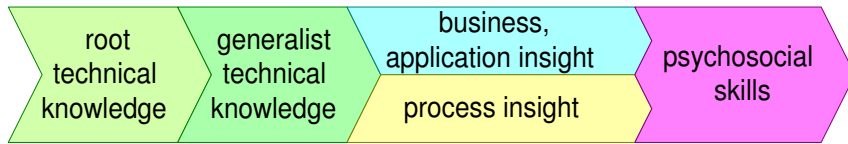


From detail to overview

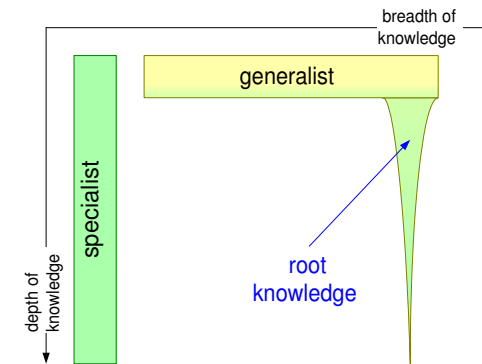
	Quantity per year (order-of-magnitude)	architect time per item
driving views	10	100 h
shared issues	10^2	1 h
touched details	10^4	0.5 – 10 min
seen details	$10^5 - 10^6$	0.1 – 1 sec
product details	$10^7 - 10^{10}$	
real-world facts	infinite	

Personal characteristics of a System Architect

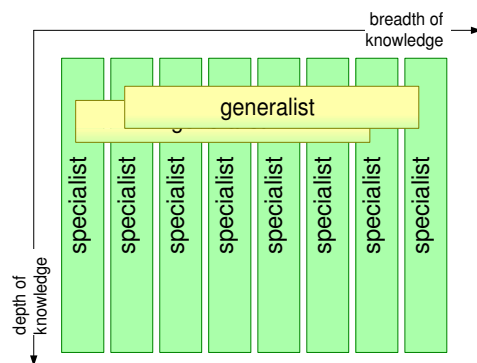
Typical growth of a Architect



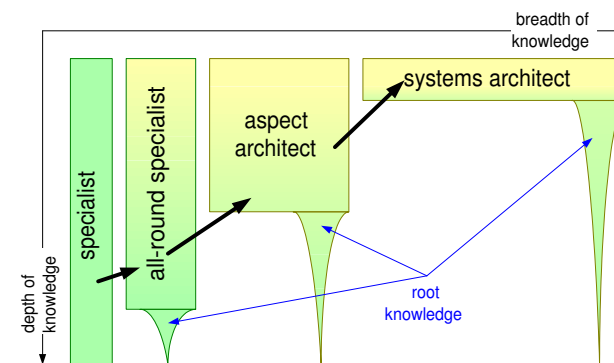
Generalist vs Specialist



Complementary Roles



Role Spectrum



Module Supporting Processes

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

Abstract

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

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: 1.2



Embedded Systems
INSTITUTE

Granularity of Documentation

by *Gerrit Muller* Buskerud University College

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

`www.gaudisite.nl`

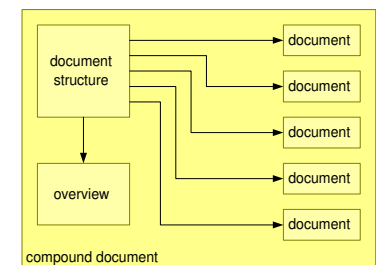
Abstract

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.

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: 1.2



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
viewing
printing
searching

easy
fast

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

single author

limited amount of reviewers

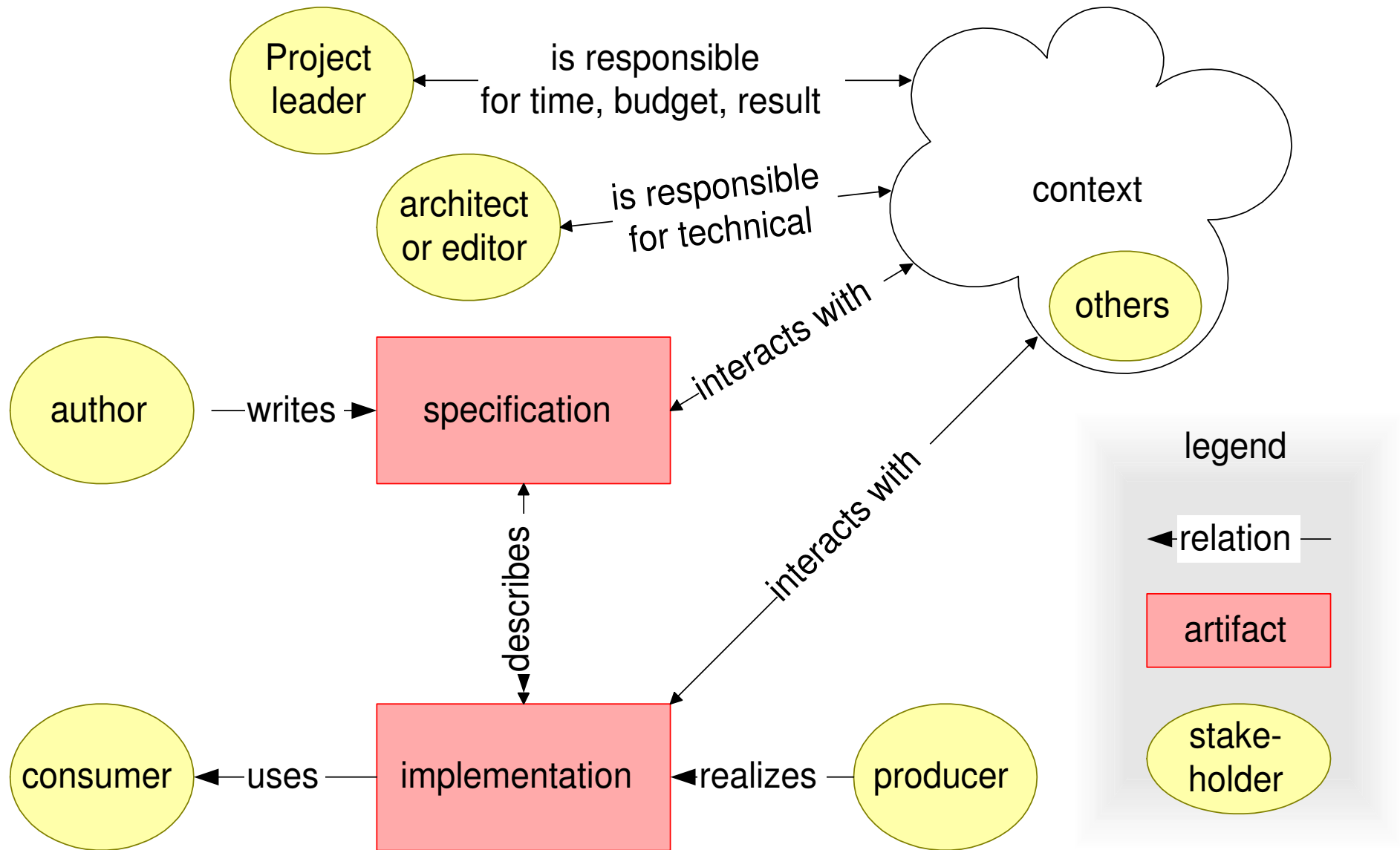
well defined documentation structure

overview specifications at higher
aggregation levels

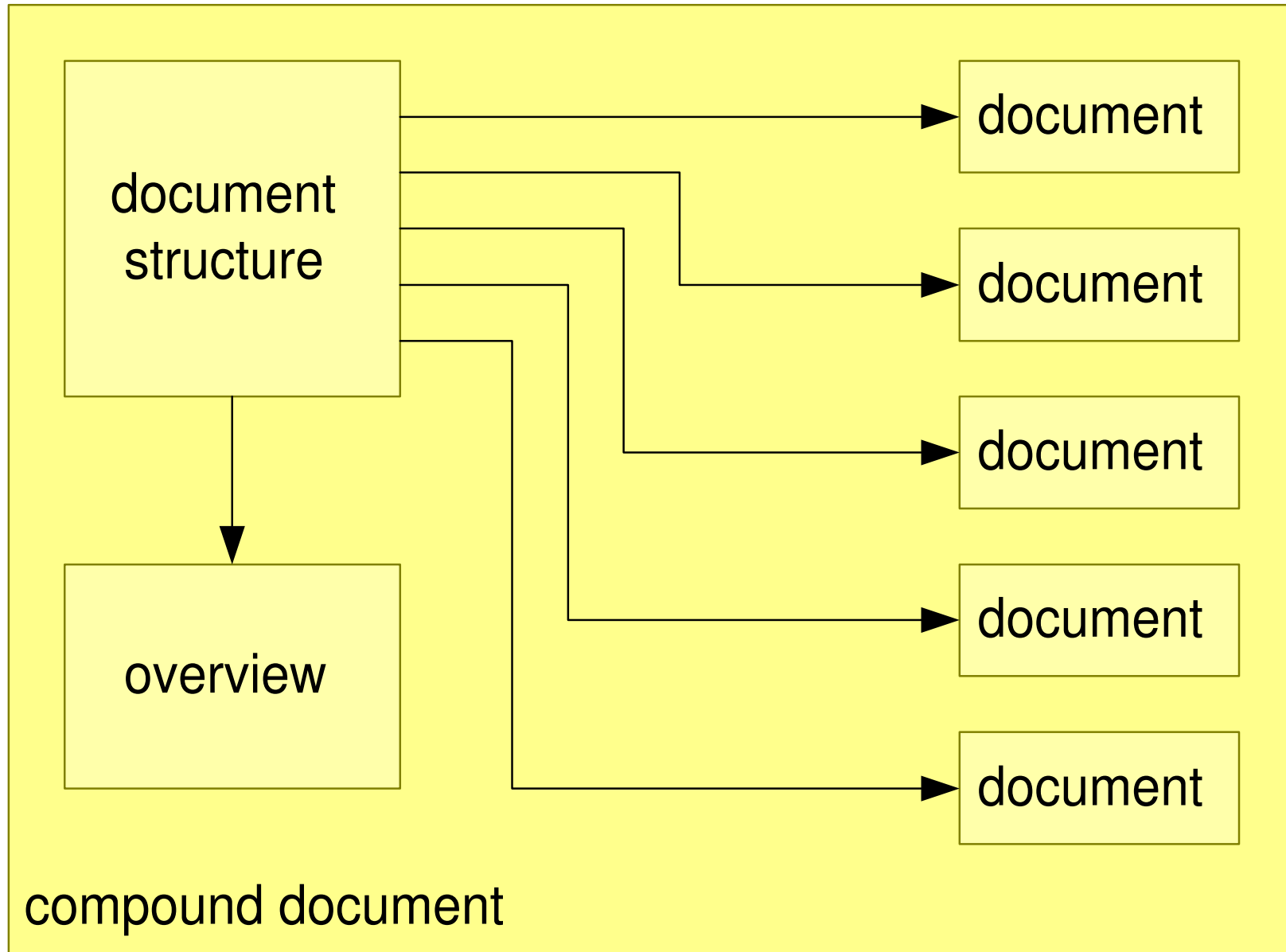
recursive application of structure and
overview

delegation of review process

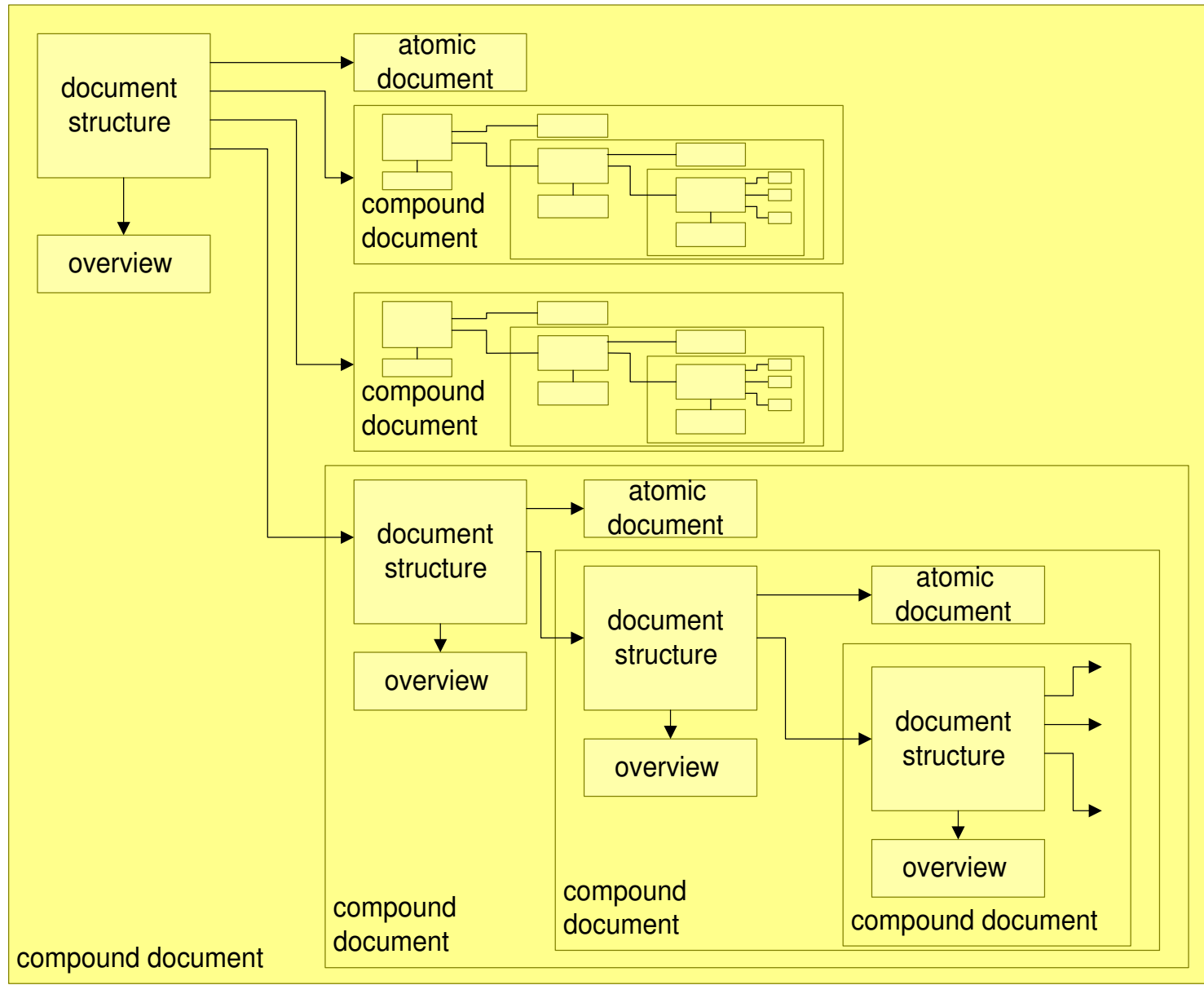
The Stakeholders of a Single Document



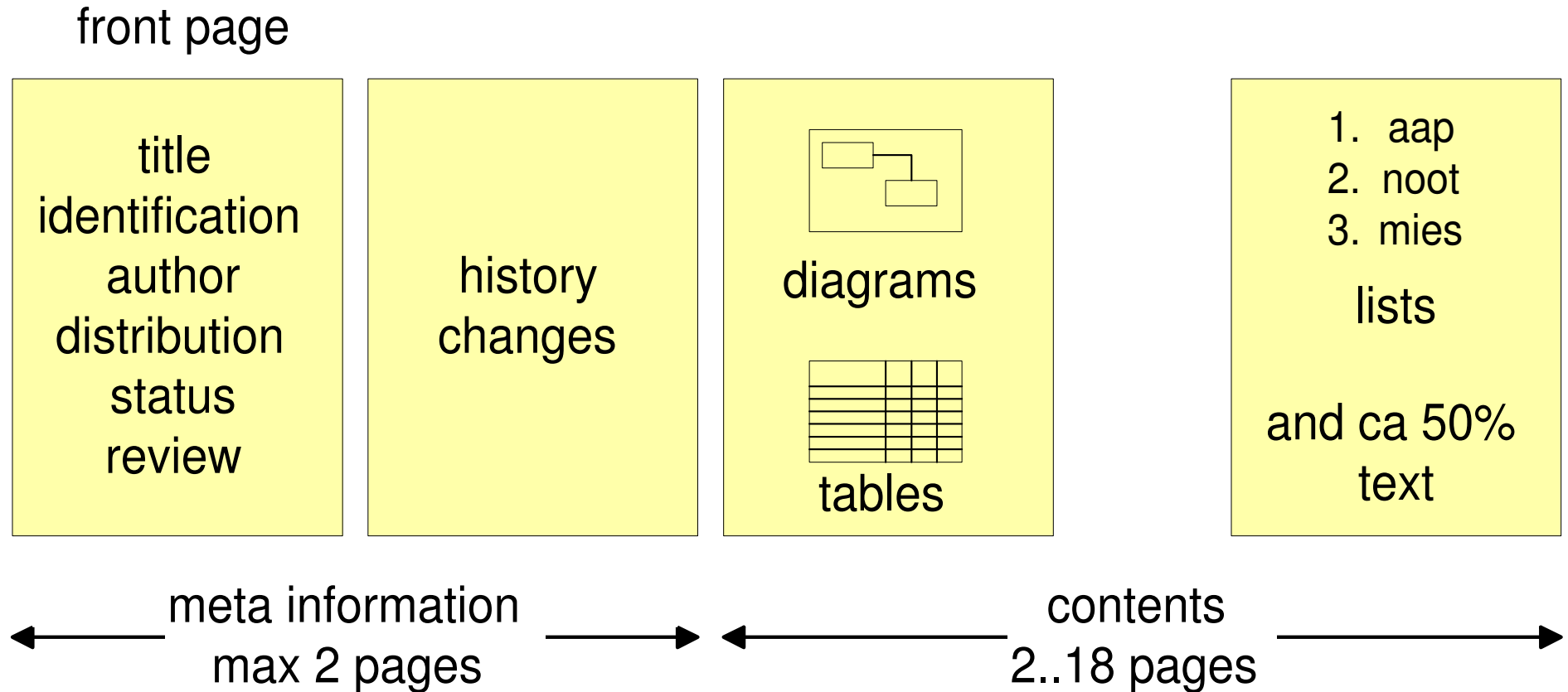
Decomposition of Large Documents



Documentation Tree by Recursive Decomposition



Payload: the Ratio between Content and Overhead



LEAN and A3 Approach to Supporting Processes

by *Gerrit Muller* Buskerud University College

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

`www.gaudisite.nl`

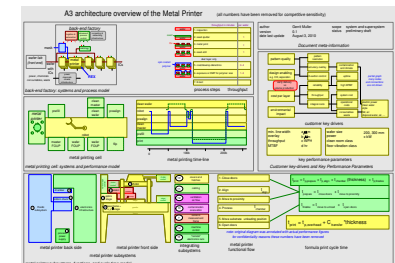
Abstract

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.

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: 0.1



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.

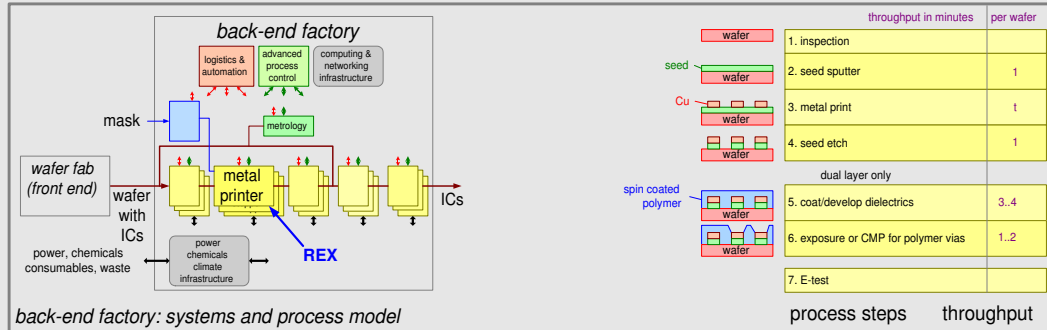
Continuous improvement and learning as cultural value .

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

Example of A3 Architecture Overview

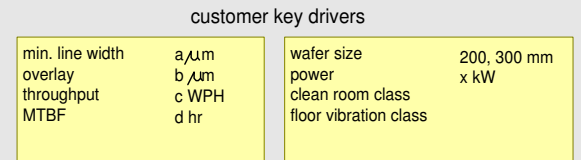
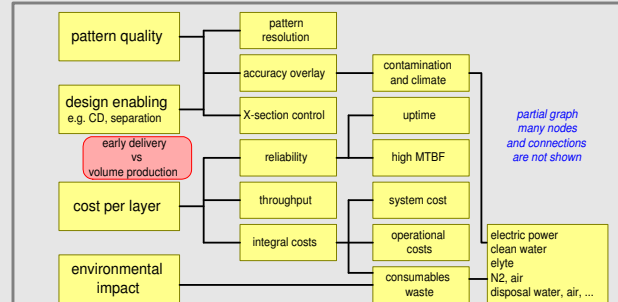
A3 architecture overview of the Metal Printer

(all numbers have been removed for competitive sensitivity)

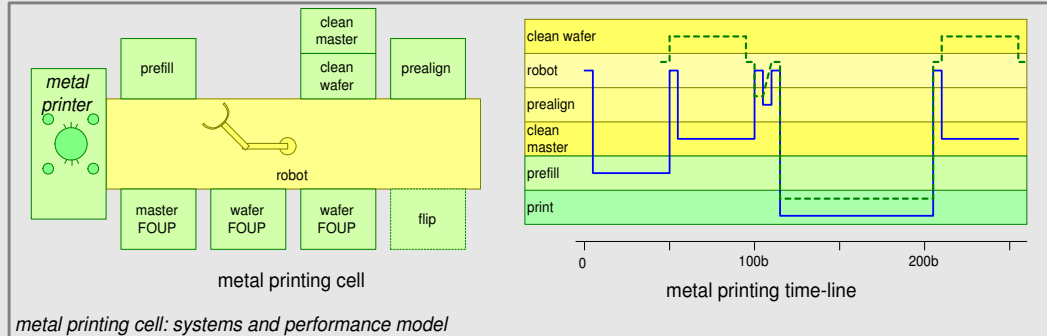


author Gerrit Muller
 version 0.1
 date last update August 3, 2010
 scope system and supersystem
 status preliminary draft

