

Lecture Requirements Engineering

by *Gerrit Muller* Embedded Systems Institute

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

`www.gaudisite.nl`

Abstract

This article describes the Requirements Engineering session part of the Software Engineering block in the OOTI curriculum of the Technical University Eindhoven.

The focus of this course is on capturing and managing requirements. The notion of key drivers and story telling will be introduced as a means to capture and manage. During the course an exercise is used based on video distribution via satellite. The students have to elicit the requirements for the required systems, working in teams of 4 students. Every student writes an individual report about the exercise.

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.

July 20, 2011
status: finished
version: 1.1

block 1: teacher provides case	
What are requirements, black box, SMART	1/2 day
homework: make requirement specification	
Customer and Application view, Story telling	1/2 day
homework: improve requirement specification	
Discussion of requirement specification per team	1/2 day

block 2: actual current case of OOTI education	
Financial viewpoint, Presentation to management	1/2 day
homework: make presentation outline	
Documentation How-to Coaching and discussion	1/2 day
homework: make presentation	
Presentation project case to management team	1/2 day
individual report	

Program

time	subject
Session 1	What are requirements, black box, SMART
Session 2	Customer and Application view, Story telling
Session 3	Discussion of requirement specification per team
Session 4	Financial viewpoint, Presentation to management
Session 5	Documentation How-to, Coaching and discussion session
Session 6	Presentation of project case to management team

Schedule

block 1; teacher provides case

What are requirements, black box, SMART	1/2 day
homework: make requirement specification	
Customer and Application view, Story telling	1/2 day
homework: improve requirement specification	
Discussion of requirement specification per team	1/2 day

block 2; actual current case of OOTI education

Financial viewpoint, Presentation to management	1/2 day	
homework: make presentation outline		
Documentation How-to	Coaching and discussion	1/2 day
homework: make presentation		
Presentation project case to management team	1/2 day	
individual report		

Case Instructions

1. Block 1 session 1: Make an initial requirements specification
2. Block 1 session 2: Improve and complete requirements specification
3. Block 2 session 4: Make an outline of a presentation of maximum 10 minutes, target audience: management team of your company
4. Block 2 session 5: Prepare and exercise presentation
5. Block 2 session 6: Write an individual report reflecting on: requirement specification, management presentation, lessons learned and how to do it next time.

Case: Recommended Steps

1. Make a black box view of the system
2. Make some initial drafts and designs to explore the problem.
3. Make a story which helps to understand the products, make sure to use the criteria for a story.
4. Look from all stakeholder points of view towards the problem and identify what they need and what they expect.
5. Analyze the information obtained so far and extract the underlying requirements.
6. Abstract the key drivers behind the requirements.
7. Make a top-down description of the requirements.

Case: Questions for Individual Report

- What are the most important lessons you learned from these exercise (requirement specification, management presentation)?
- Which roles did the members of the group play during the exercise?
- How would you approach such a problem the next time?
- Which stakeholders understand your group presentation? Are they happy with the presentation?

Submission of Homework

Homework instructions

specification minimum 4 hours work/person, maximum 8 hours

maximum 6 A4 pages

presentation minimum 2 days work/ person, maximum 5 days

filename: [OOTI<year>] spec|presentation <team id>.<version number>

e.g. [OOTI2008] spec team1.1.doc

all team members on front page

email to: <gerrit . muller @ gmail . com>

subject: [OOTI<year>] spec|presentation <team id>

from/cc: <all email addresses of team members>

when: 48 hours before next lecture

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.

July 20, 2011
status: concept
version: 1.3



Embedded Systems
INSTITUTE

Fundamentals of Requirements Engineering

by *Gerrit Muller* Buskerud University College

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

`www.gaudisite.nl`

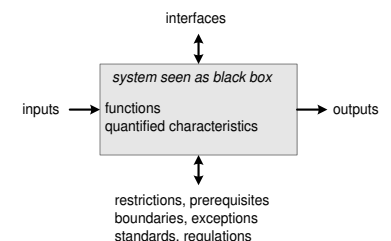
Abstract

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

Distribution

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

July 20, 2011
status: concept
version: 0.1



Definition of “Requirement”