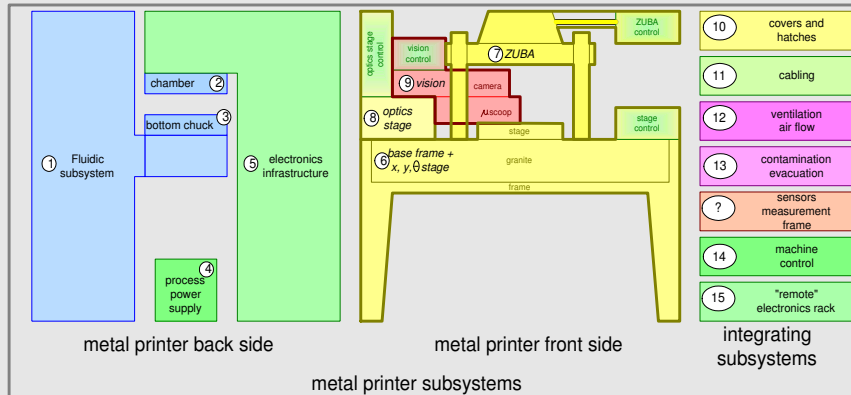
Document meta-information



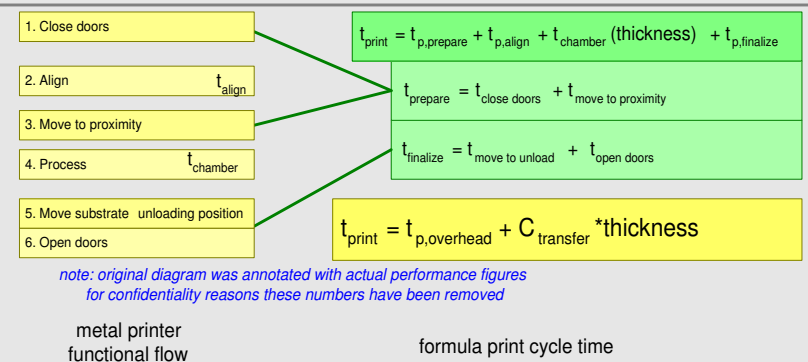
Customer key-drivers and Key Performance Parameters



metal printing cell: systems and performance model



metal printer subsystems, functions, and cycle time model



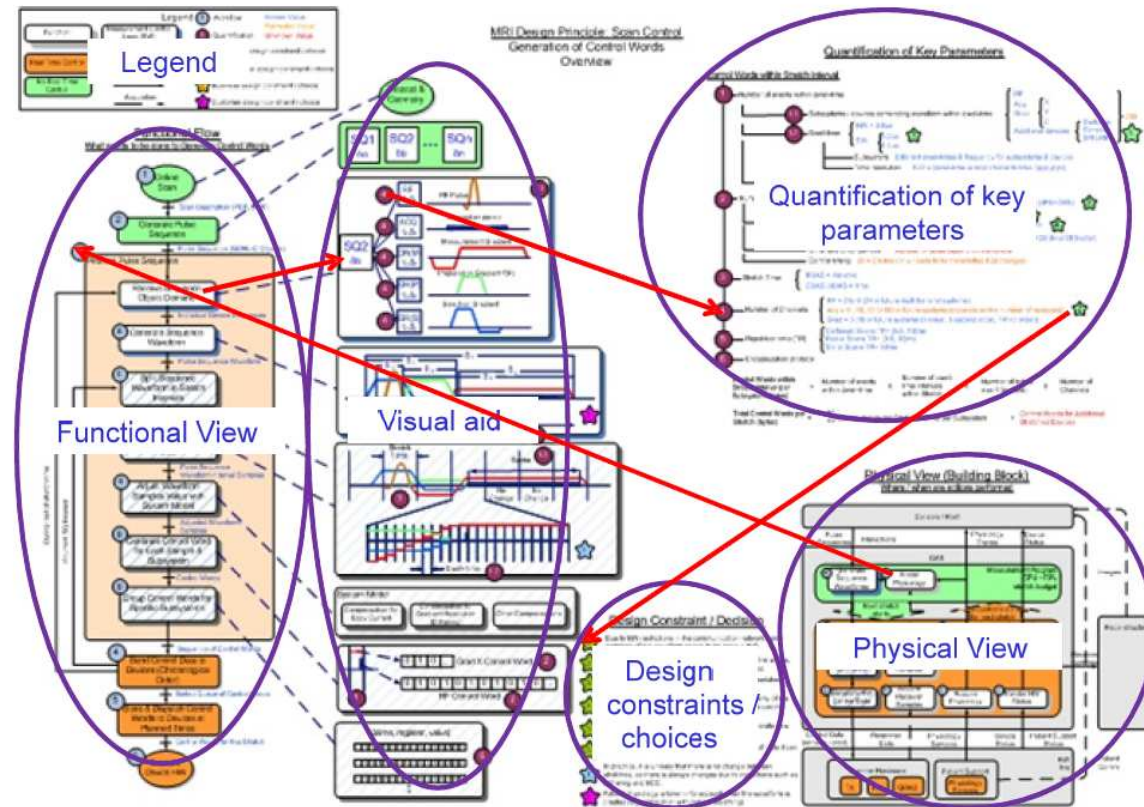
note: original diagram was annotated with actual performance figures for confidentiality reasons these numbers have been removed

multiple related views

quantifications

one topic
per A3

capture
"hot" topics



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

digestible
(size limitation)

practical
close to stakeholder experience

Light Weight Review Process

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

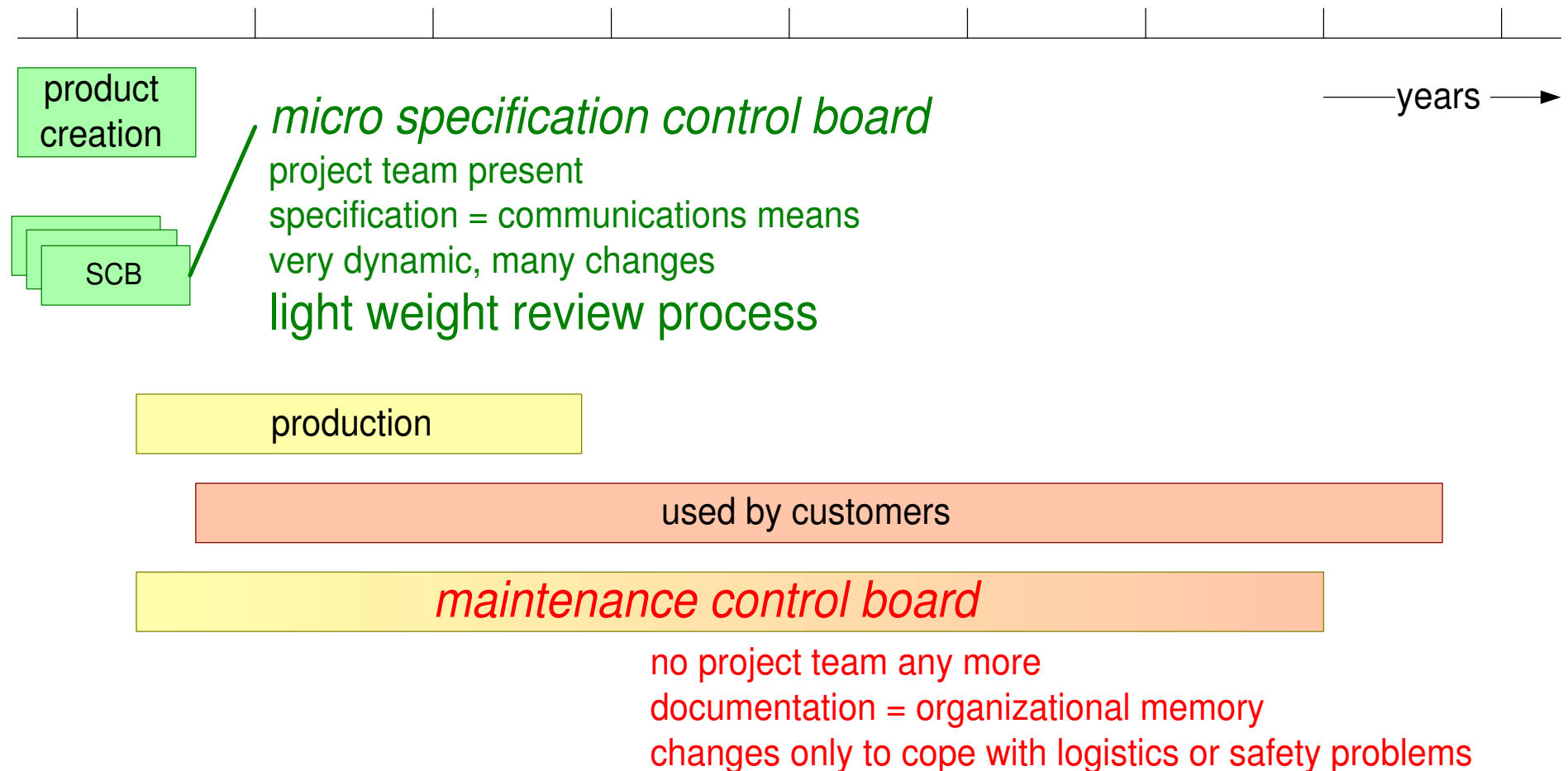
Distribution

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

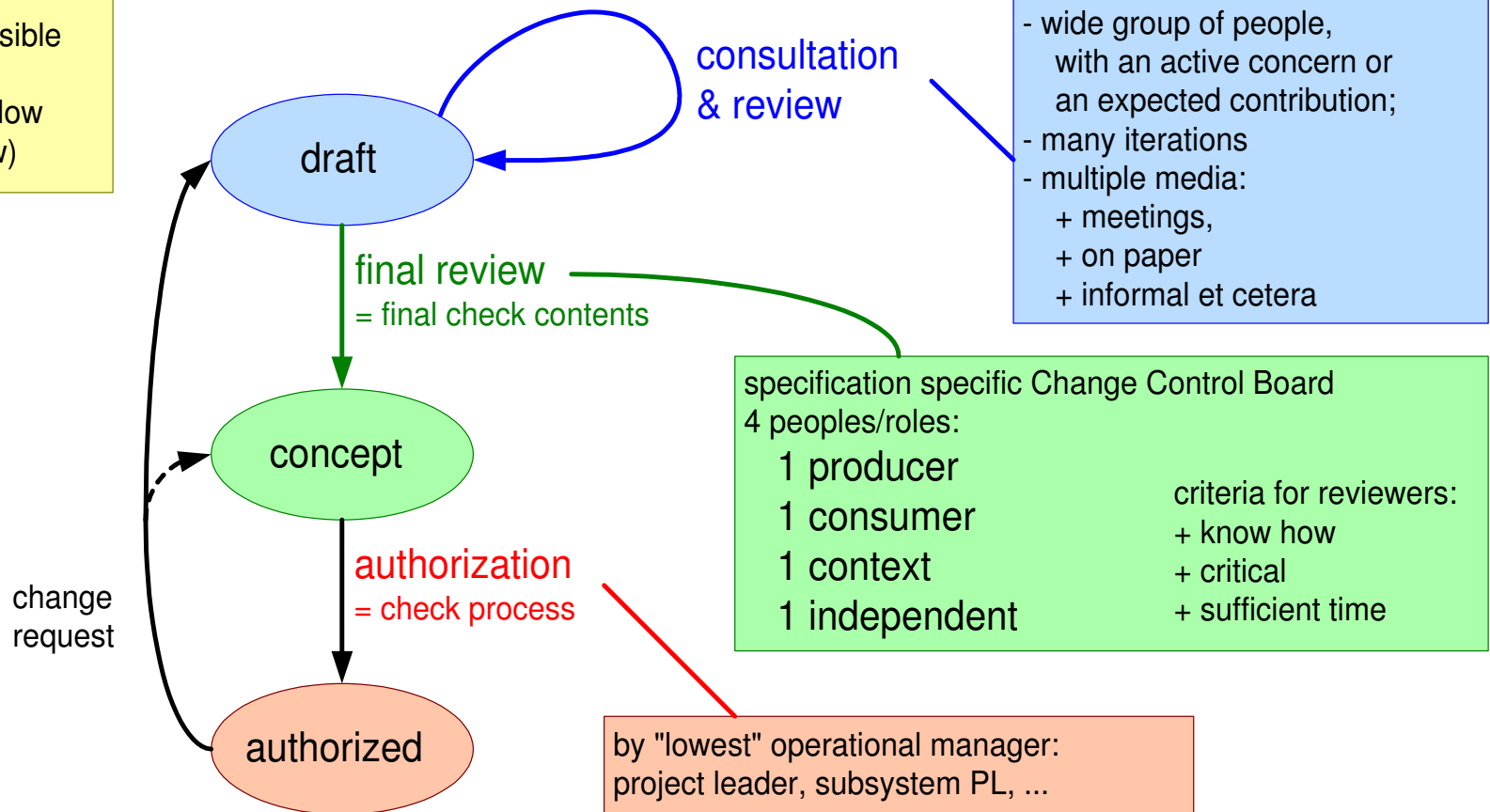


Product Life Cycle and Change Management



Light Weight Specification Review Process

the author is responsible for contents and organization of the flow (consults and review)



Template How To

by *Gerrit Muller* Embedded Systems Institute

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

`www.gaudisite.nl`

Abstract

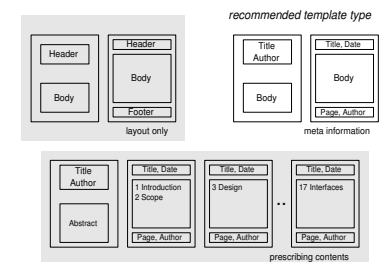
The introduction of a new process (way of working) is quite often implemented by supplying ready-to-go tools and templates. This implementation mainly serves the purpose of a smooth introduction of the new process.

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.

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: 1.6



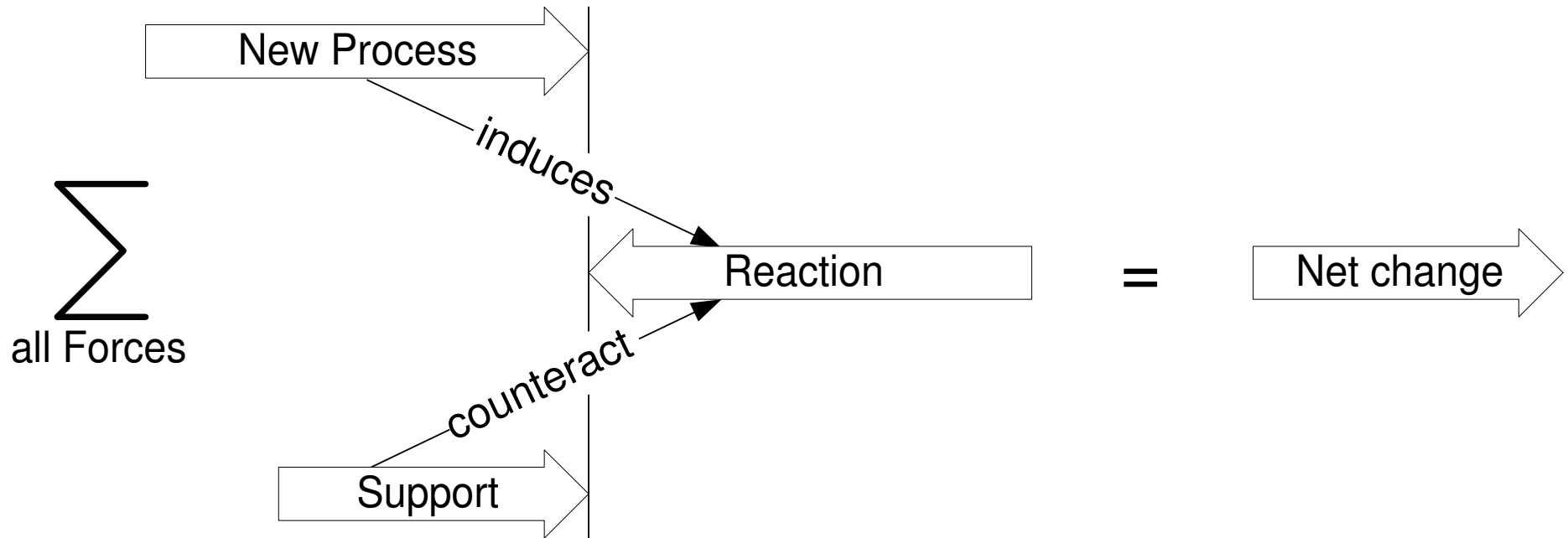
Rationale for Templates

- 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)

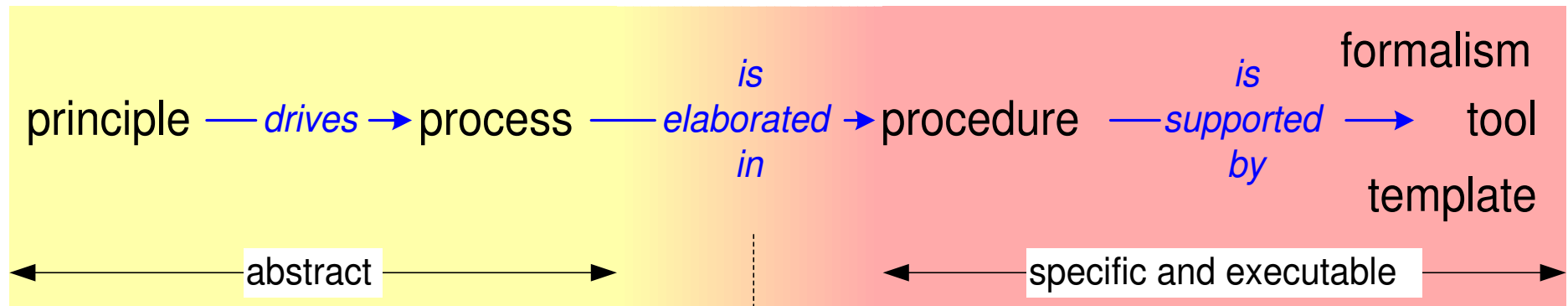
Bogus Arguments for Templates

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

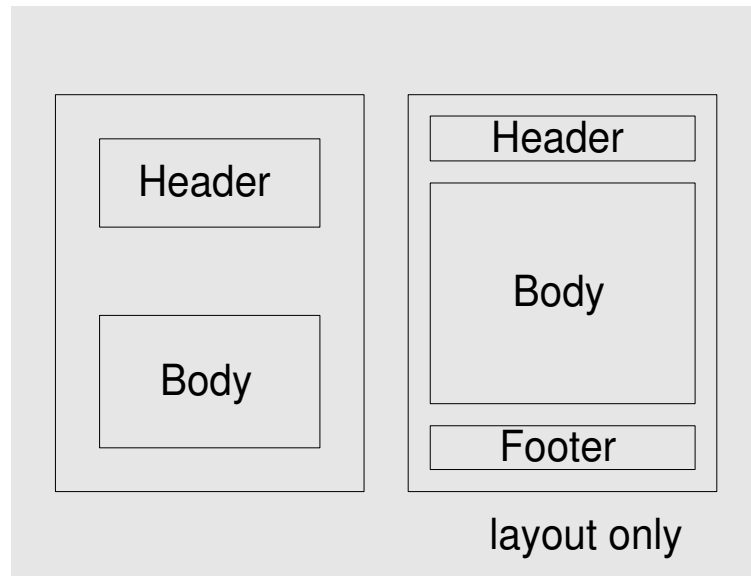
Forces of Change: Action = - Reaction



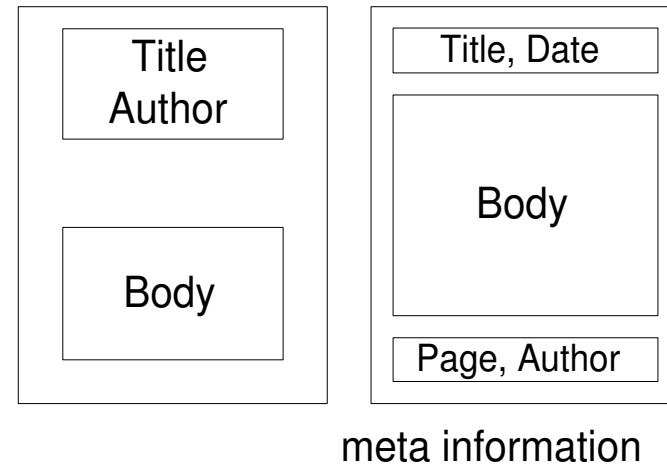
Template as Support for Process



Types of Templates



recommended template type



Recommendation

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

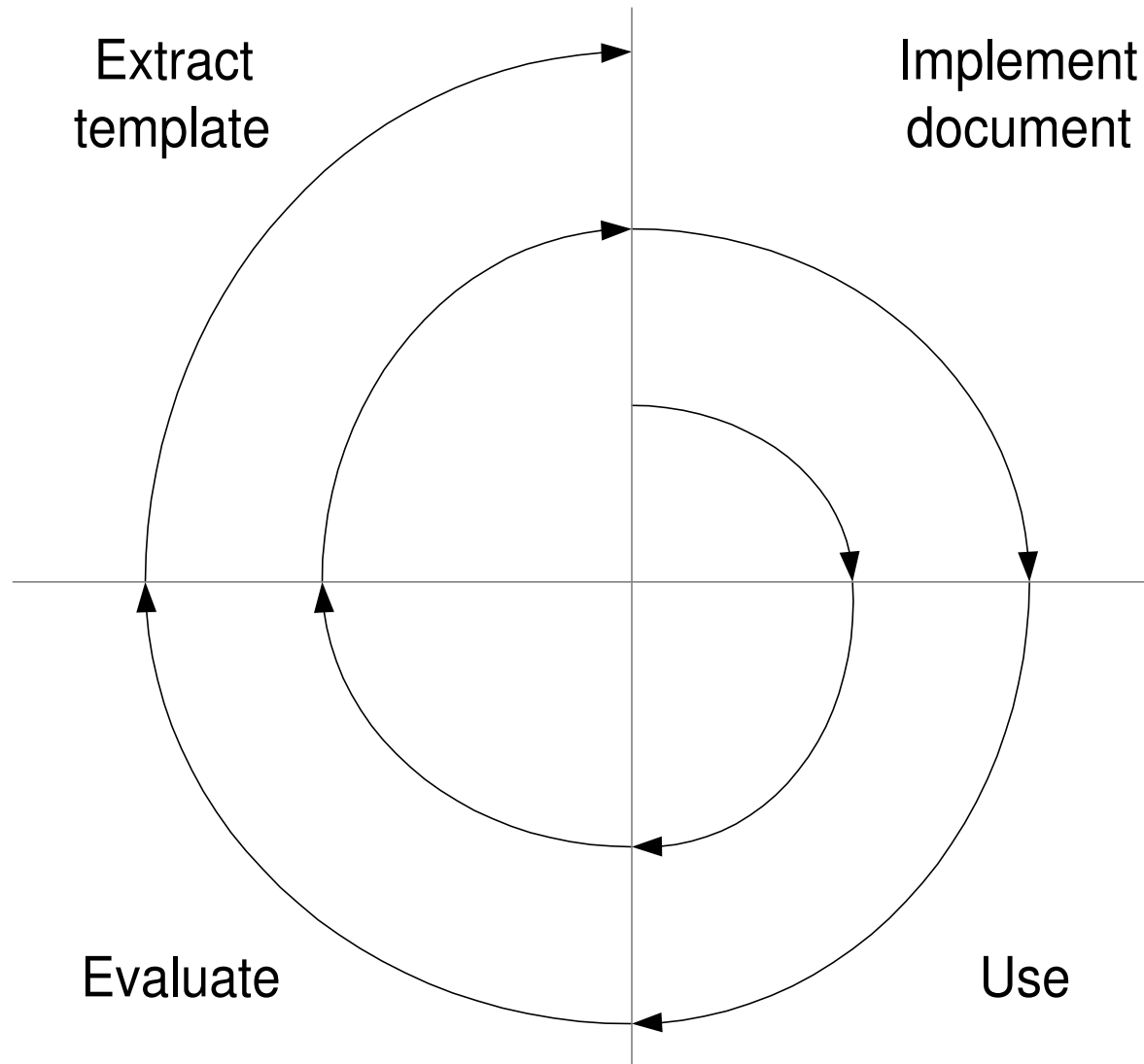
- Use templates for meta-information.
- Use checklists for structure and contents.

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* Buskerud University College

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

`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.

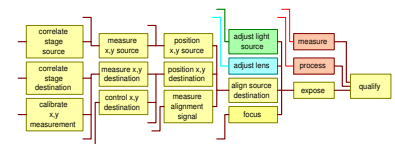
Distribution

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.

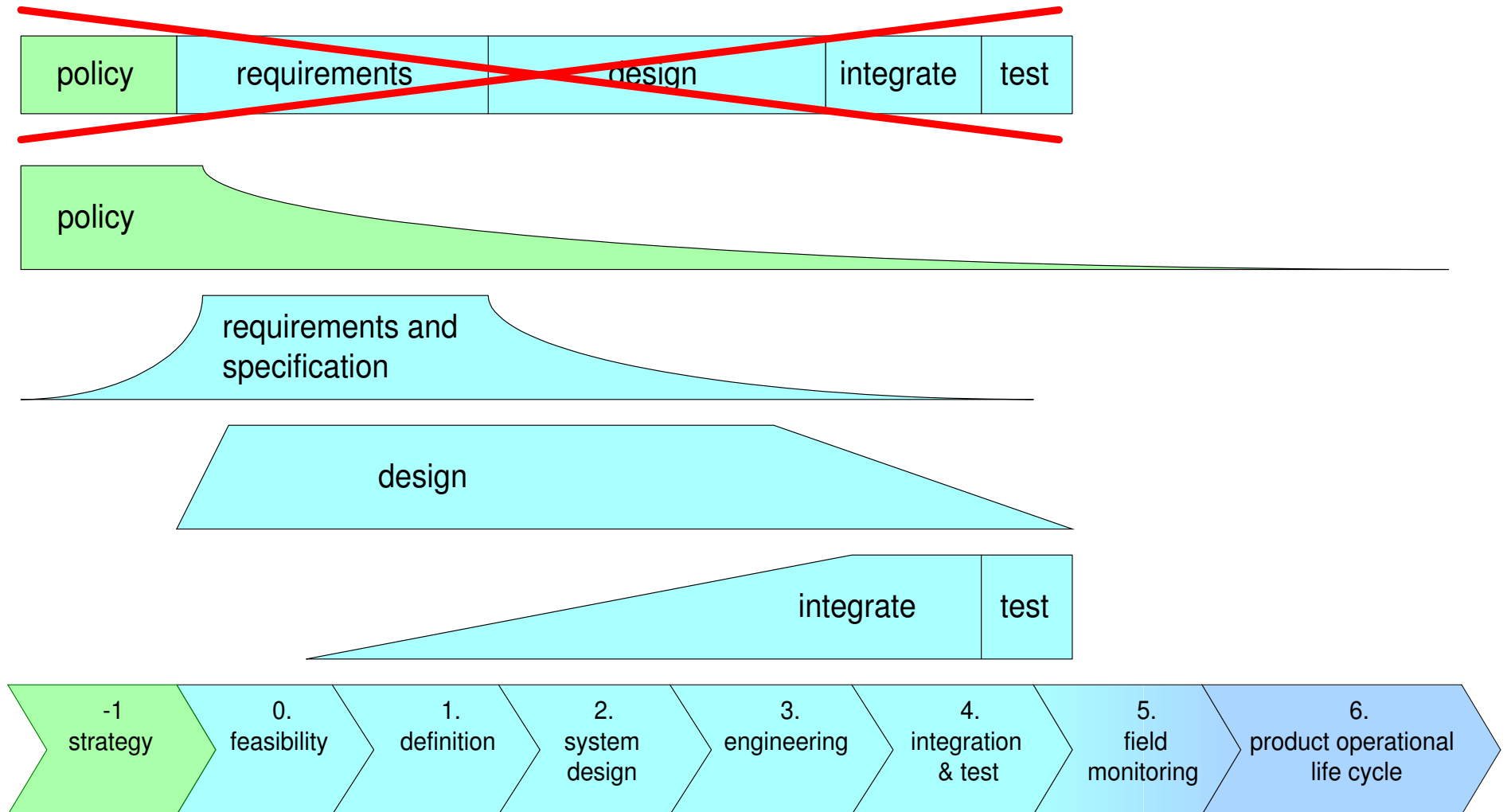
February 11, 2012

status: concept

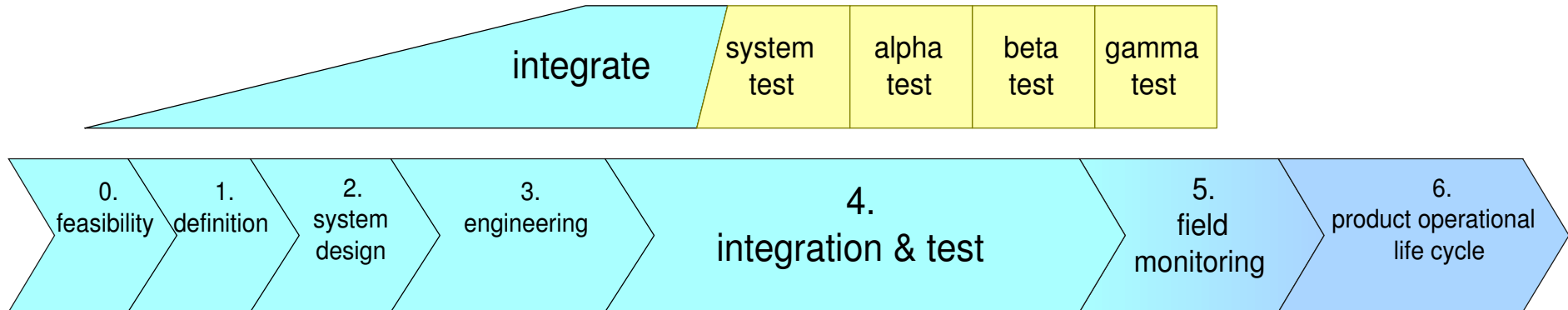
version: 0.2



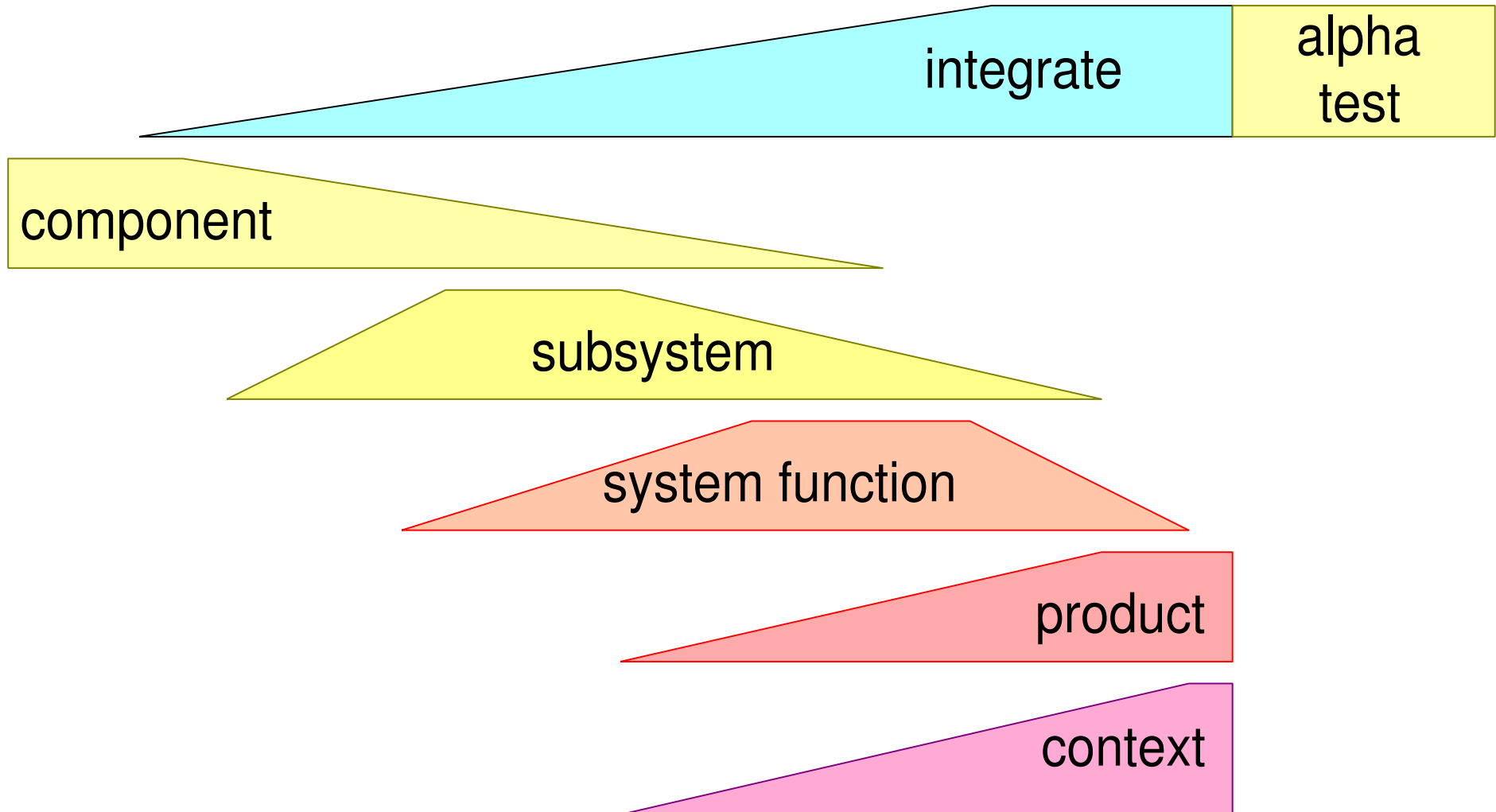
Typical Concurrent Product Creation Process



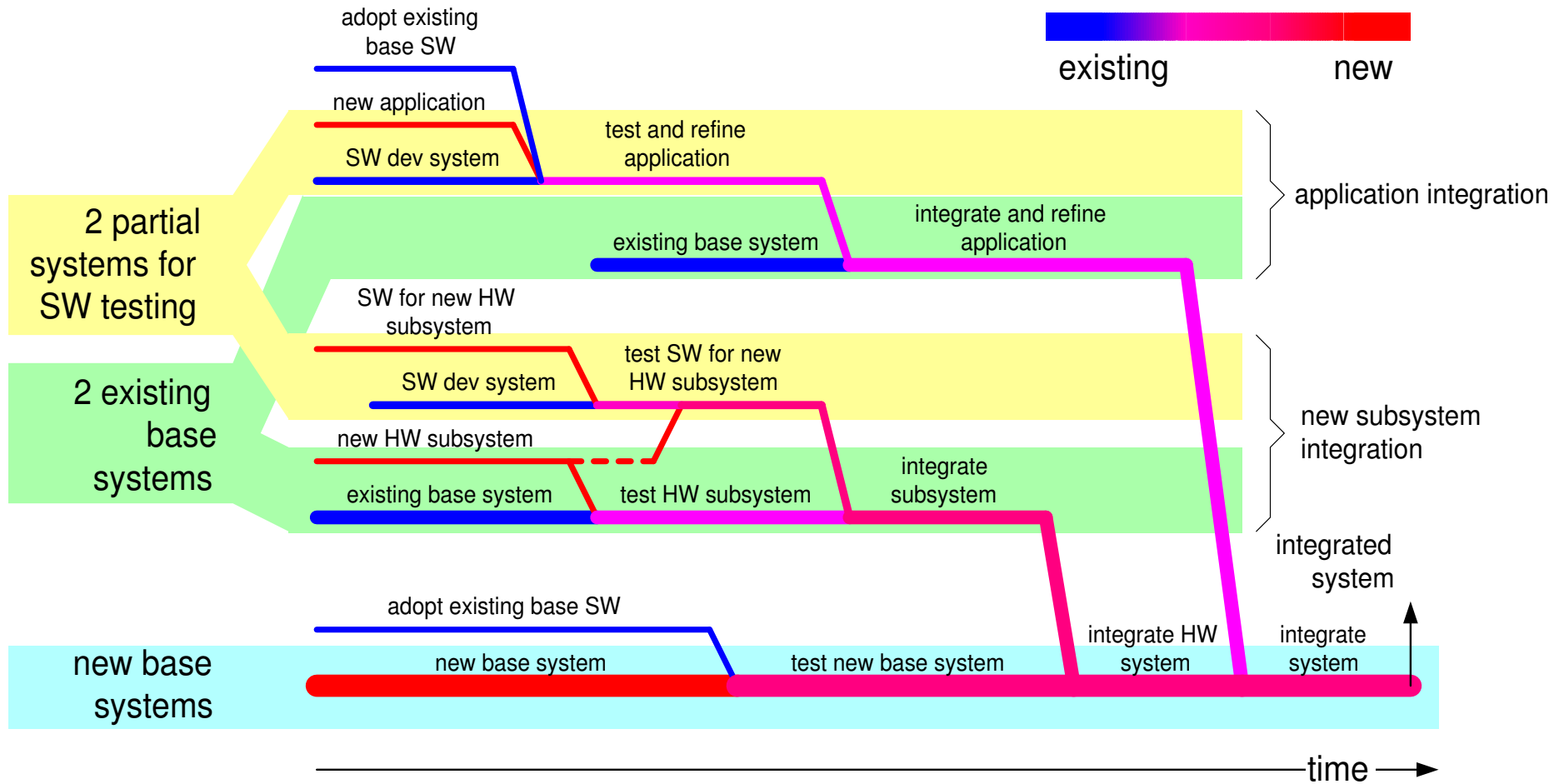
Zooming in on Integration and Tests



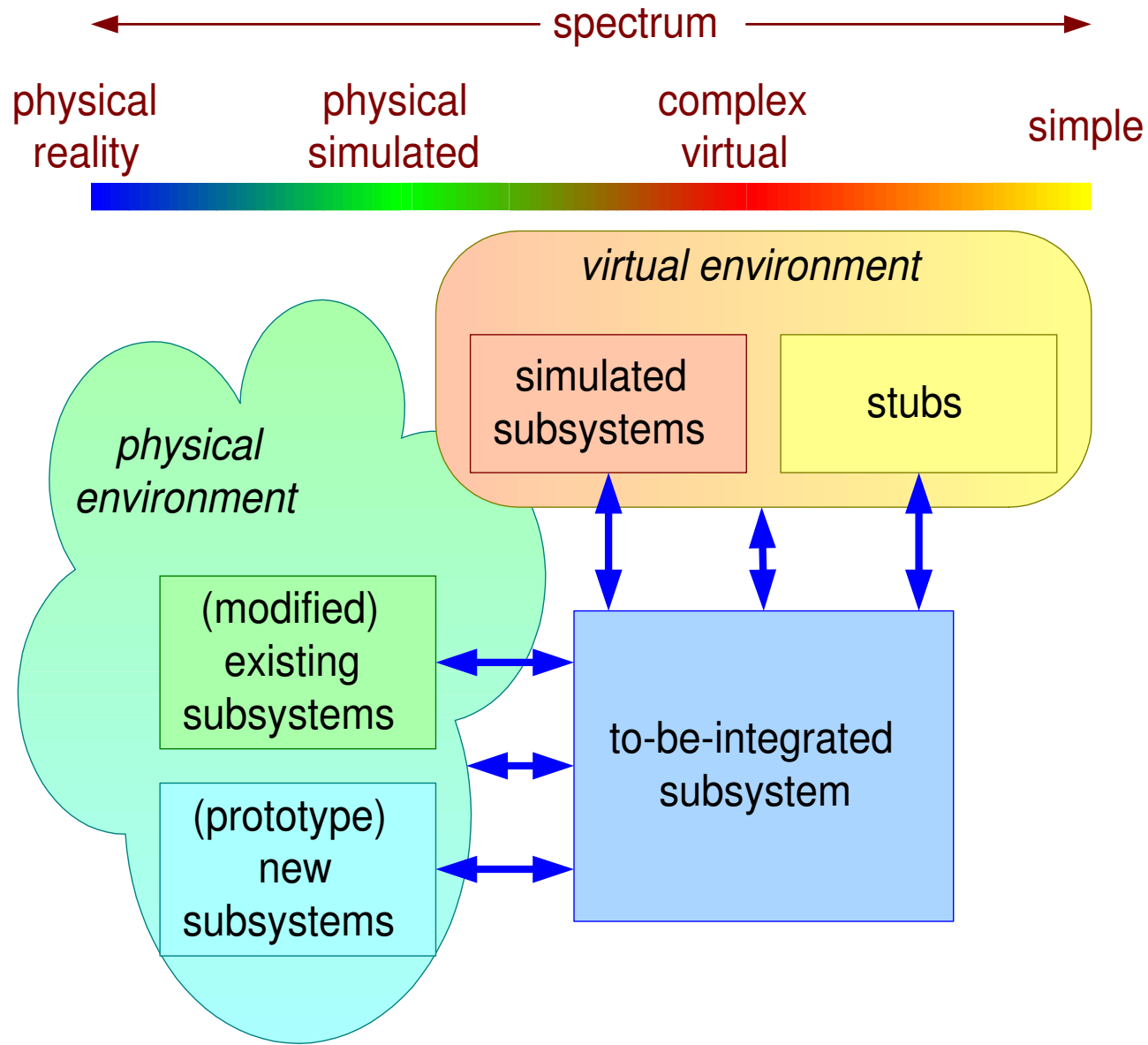
Integration Takes Place in a Bottom-up Fashion



Transition from Previous System to New System



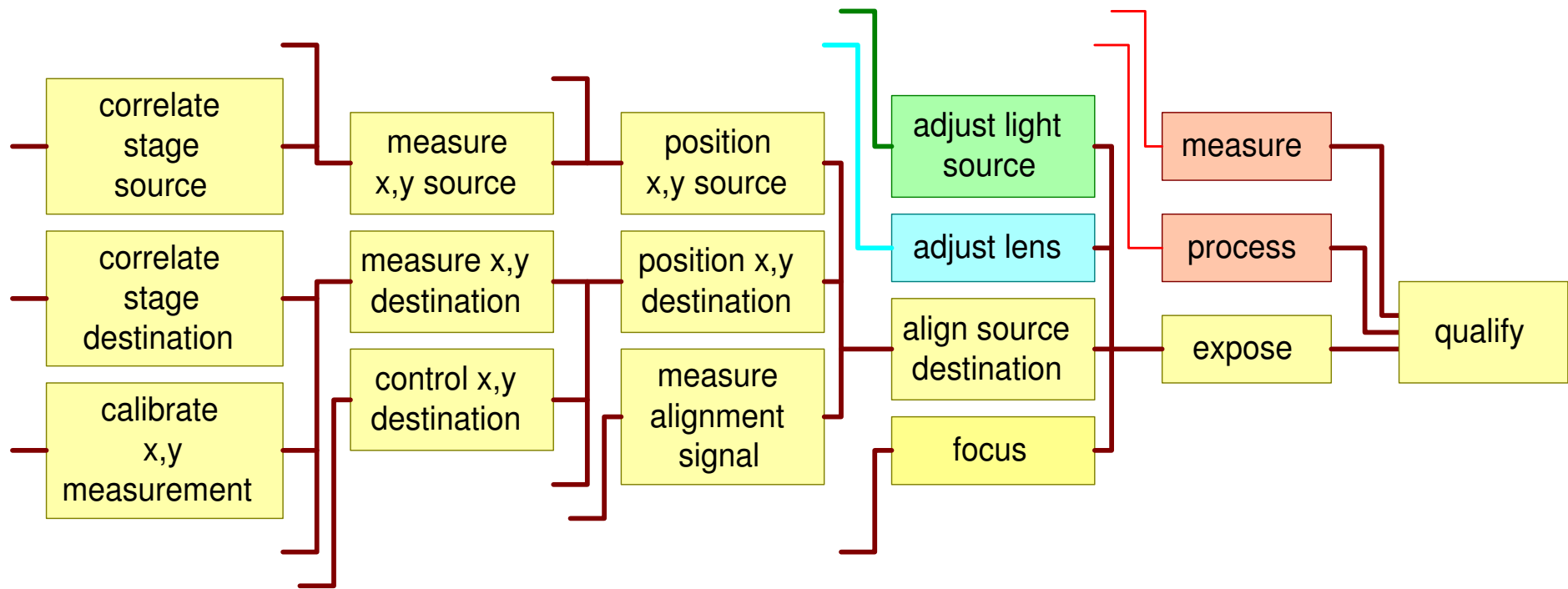
Alternatives to Integrate a Subsystem Early in the Project



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

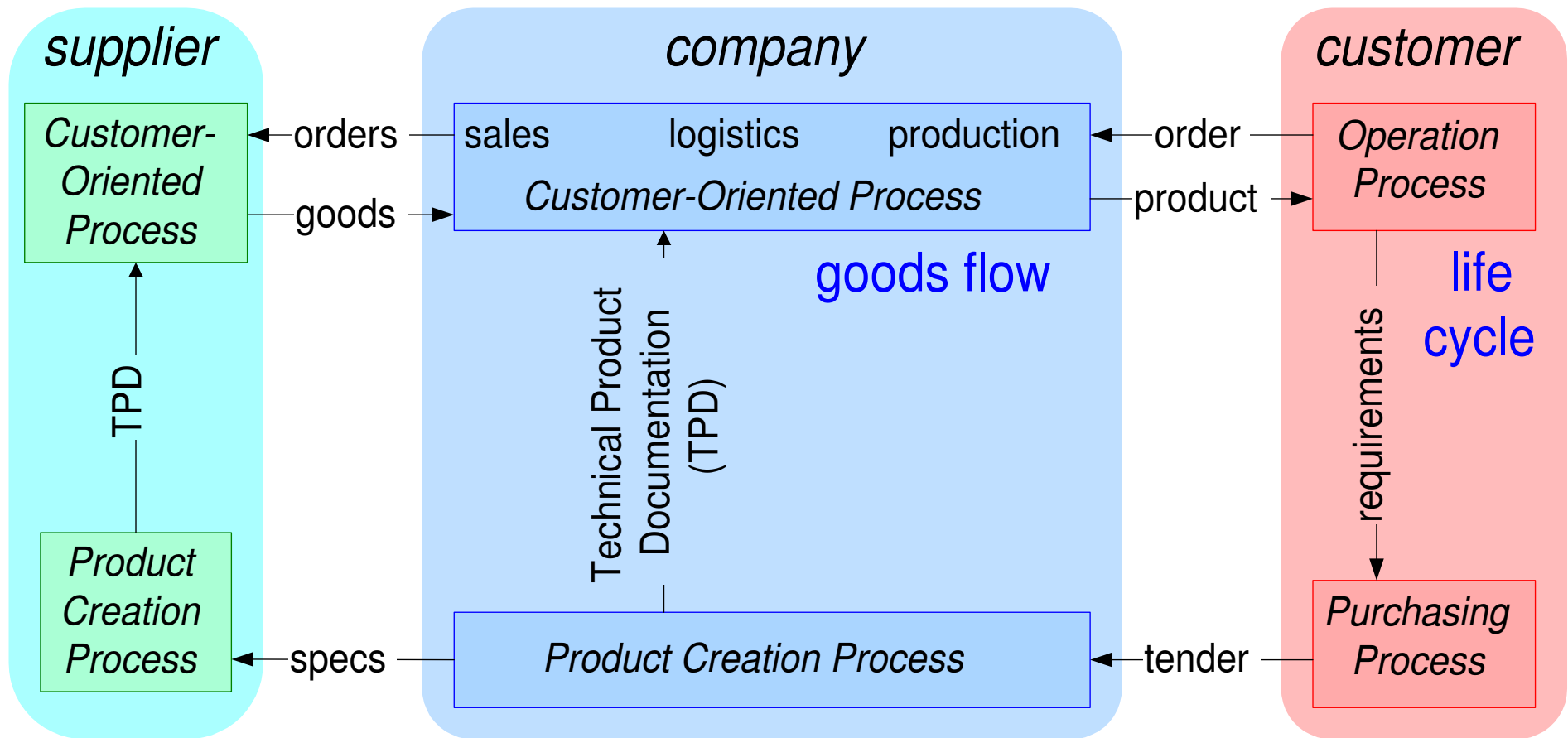
engineers

design
component test
troubleshooting
participate in test

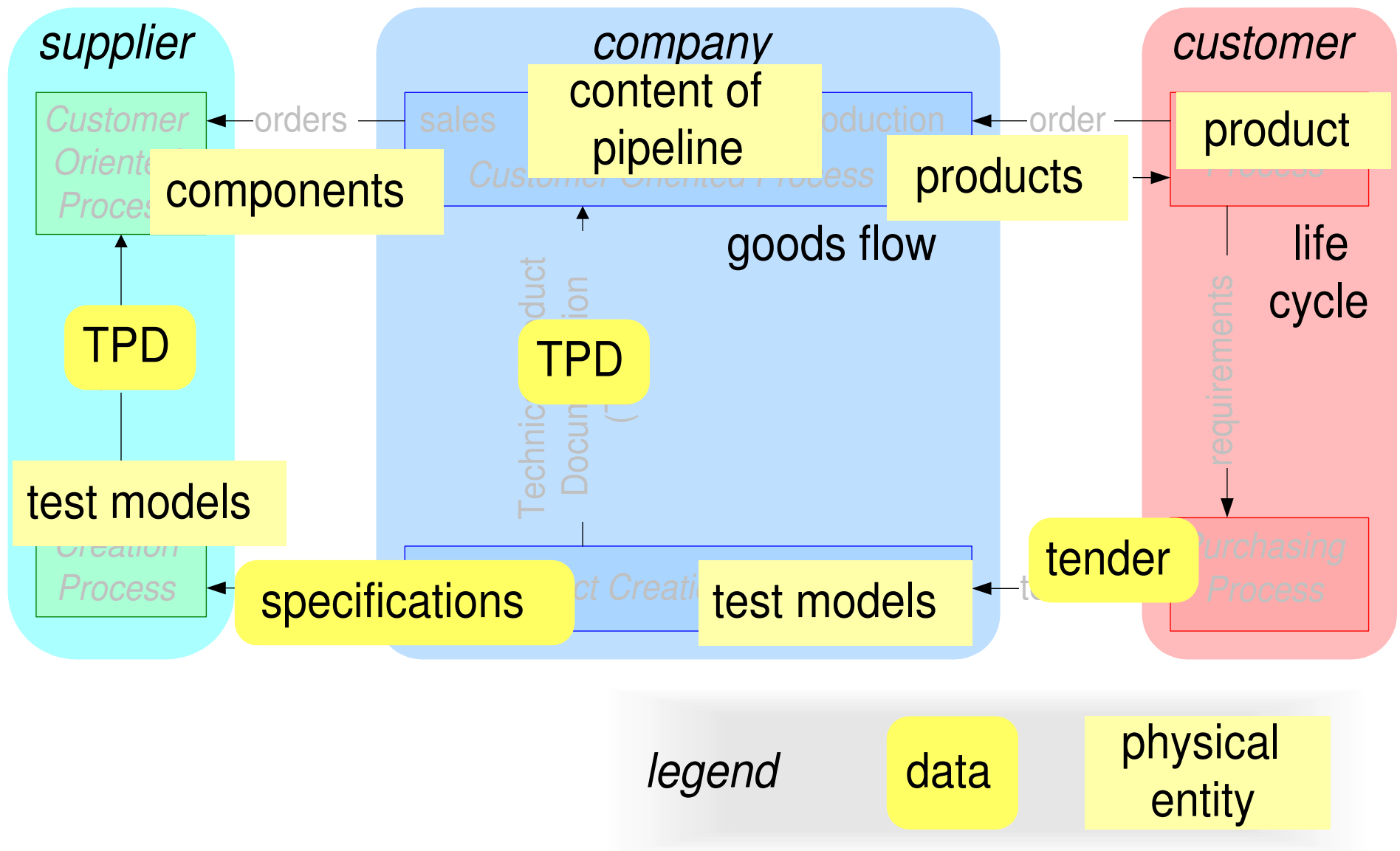
machine owner

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.

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



Fundamentals of Requirements Engineering

by *Gerrit Muller* Buskerud University College

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

`www.gaudisite.nl`

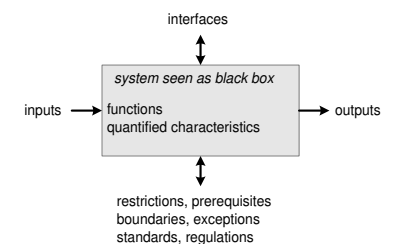
Abstract

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”

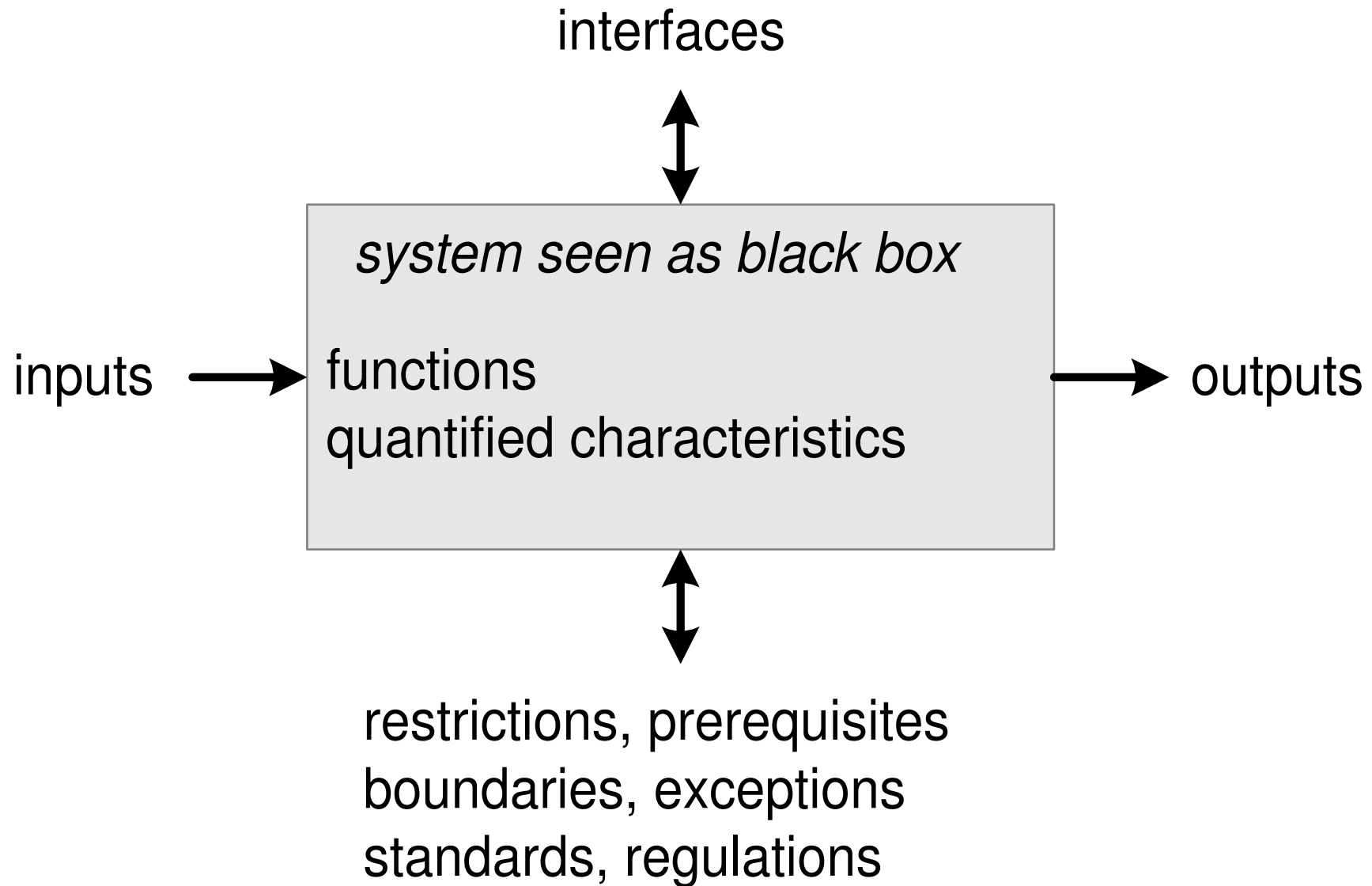
Requirements describing the needs of the customer:
Customer Needs

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

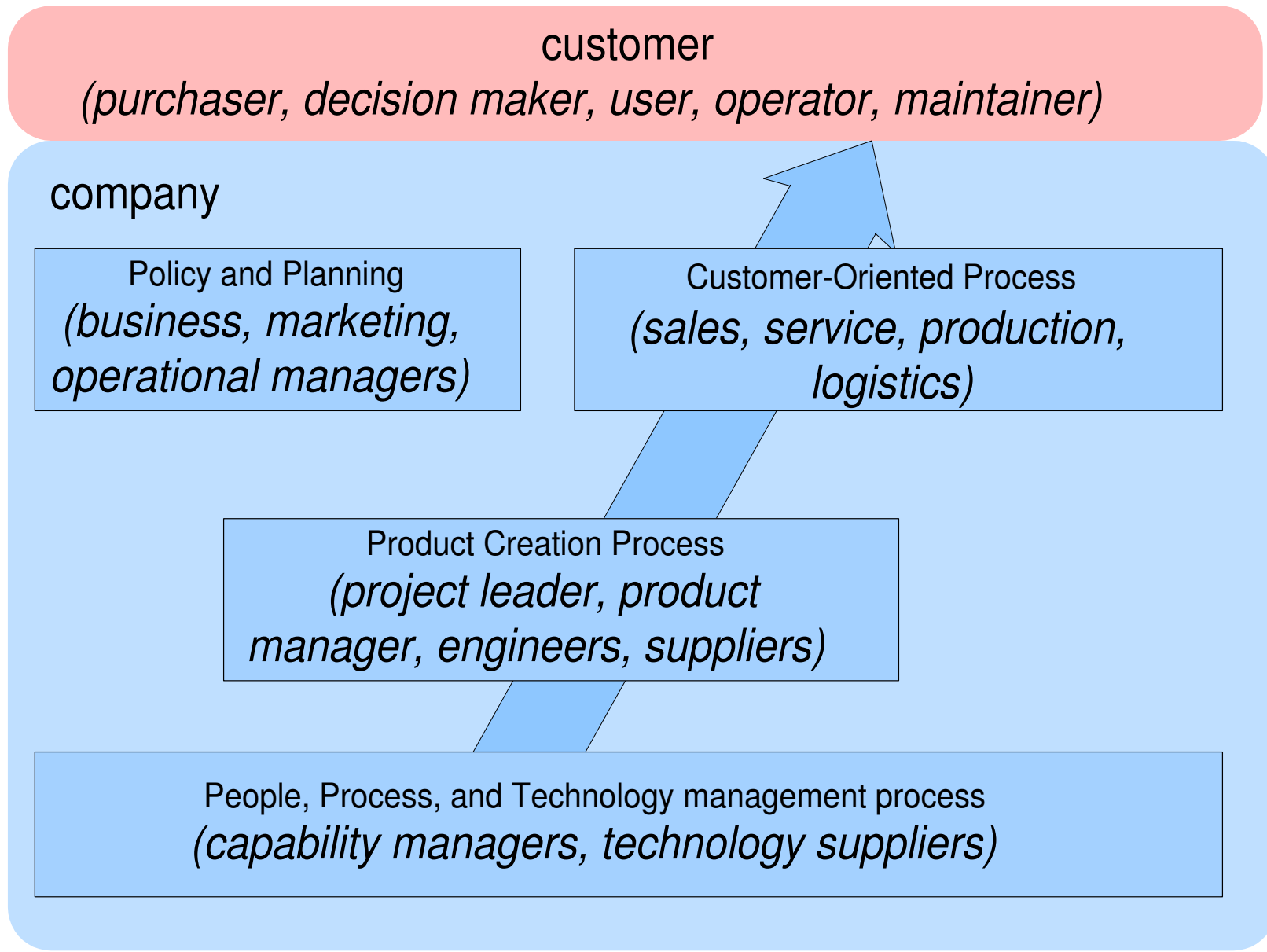
The requirements management process recursively applies definition 2 for every level of decomposition.

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

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`

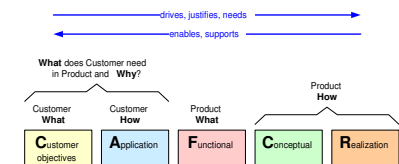
Abstract

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.

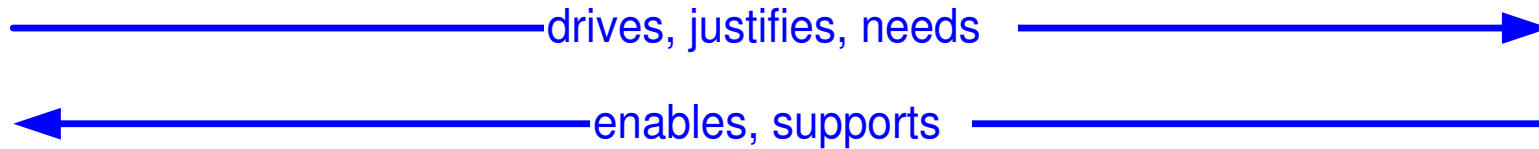
Distribution

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.

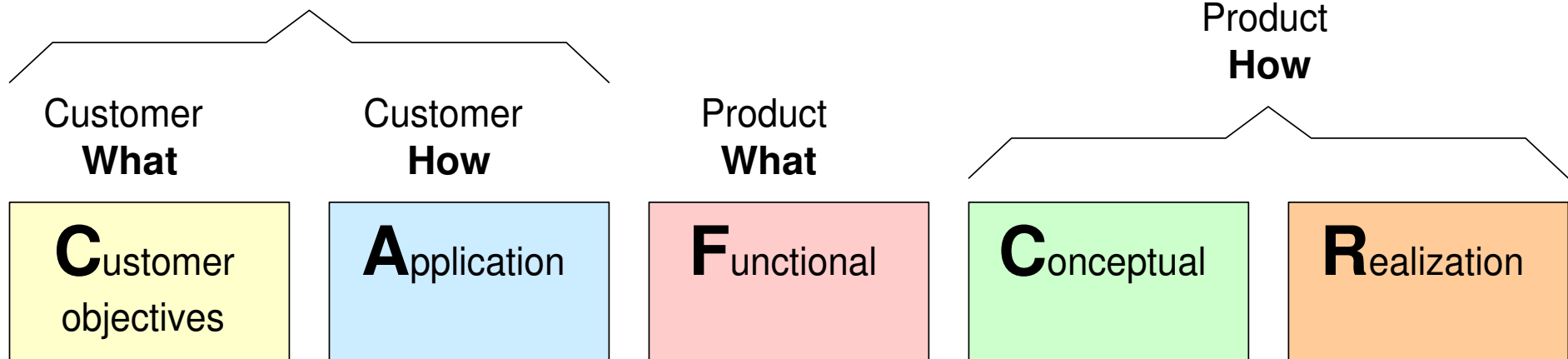
February 11, 2012
status: draft
version: 0.4



The “CAFCR” model

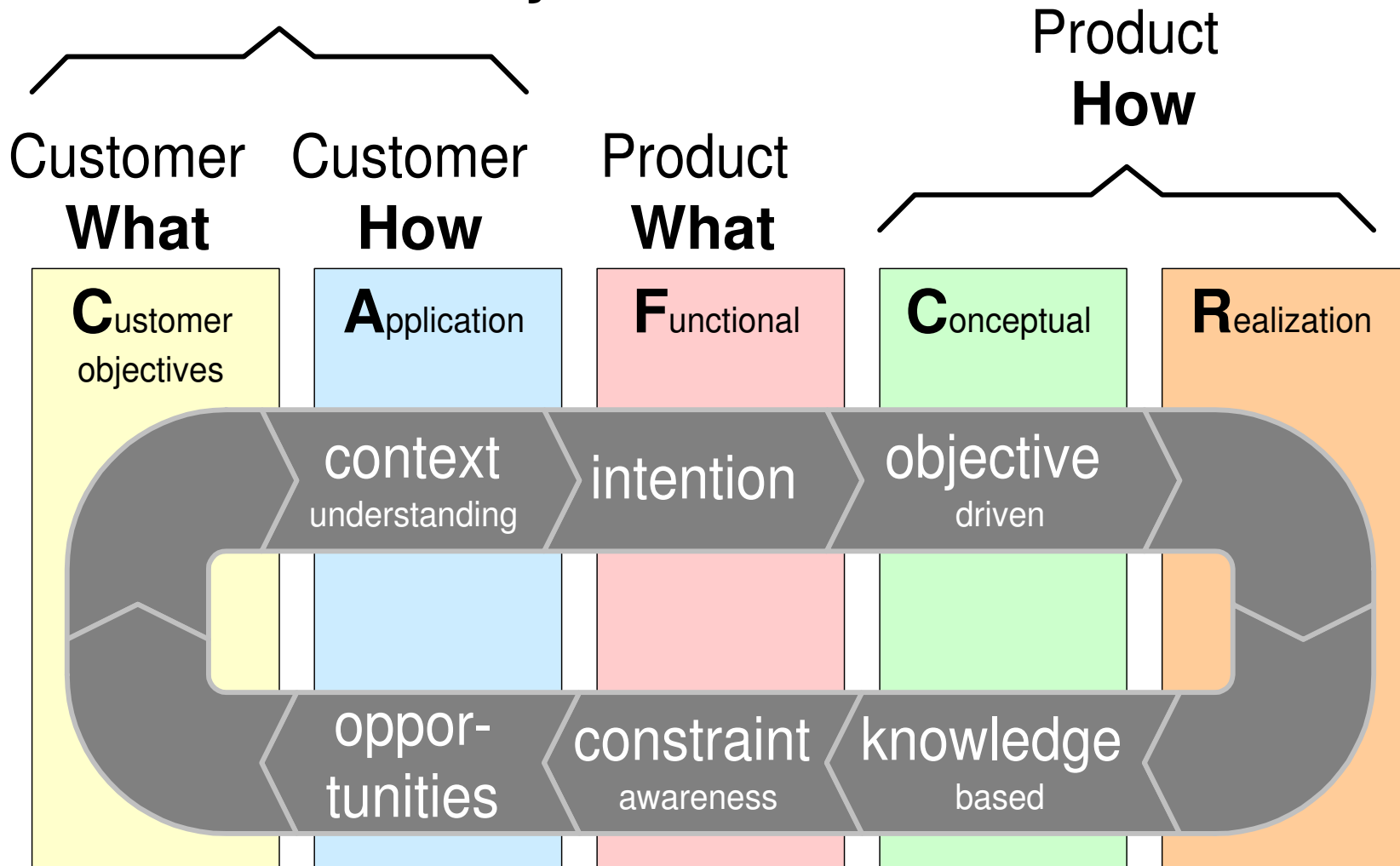


What does Customer need
in Product and **Why?**

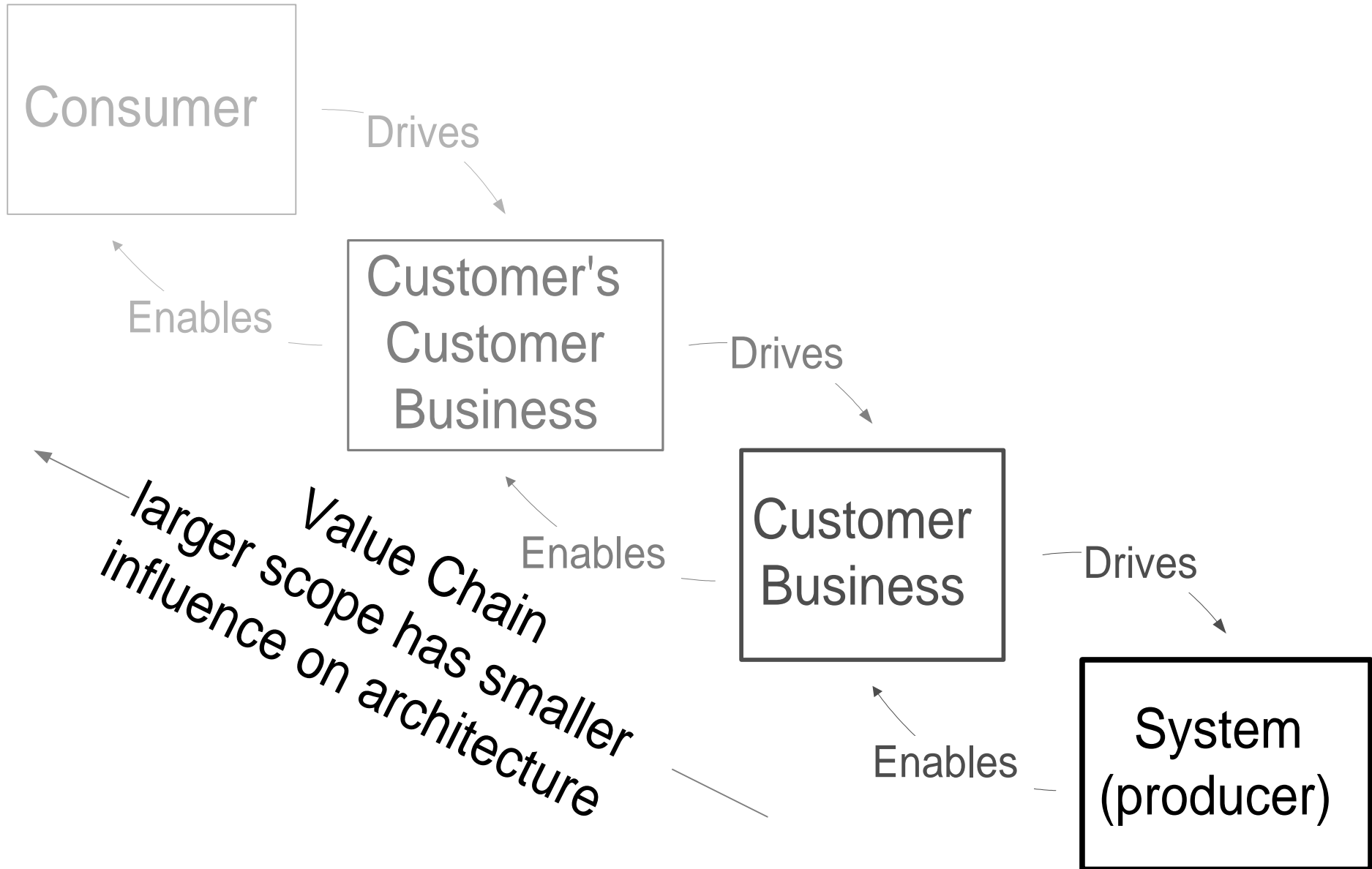


Integrating CAFCR

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



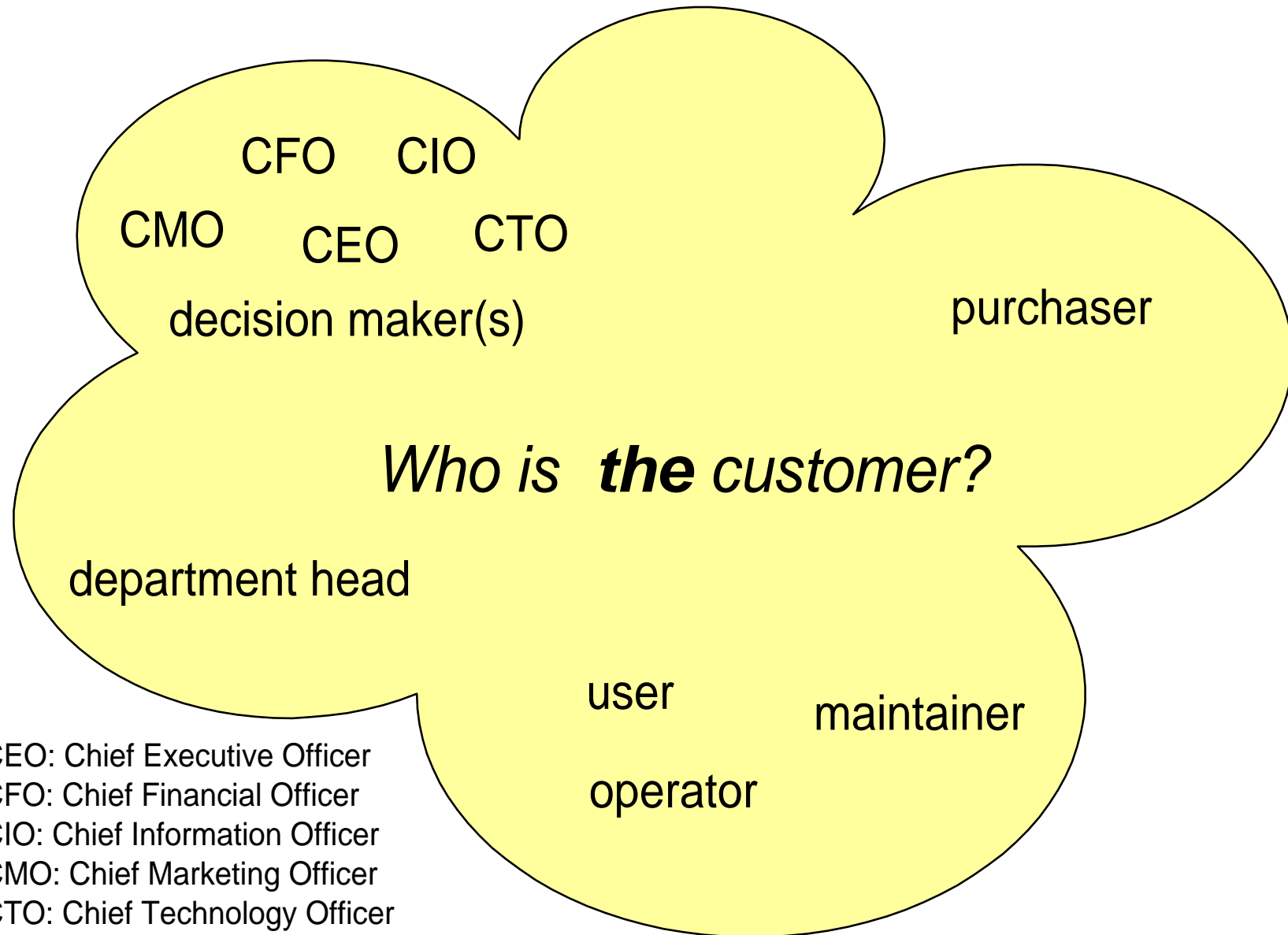
CAFCR can be applied recursively



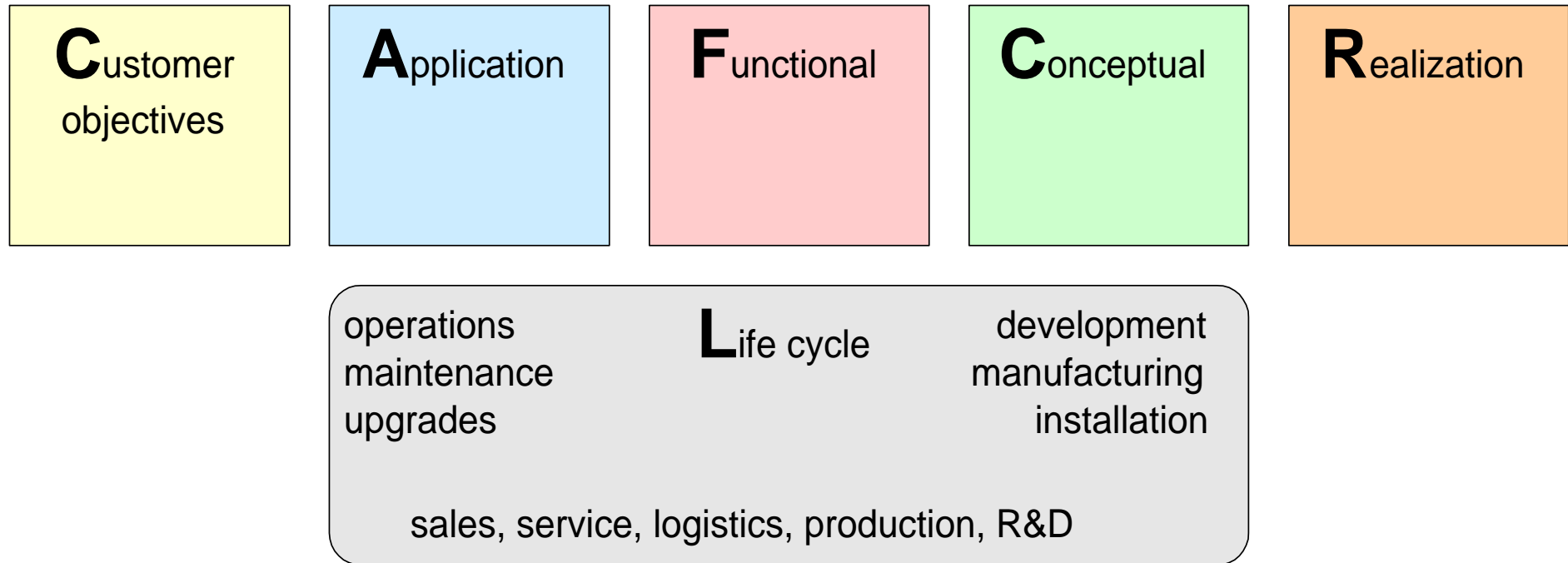
Market segmentation

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`

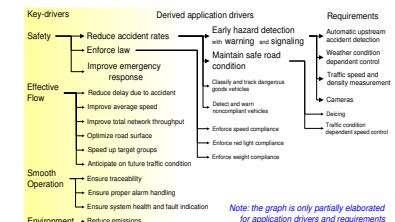
Abstract

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

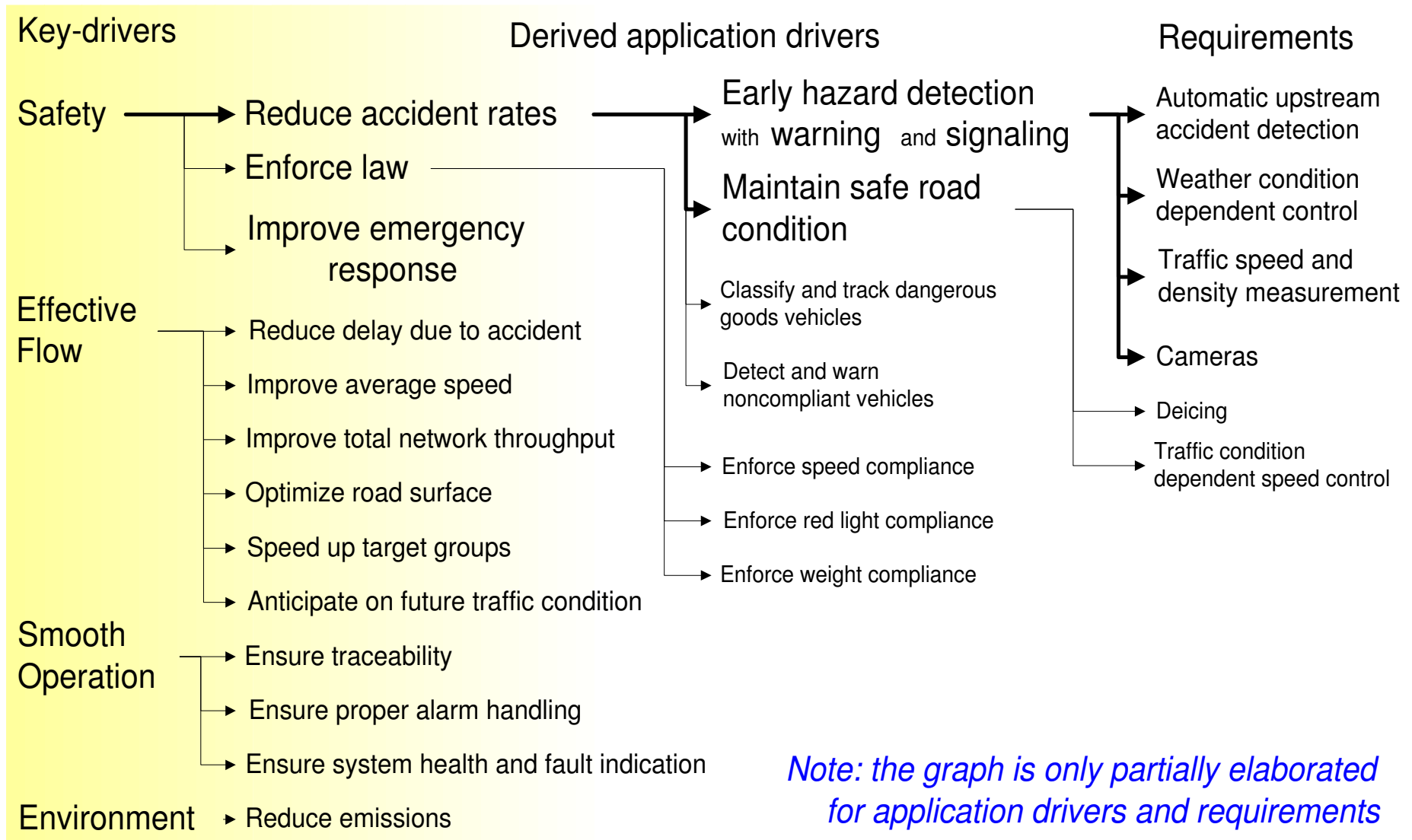
Distribution

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



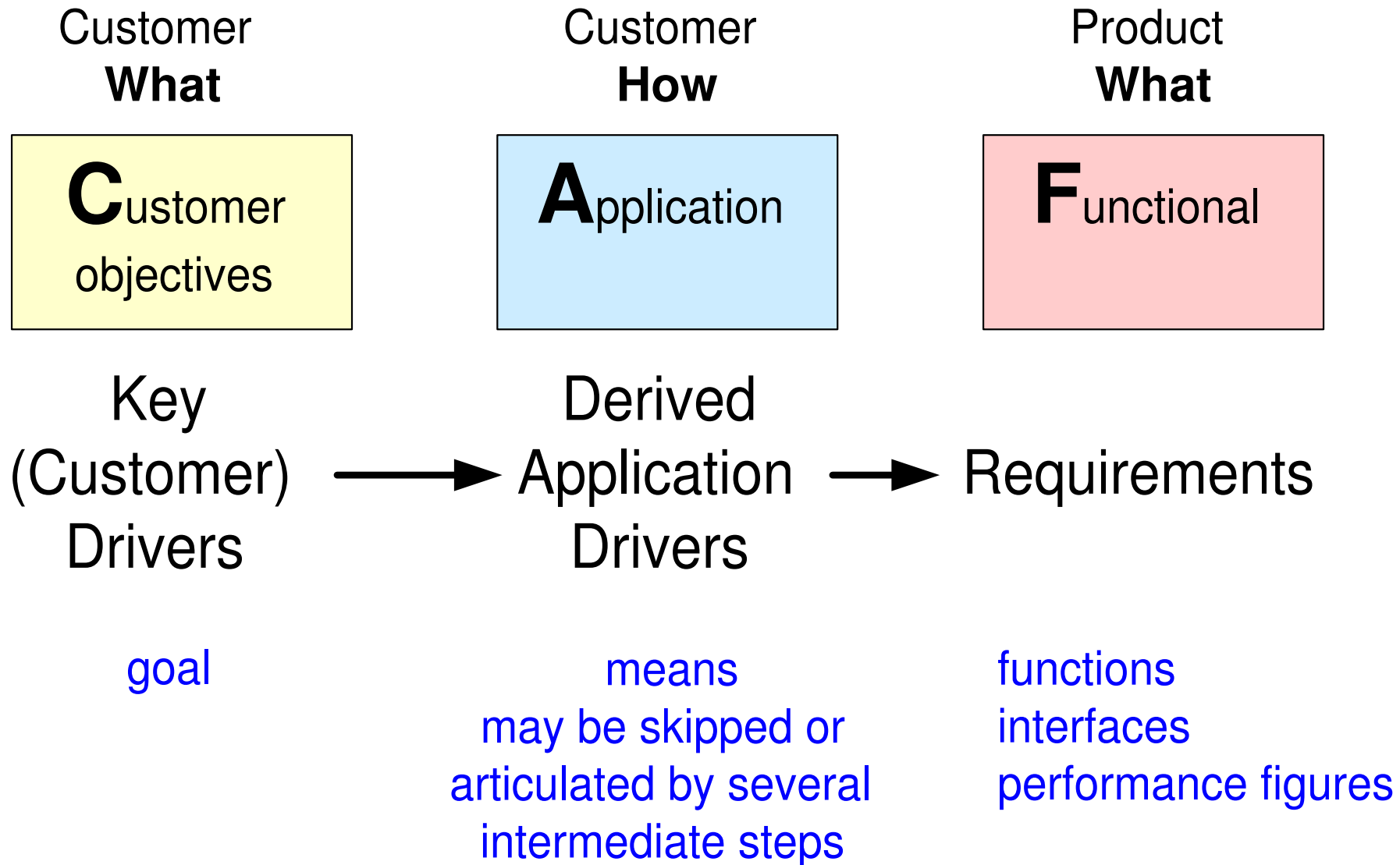
Method to create Key Driver Graph

- | | |
|--|--|
| • Define the scope specific. | in terms of stakeholder or market segments |
| • Acquire and analyze facts | extract facts from the product specification
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* Buskerud University College

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

`www.gaudisite.nl`

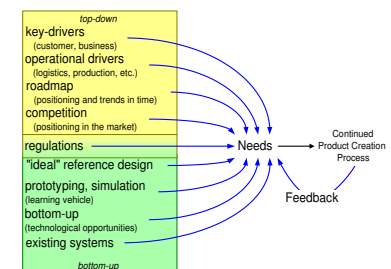
Abstract

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.

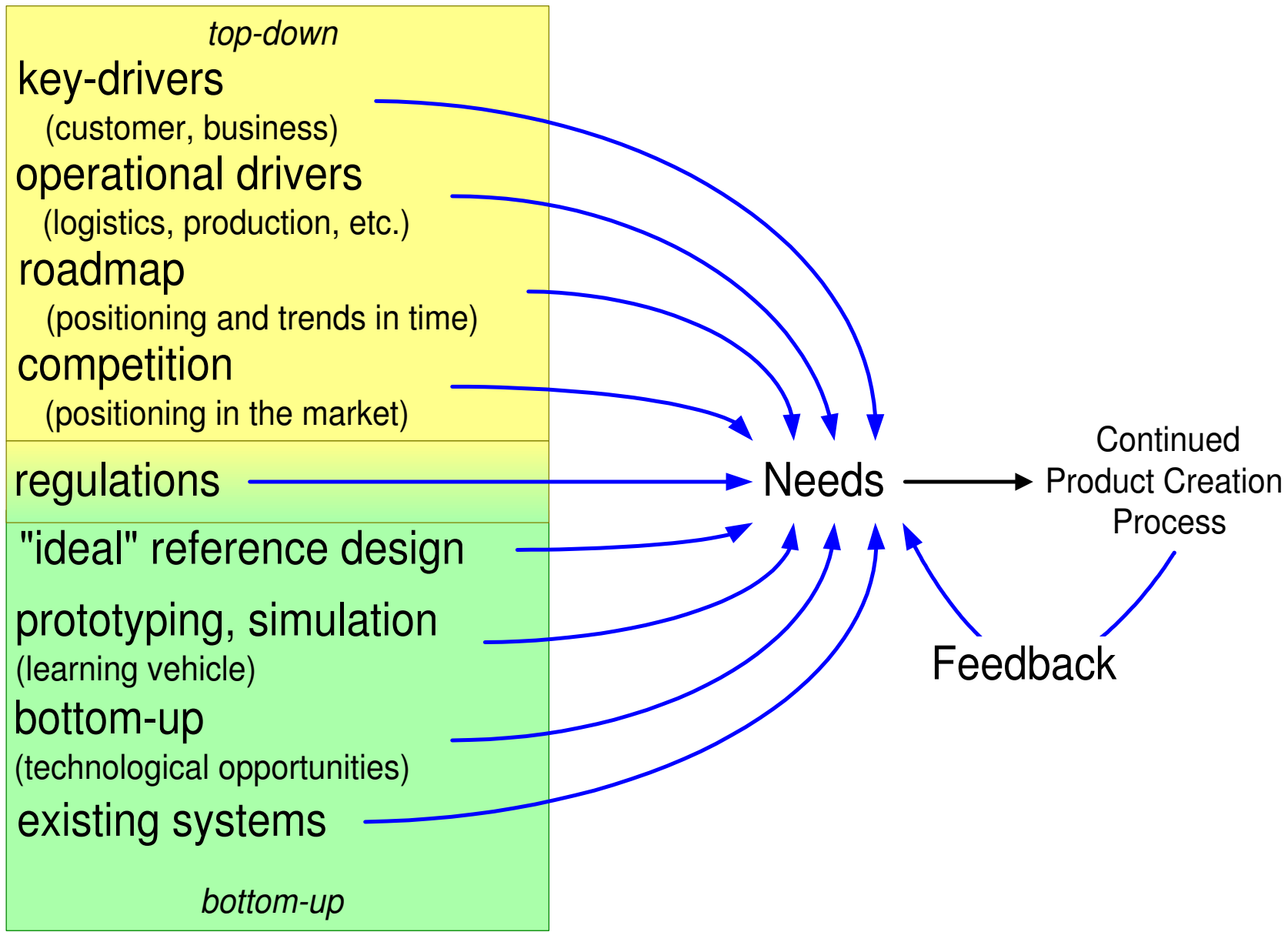
Distribution

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.

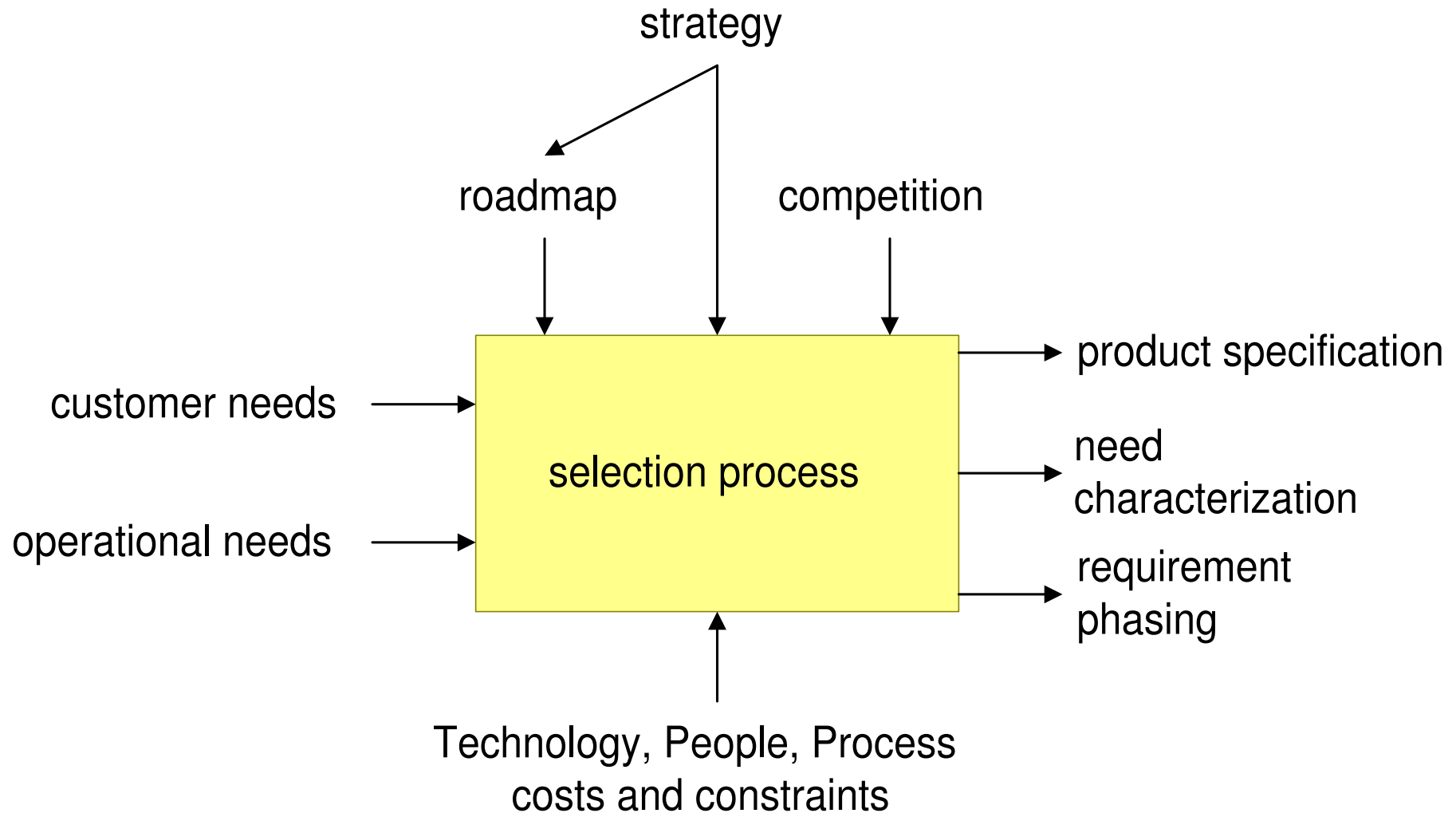
February 11, 2012
status: draft
version: 0



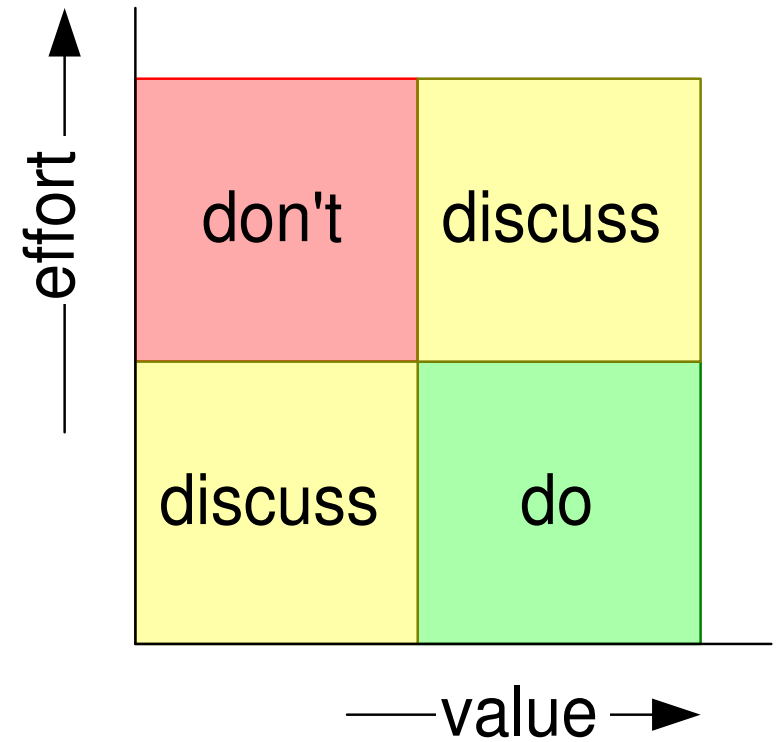
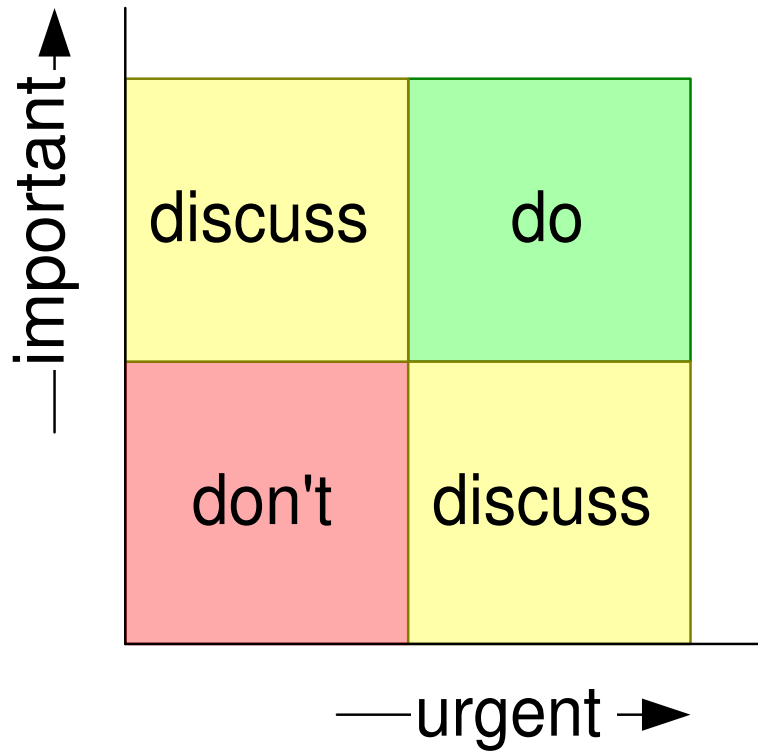
Complementary Viewpoints to Capture Requirements



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.

Module Roadmapping

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

Abstract

This module addresses roadmapping.

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: 1.1



Embedded Systems
INSTITUTE

Roadmapping

by *Gerrit Muller* Buskerud University College

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

`www.gaudisite.nl`

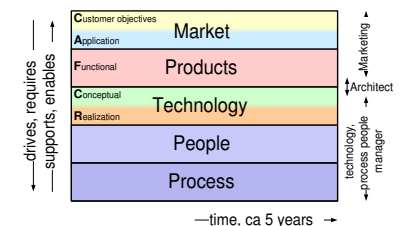
Abstract

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.

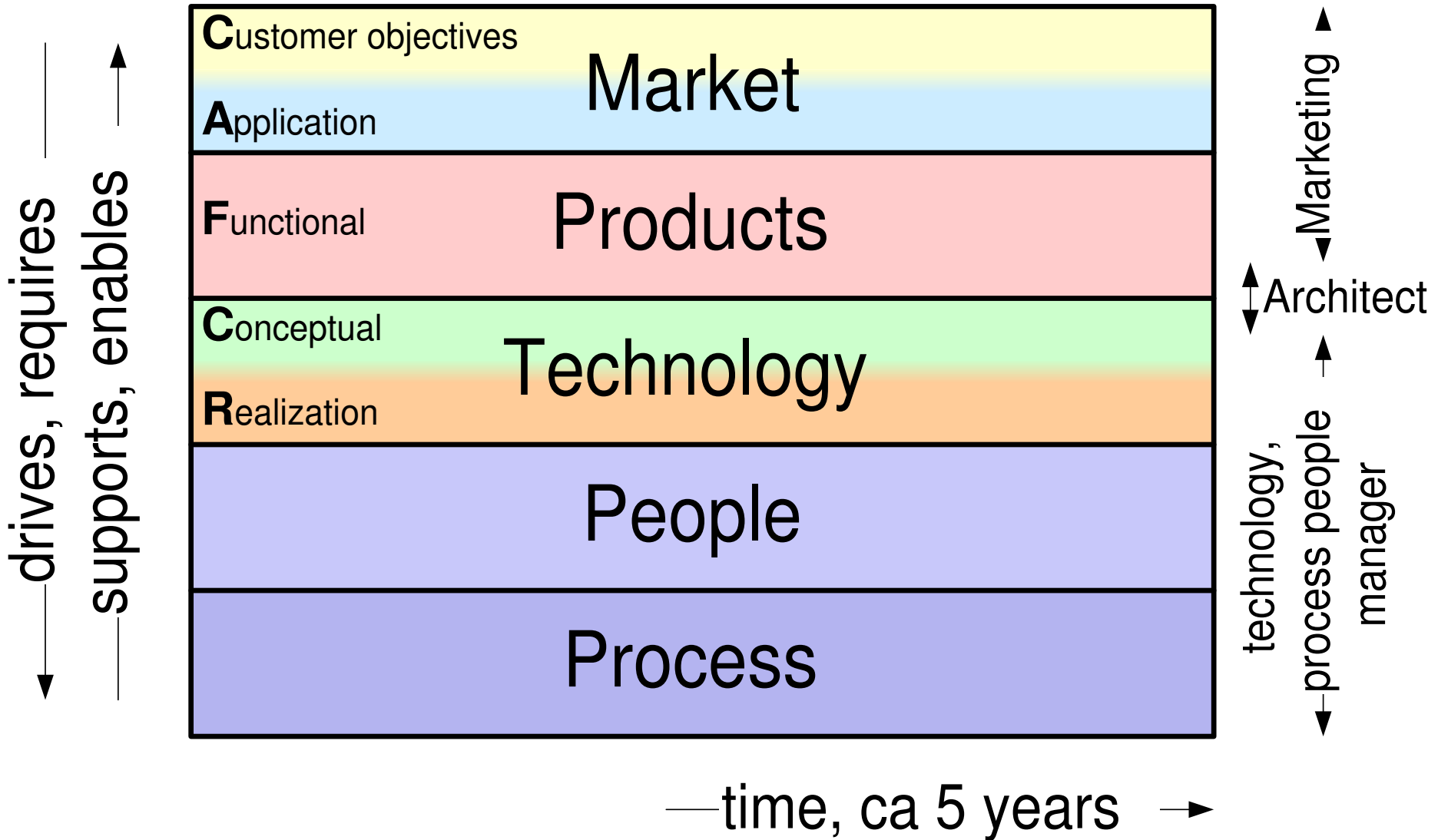
Distribution

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.

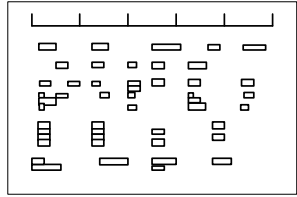
February 11, 2012
status: concept
version: 2.0



The Roadmap Integrates Five Views



Granularity of Roadmap Material

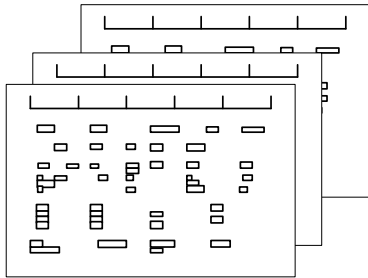


Top-level
roadmap

Single page

Poster

part of many presentations

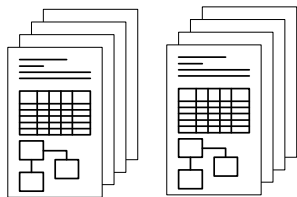


Supporting
roadmaps

Single page
per view
or per driver

Poster

part of many presentations



Supporting
reports

Document
per relevant
subject

Problems that Occur without Roadmapping

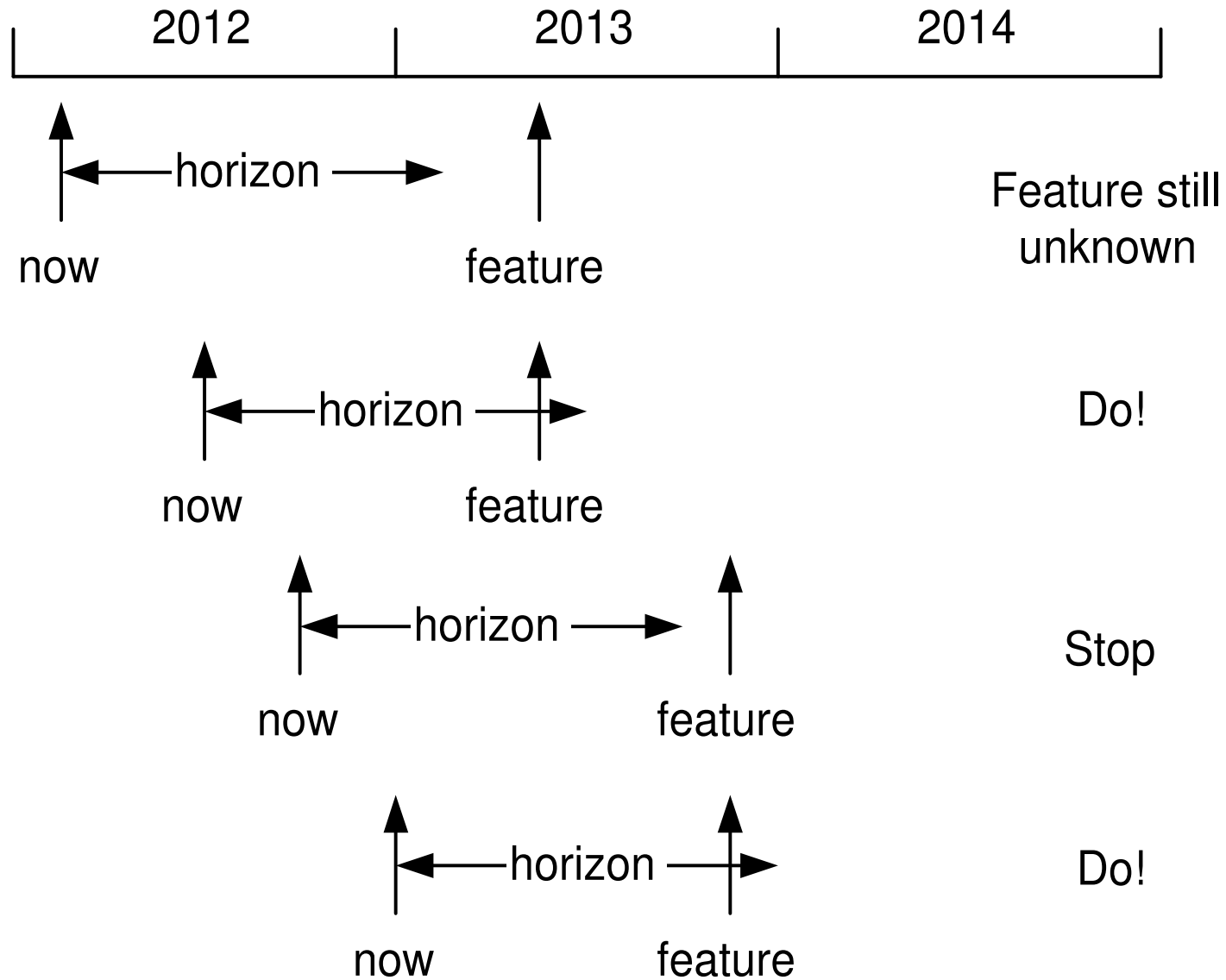
Frequent changes in product policy

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

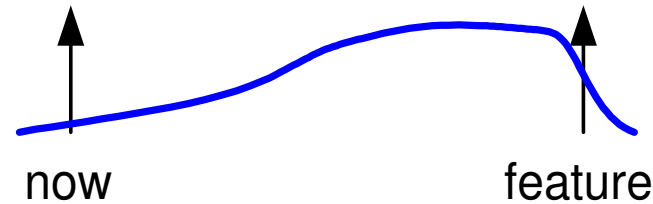
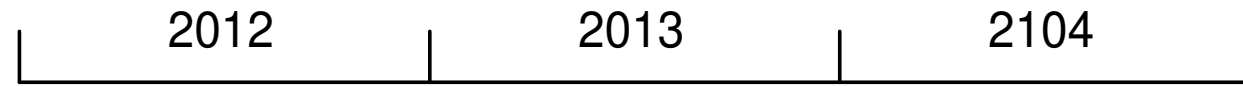
Diverging activities of teams

Missed market opportunities

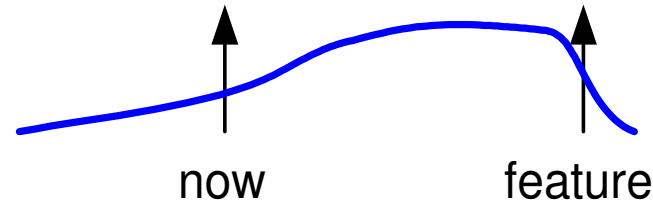
Management with a Limited Horizon



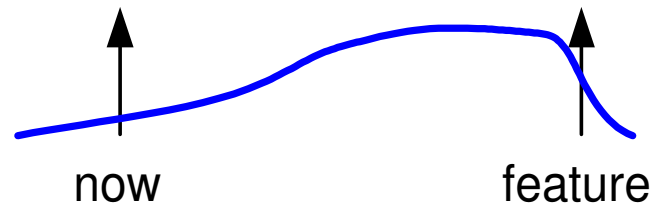
Management with a Broader Time Perspective



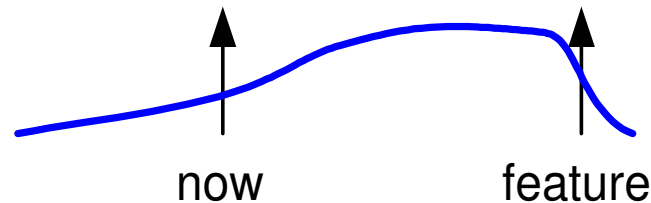
Preparation by
0.5 person



Work with
1.5 persons

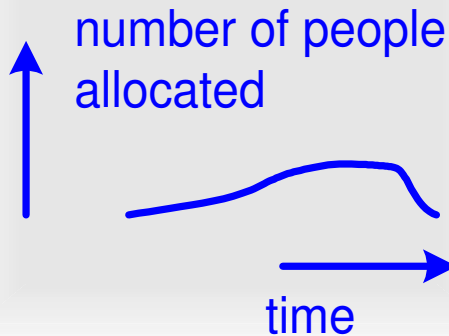


Continue with
0.5 person

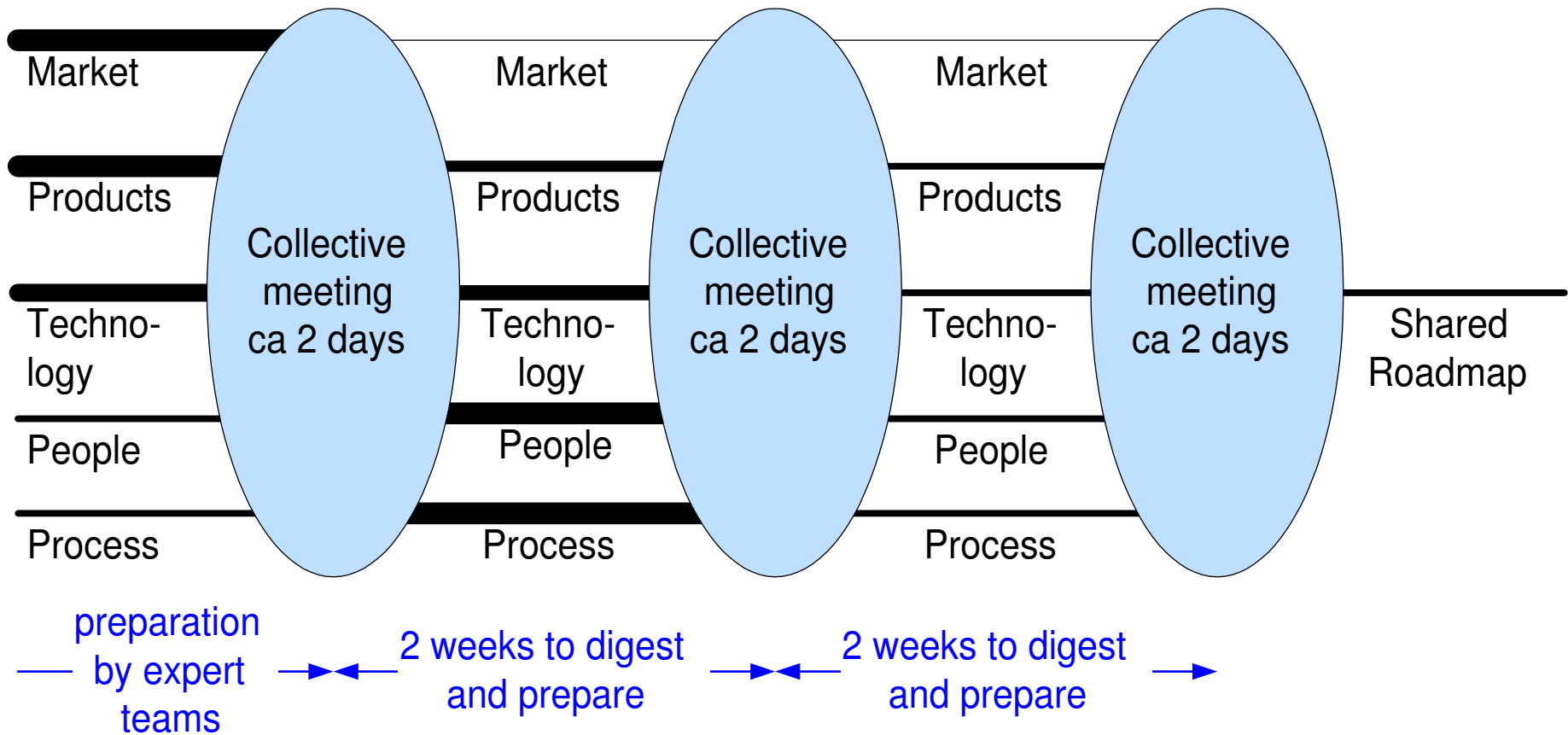


Work with
1.5 persons

legend



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

Market: What is needed by the customers?

Products: How to package technologies into products to fulfill market needs?

Technology: What technological trends are relevant? What technologies are needed?

People: What kind of and how many people are required to realize the products and technologies?

Process: What processes are required to let these people realize the products and technologies?

————time————→

3-Tier Approach

	<i>horizon</i>	<i>update</i>	<i>scope</i>	<i>type</i>
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

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* Embedded Systems Institute

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

`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, adolescence, 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 characteristics 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.

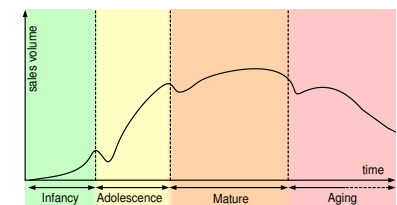
Distribution

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.

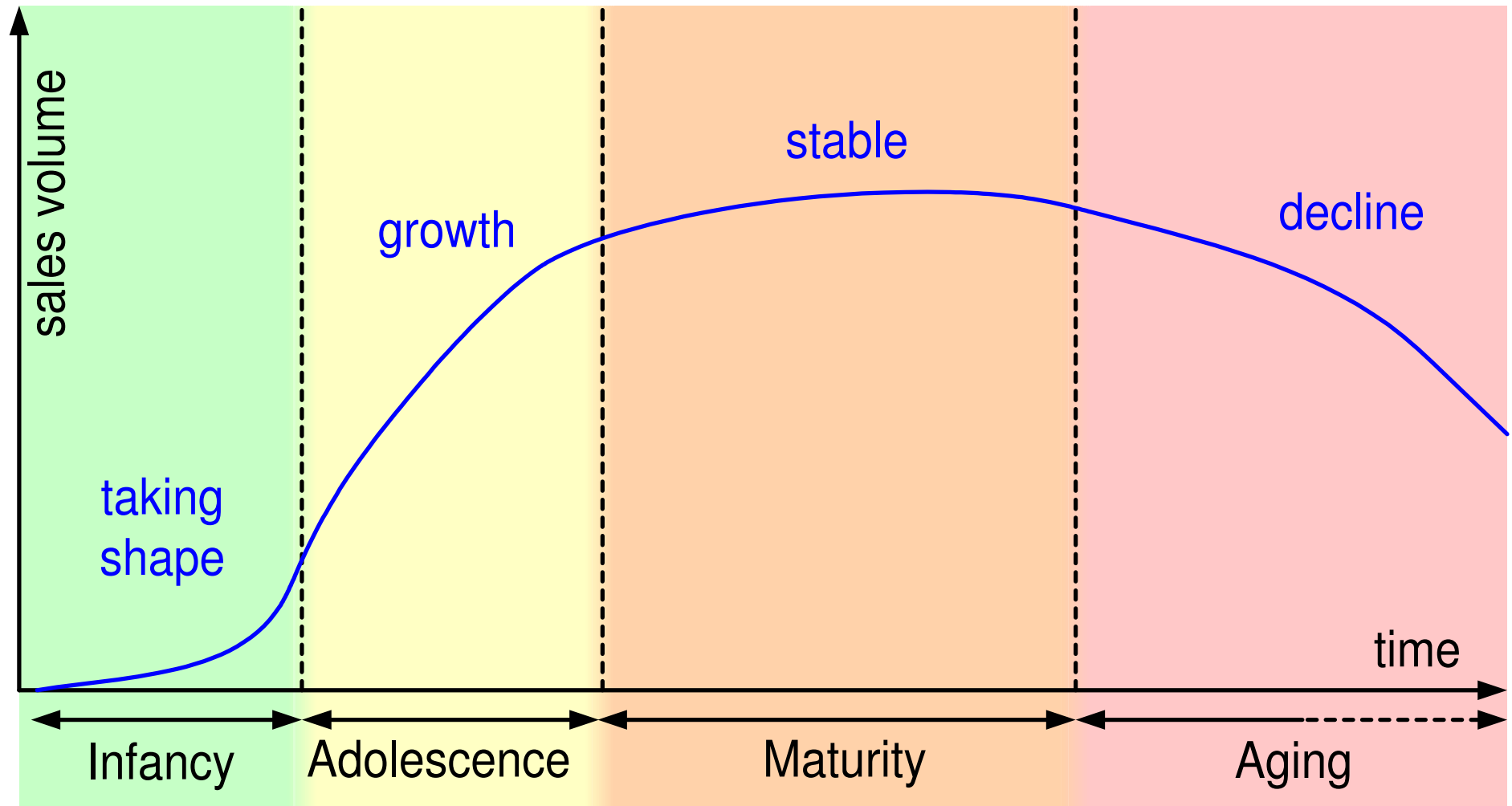
February 11, 2012

status: concept

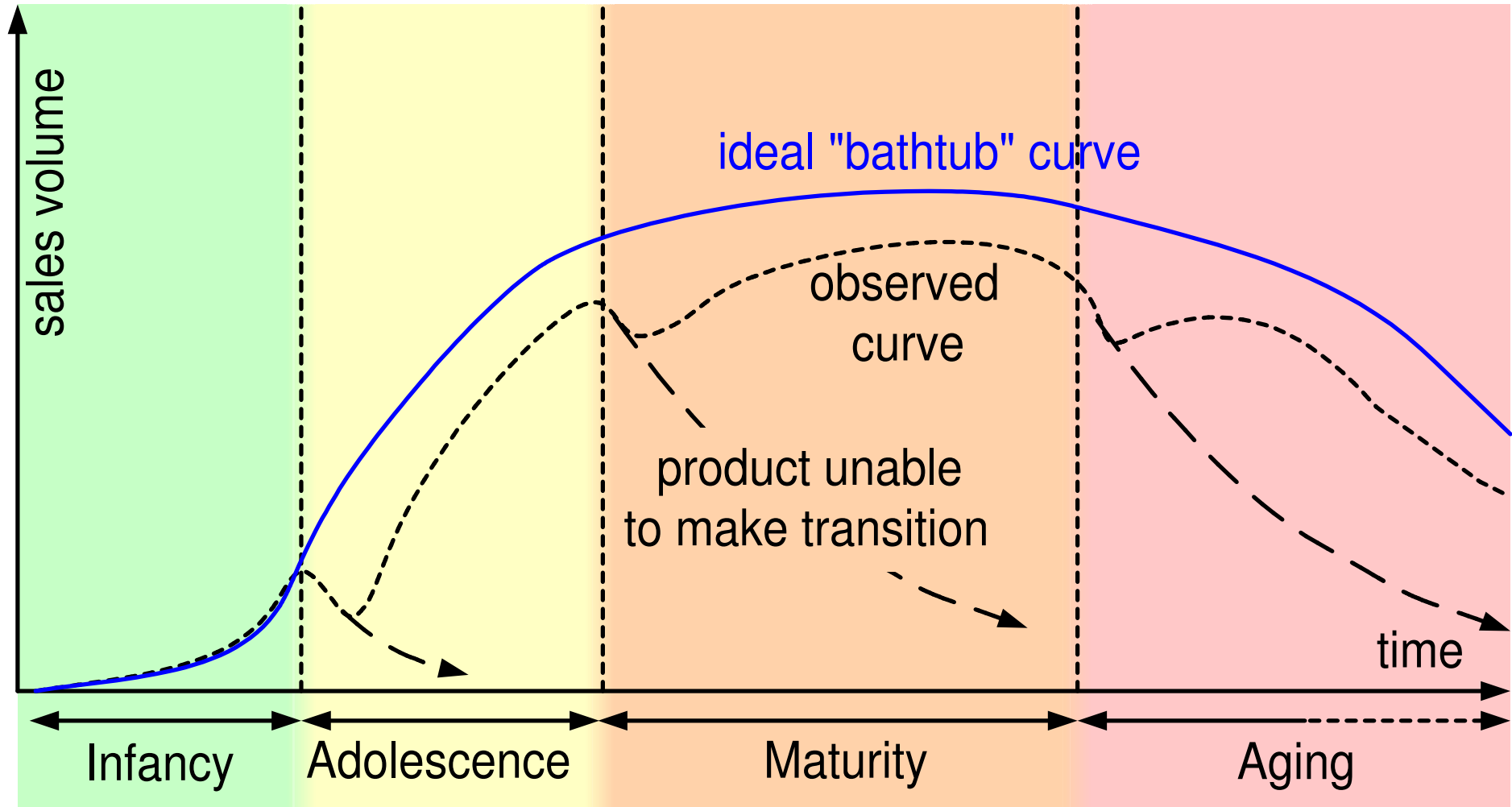
version: 1.2



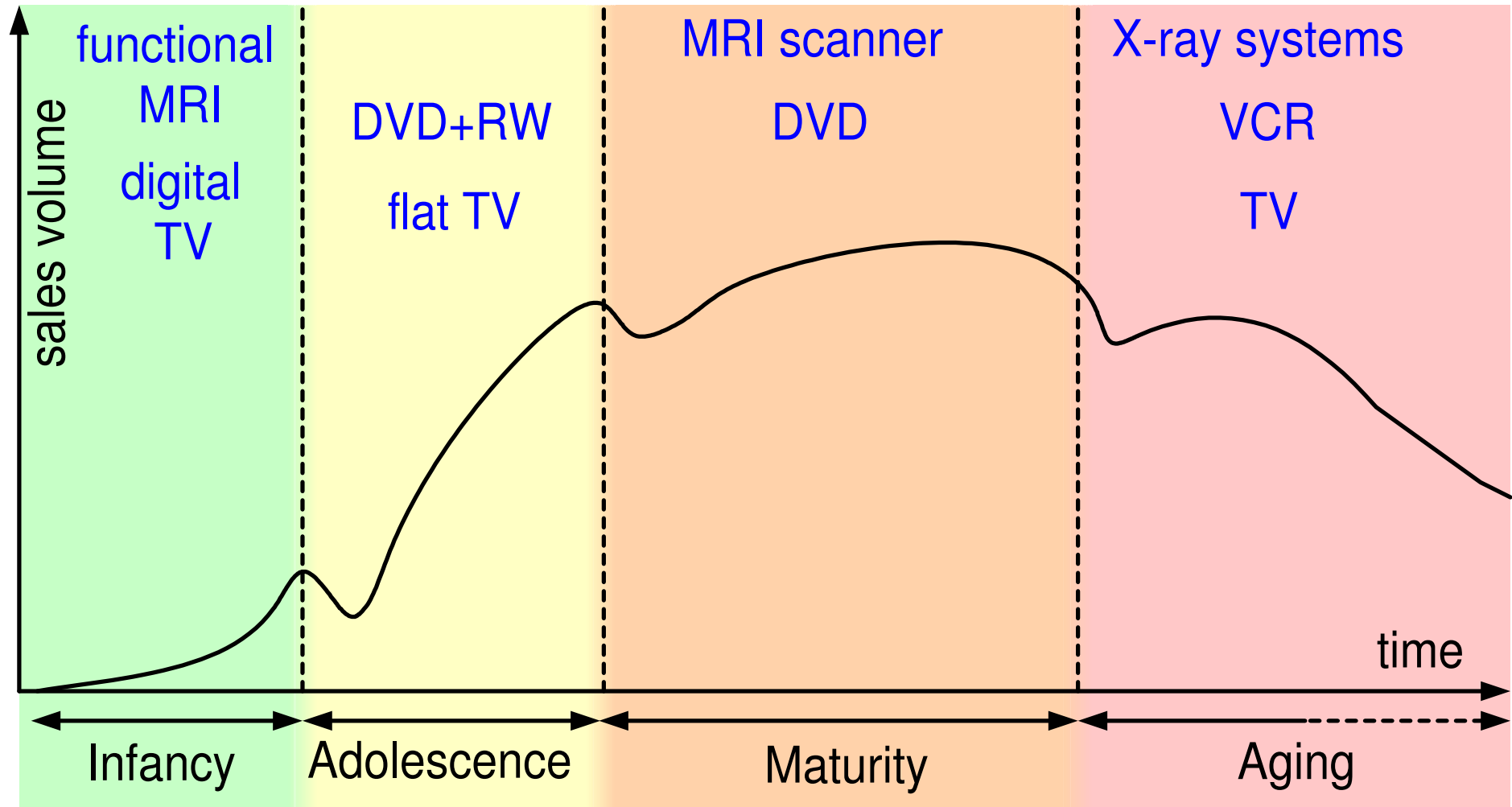
Ideal Bathtub Curve



Market Product Life Cycle Phases in Practice



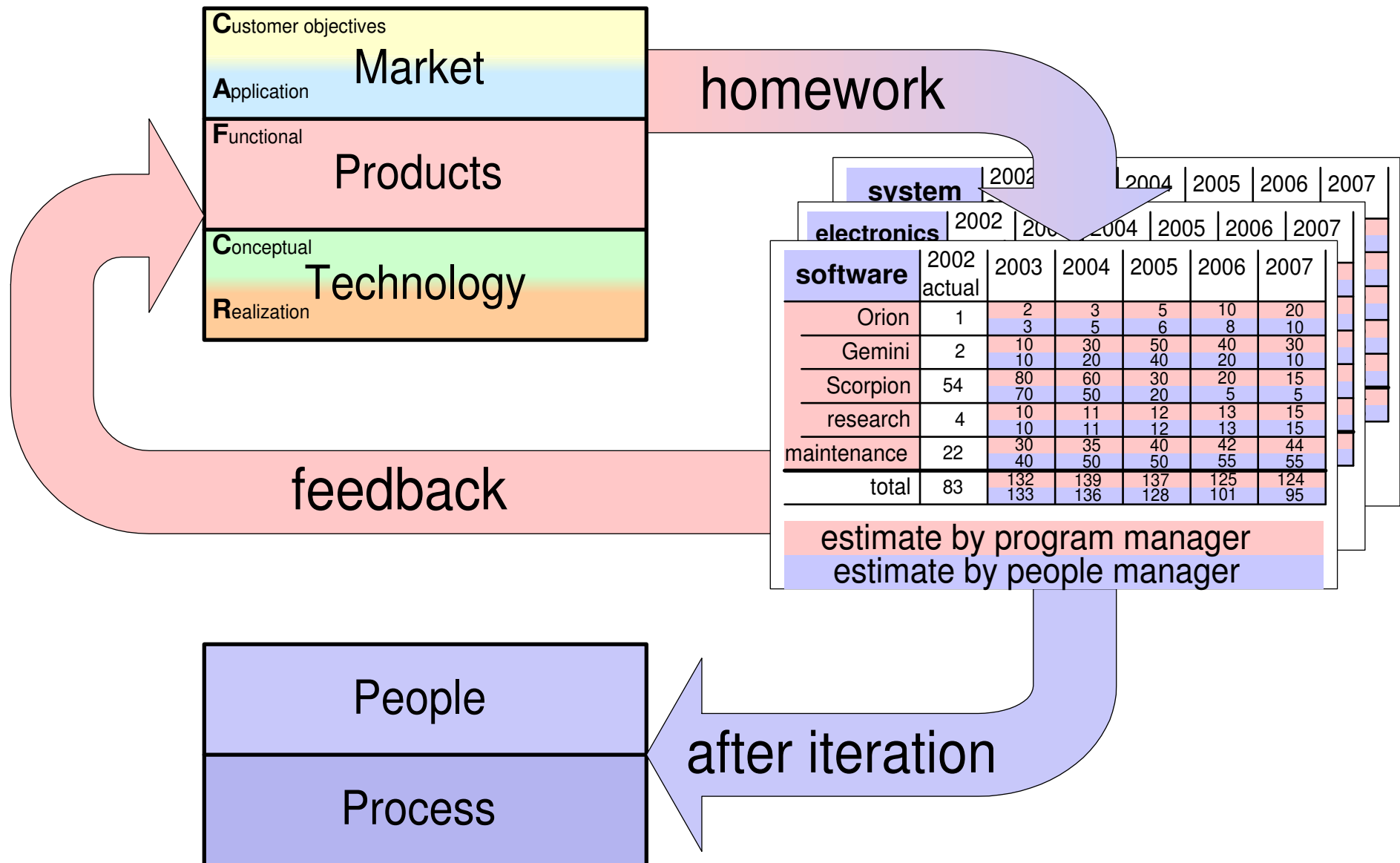
Examples of Product Classes on the Curve



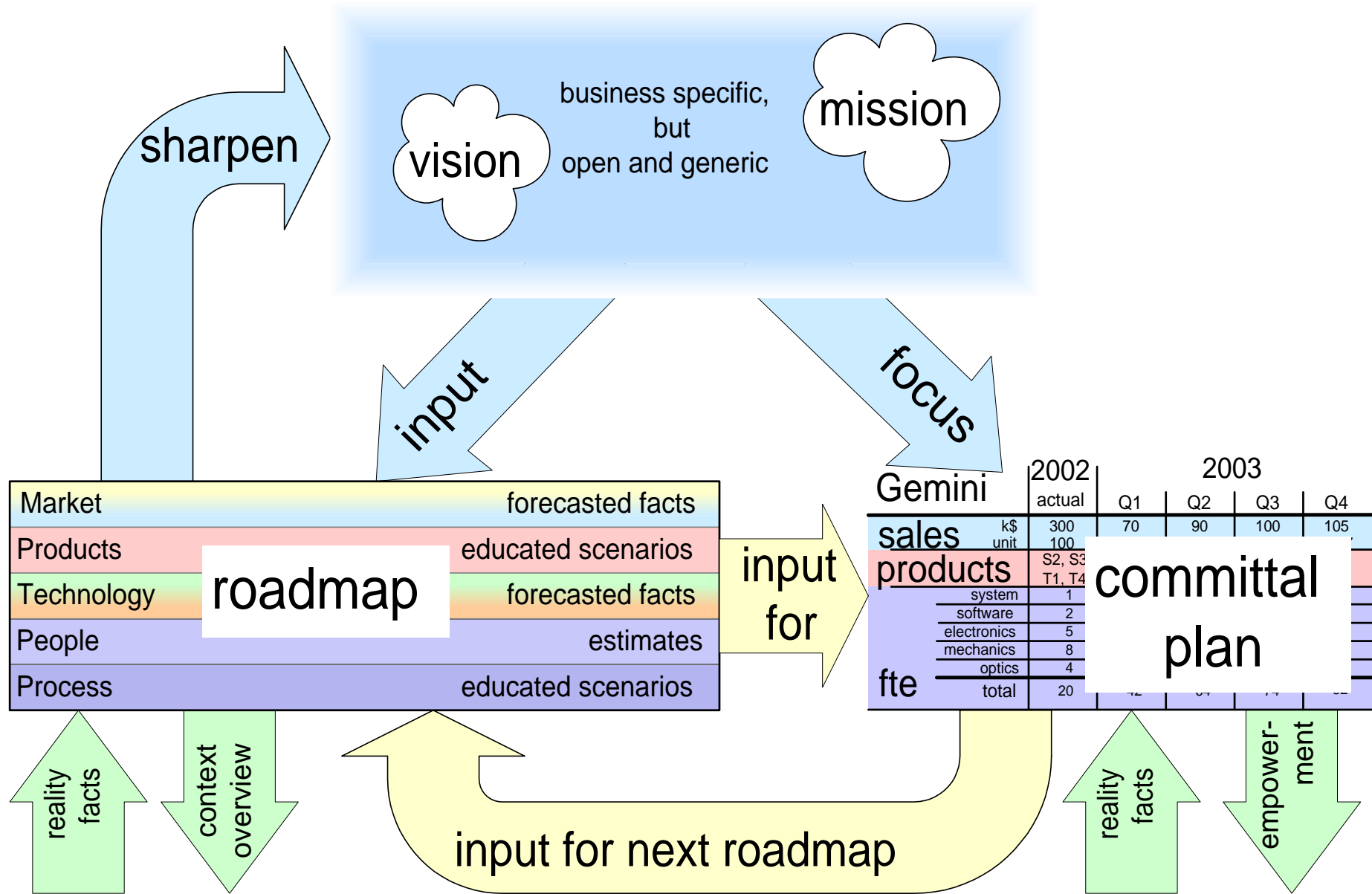
Attributes per Phase

	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.

Module Human Resource Management

by *Gerrit Muller* Embedded Systems Institute

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

`www.gaudisite.nl`

Abstract

The module Human Resource Management addresses the HRM aspects of systems architects, such as the profile of an architect, selection, education, appraisal and motivation.

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: 1.0

logo
TBD

Function Profiles; The Sheep with Seven Legs

by *Gerrit Muller* Buskerud University College

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

`www.gaudisite.nl`

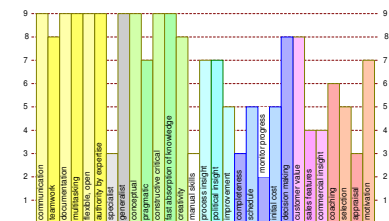
Abstract

The profile of a system architect is quantified for a large list of system architect related characteristics. For comparison the function profiles of related functions are given as well. This profile is based on personal observations and experience.

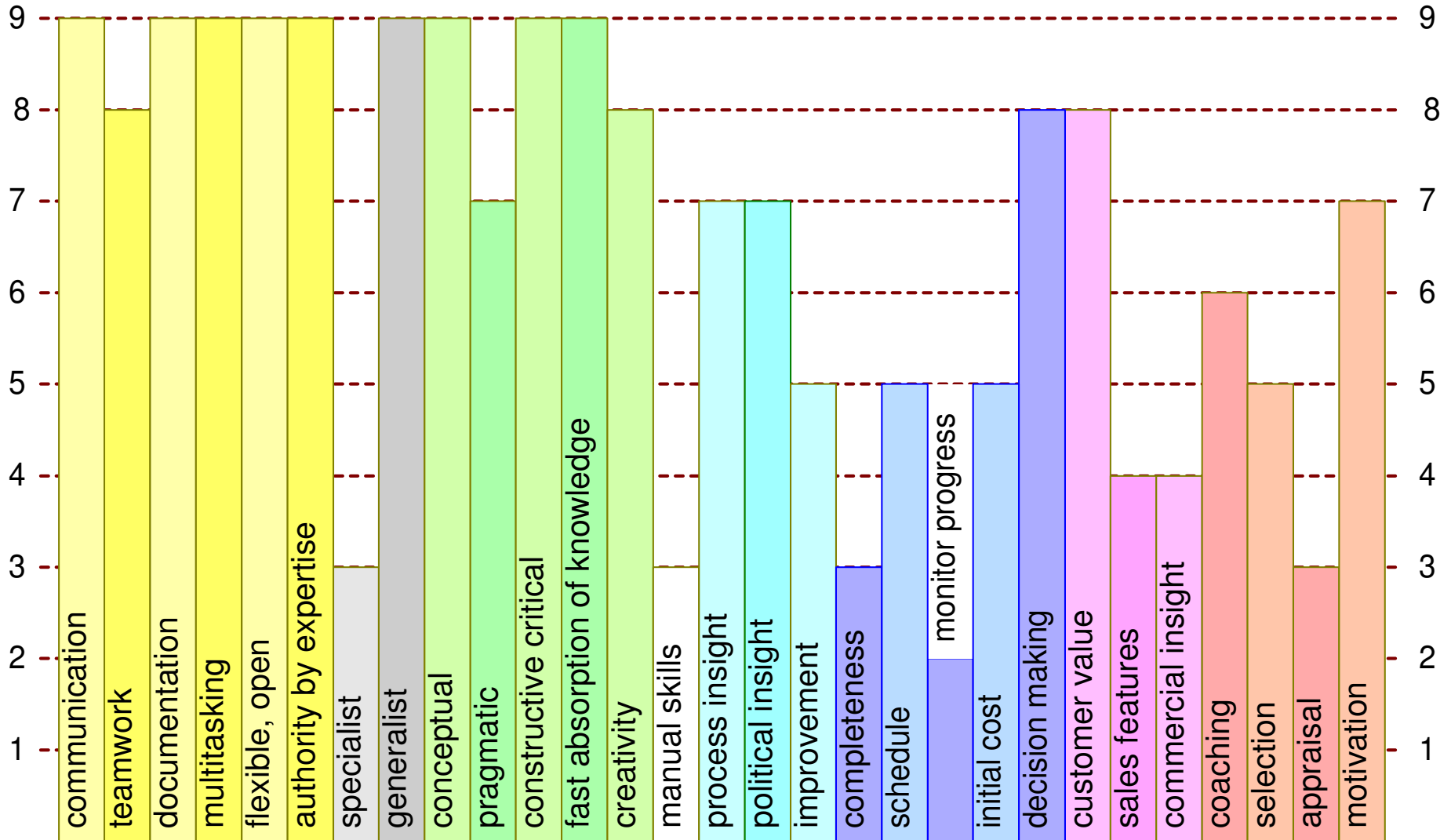
Distribution

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.

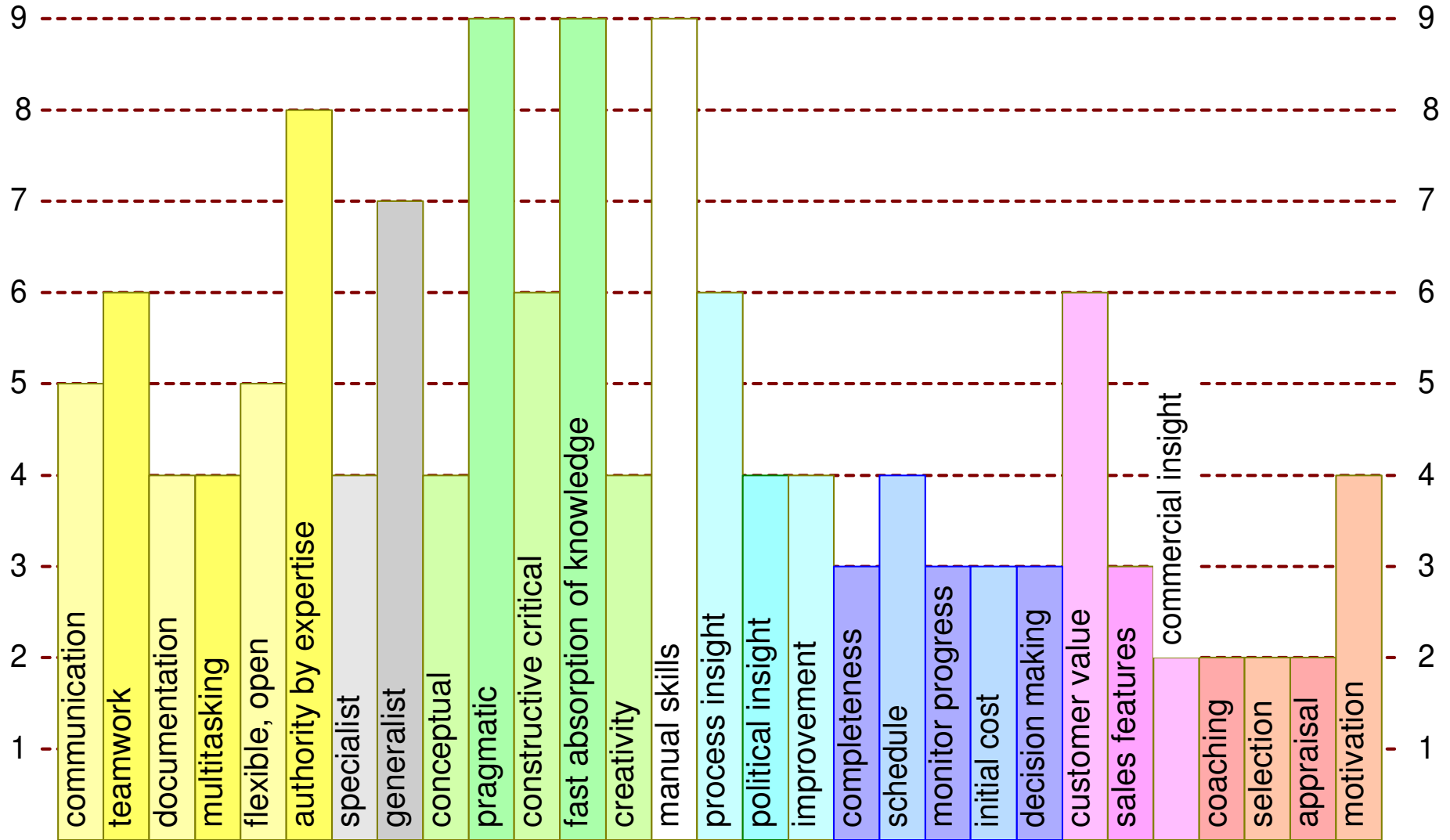
February 11, 2012
status: concept
version: 1.0



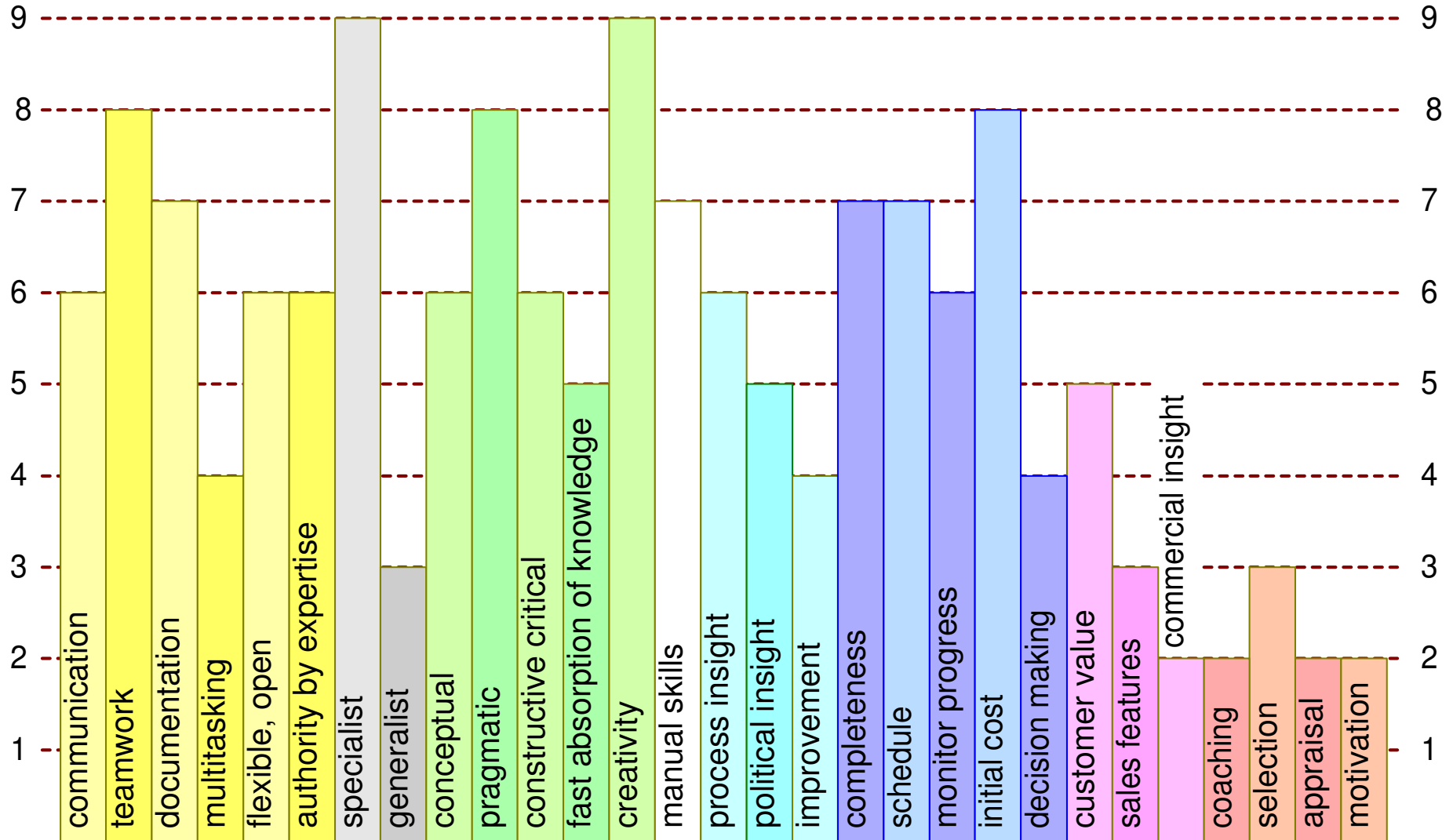
System Architect



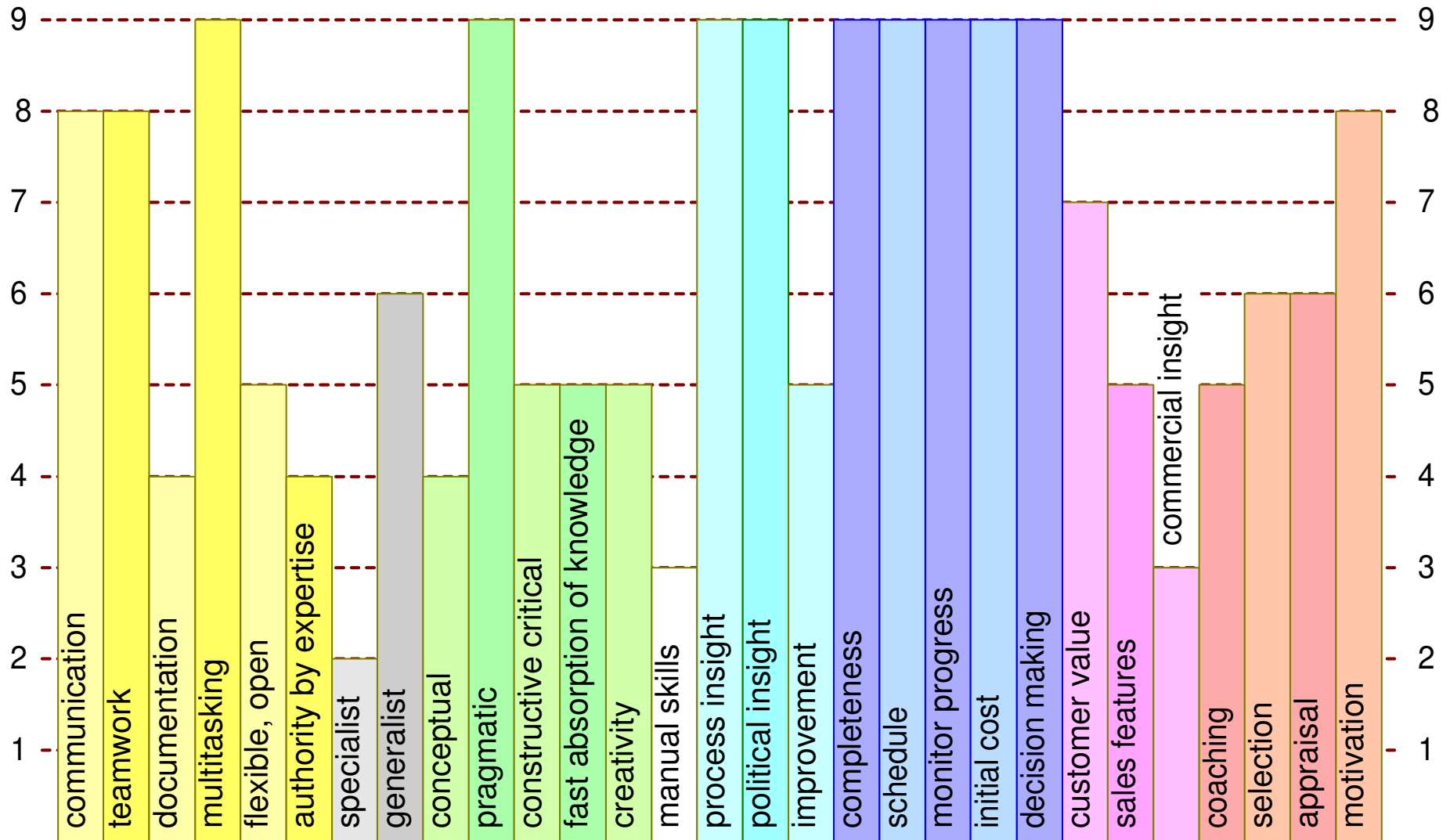
Test Engineer



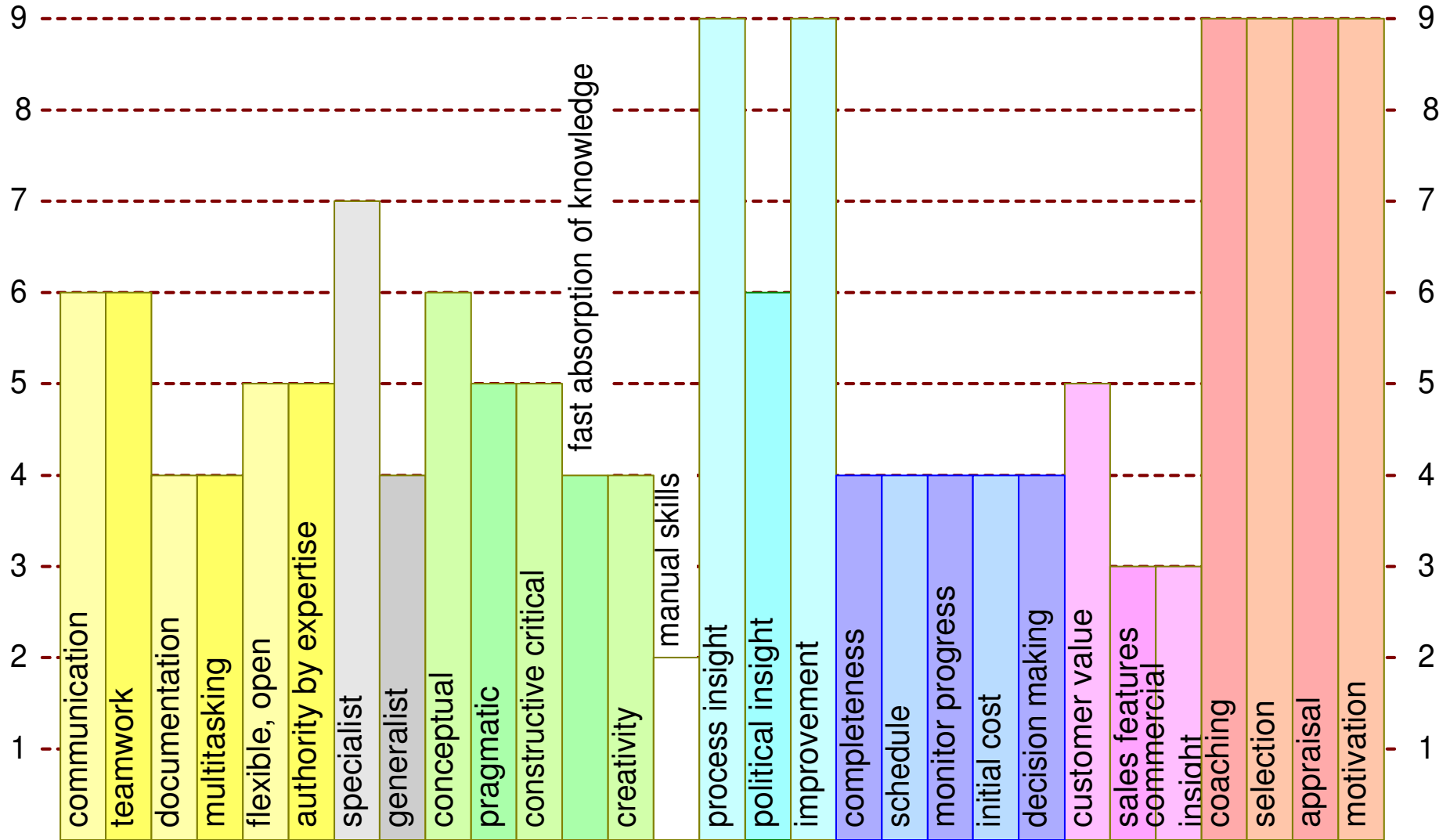
Developer



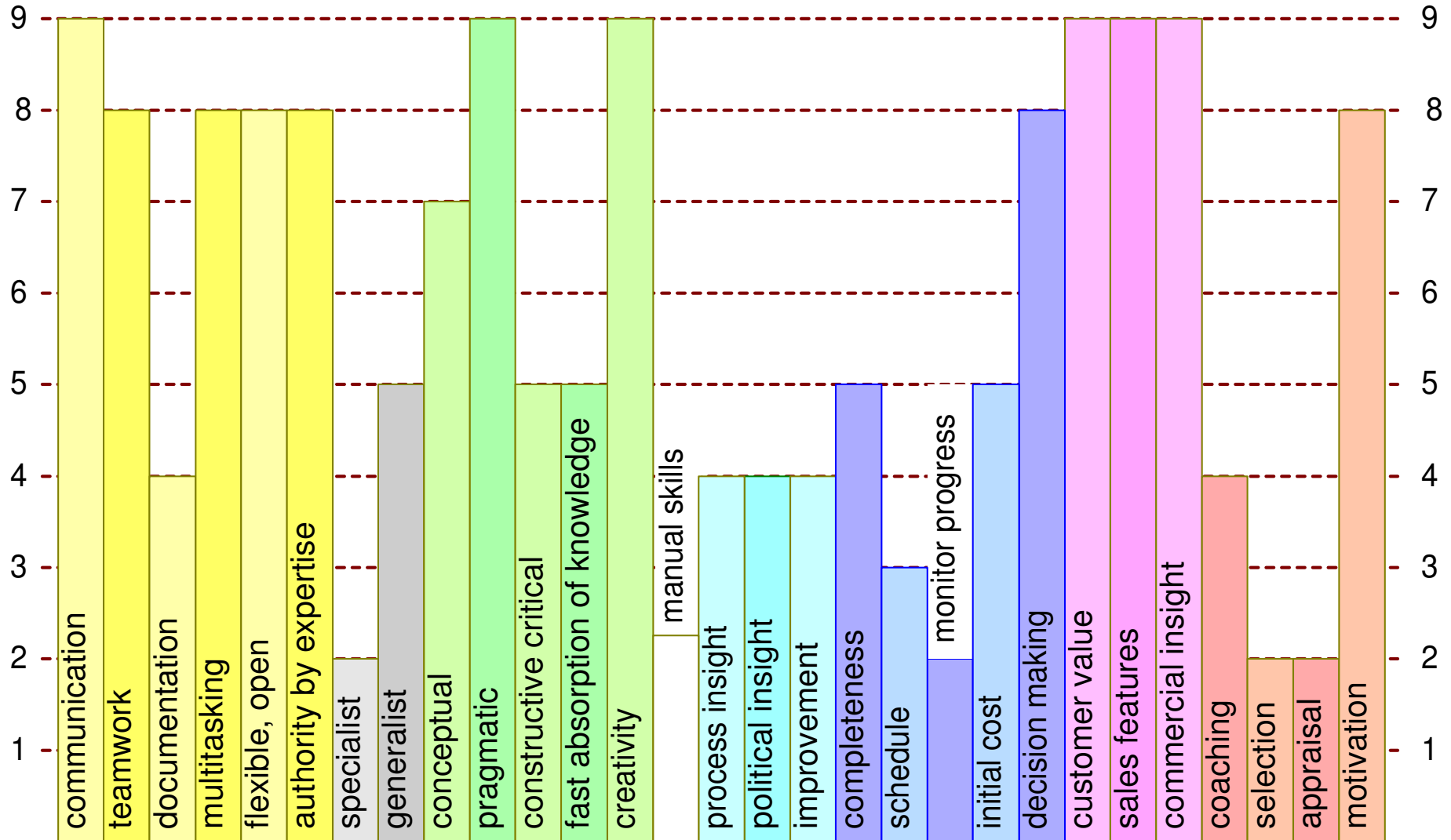
Operational Leader



Line Manager



Commercial Manager



The numbers behind the bars

	communication	teamwork	documentation	multitasking	flexible, open	authority by expertise	specialist	generalist	conceptual	pragmatic	constructive critical	fast absorption of knowledge	creativity	manual skills	process insight	political insight	improvement	completeness	schedule	monitor progress	initial cost	decision making	customer value	sales features	commercial insight	coaching	selection	appraisal	motivation
systems architect	9	8	9	9	9	9	3	9	9	7	9	9	8	3	7	7	5	3	5	2	5	8	8	4	4	6	5	3	7
test engineer	5	6	4	4	5	8	4	7	4	9	6	9	4	9	6	4	4	3	4	3	3	3	6	3	2	2	2	2	4
developer	6	8	7	4	6	6	9	3	6	8	6	5	9	7	6	5	4	7	7	6	8	4	5	3	2	2	3	2	2
operational leader	8	8	4	9	5	4	2	6	4	9	5	5	5	3	9	9	5	9	9	9	9	9	7	5	3	5	6	6	8
line manager	6	6	4	4	5	5	7	4	6	5	5	4	4	2	9	6	9	4	4	4	4	4	5	3	3	9	9	9	9
commercial manager	9	8	4	8	8	8	2	5	7	9	5	5	9	2	4	4	4	5	5	2	5	8	9	9	9	4	2	2	8

How to appraise or assess an architect?

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

Abstract

The appraisal of system architect is handicapped by the vague and abstract responsibilities of the system architect. The success criteria for architecting are discussed. An approach to "measure" or assess the architect is described.

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: 0.1

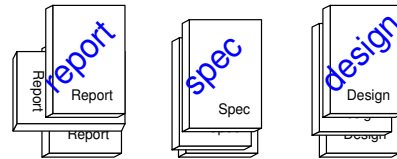


abstract (vague) responsibilities
lot of overlap of responsibilities

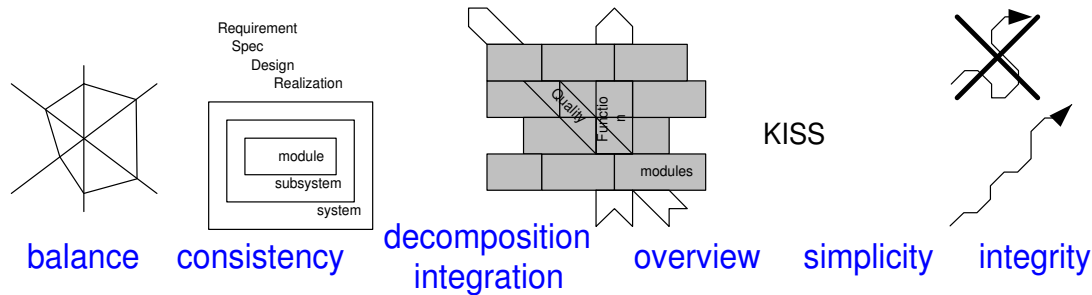
- difficult to define yardstick
- difficult to measure
- difficult to compare
- difficult to certify
- difficult to translate in (financial) consequences

How to assess an architect?

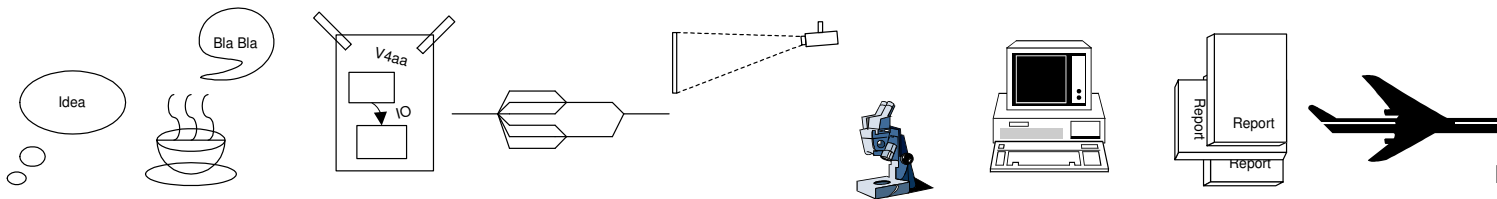
Tangible deliverables based upon many invisible activities



Deliverables
paperwork only



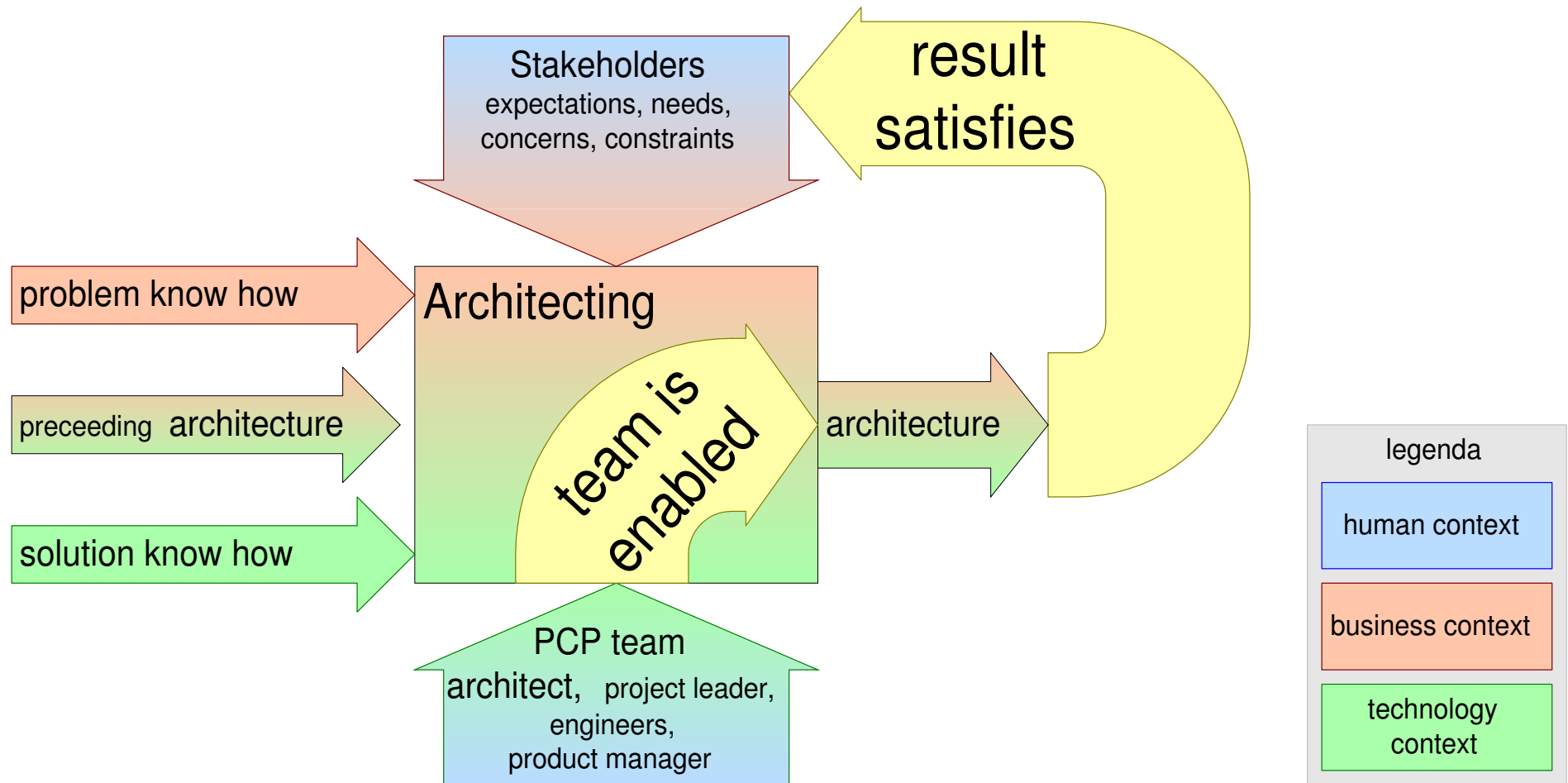
Responsibilities
abstract and qualitative



many very detailed
Activities
necessary but invisible

thinking, talking, discussing, scheduling, presenting, measuring, writing, reviewing, visiting customers
analyzing, listening, brainstorming, supporting, teaching, testing, reading, visiting trade-shows
simulating, communicating, troubleshooting, selling, integrating, browsing, consolidating, visiting suppliers

Criteria for successful architecting



Yardsticks for architect assessment

formalized expectations

function appraisal system,
f.i. from Hay Management Consultants

impact
scope of control
freedom of thinking

job description

deliverables
timing

career development plan

skills
know how

actual architect performance

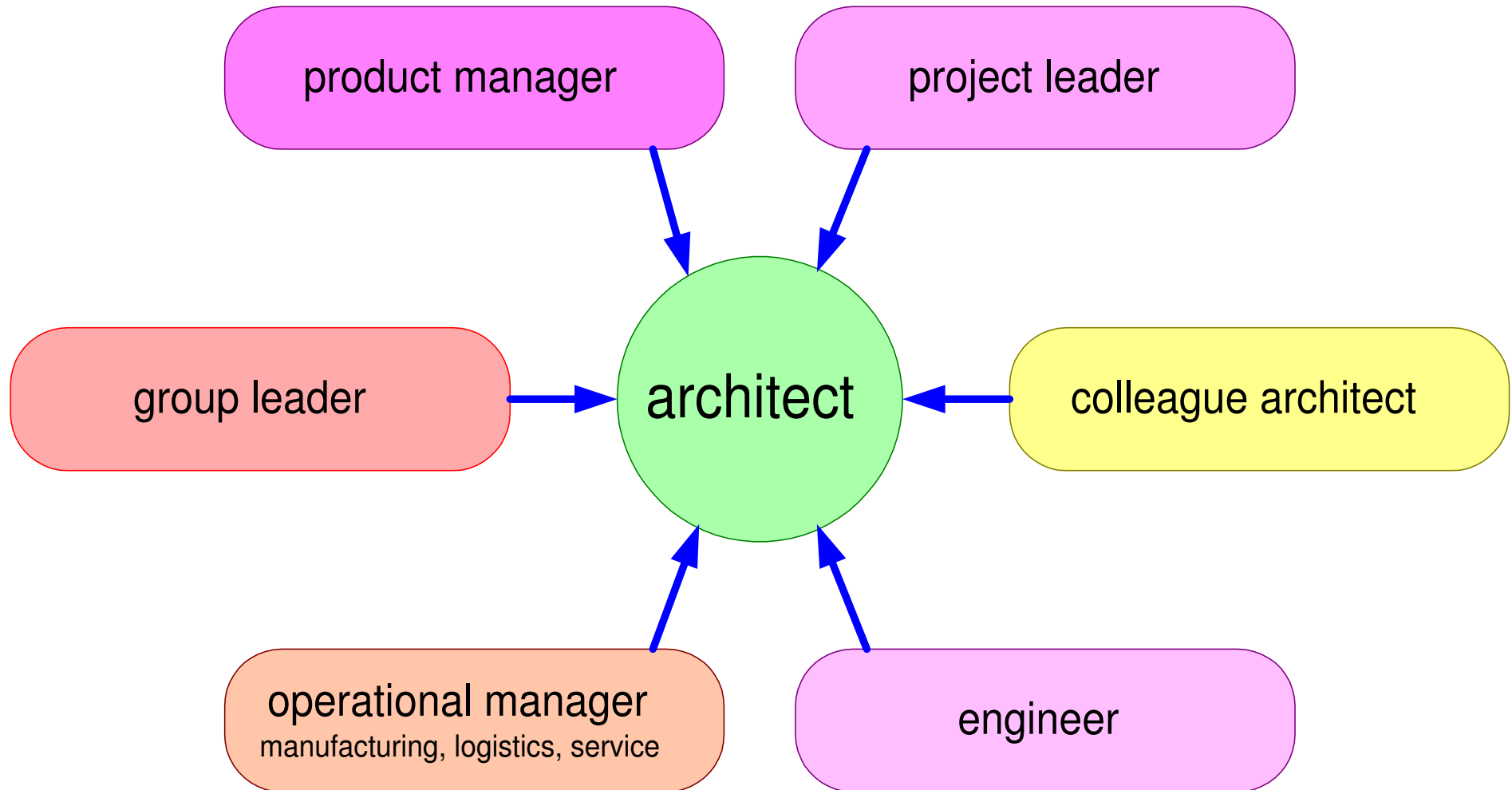
architecture fitness

sales turnover
business success
market continuity

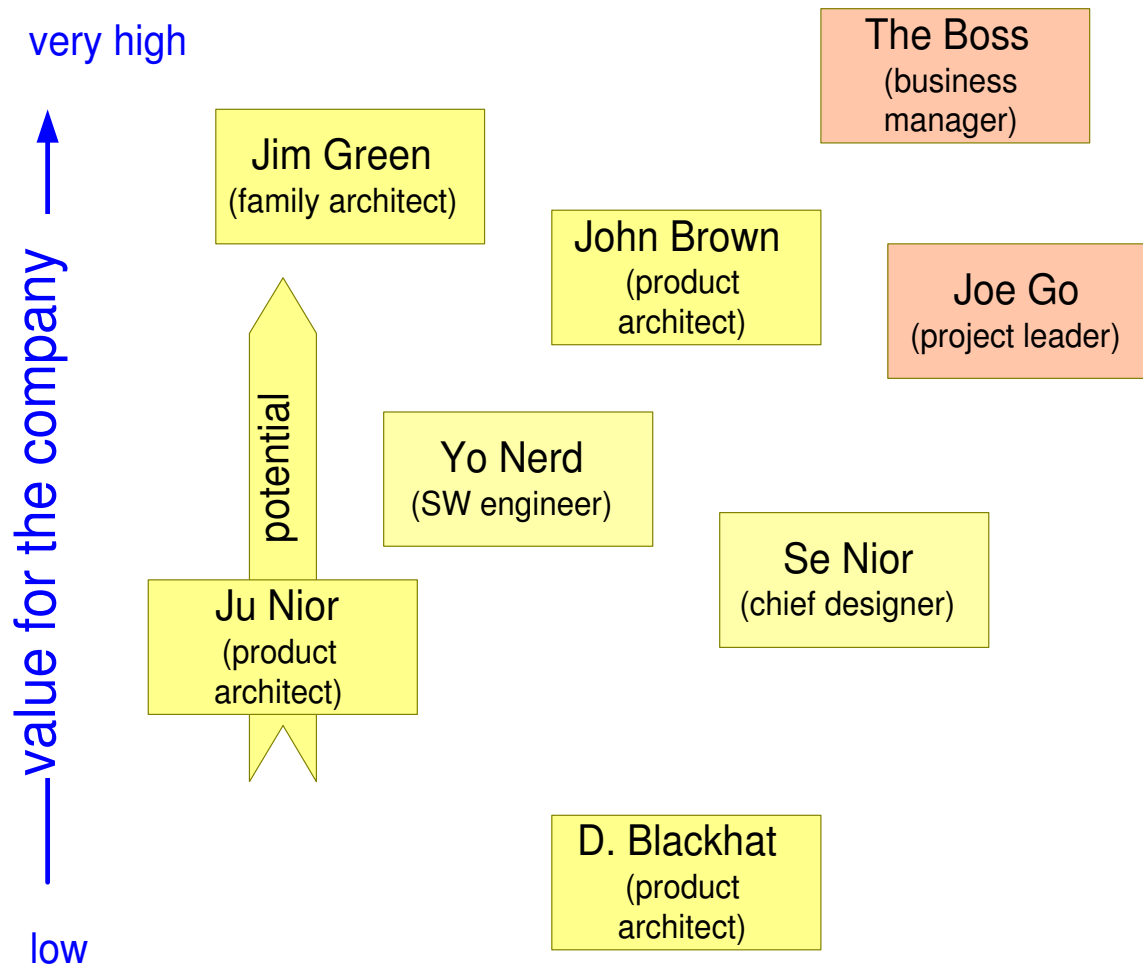
internal *stakeholder* satisfaction

contribution
deliverables
timing
skills
know how

360 degree assessment



Ranking as trigger for discussions



1 ask for ranking

2 ask for justification
(why ...?)

3 clarify criteria

4 iterate ranking and
justification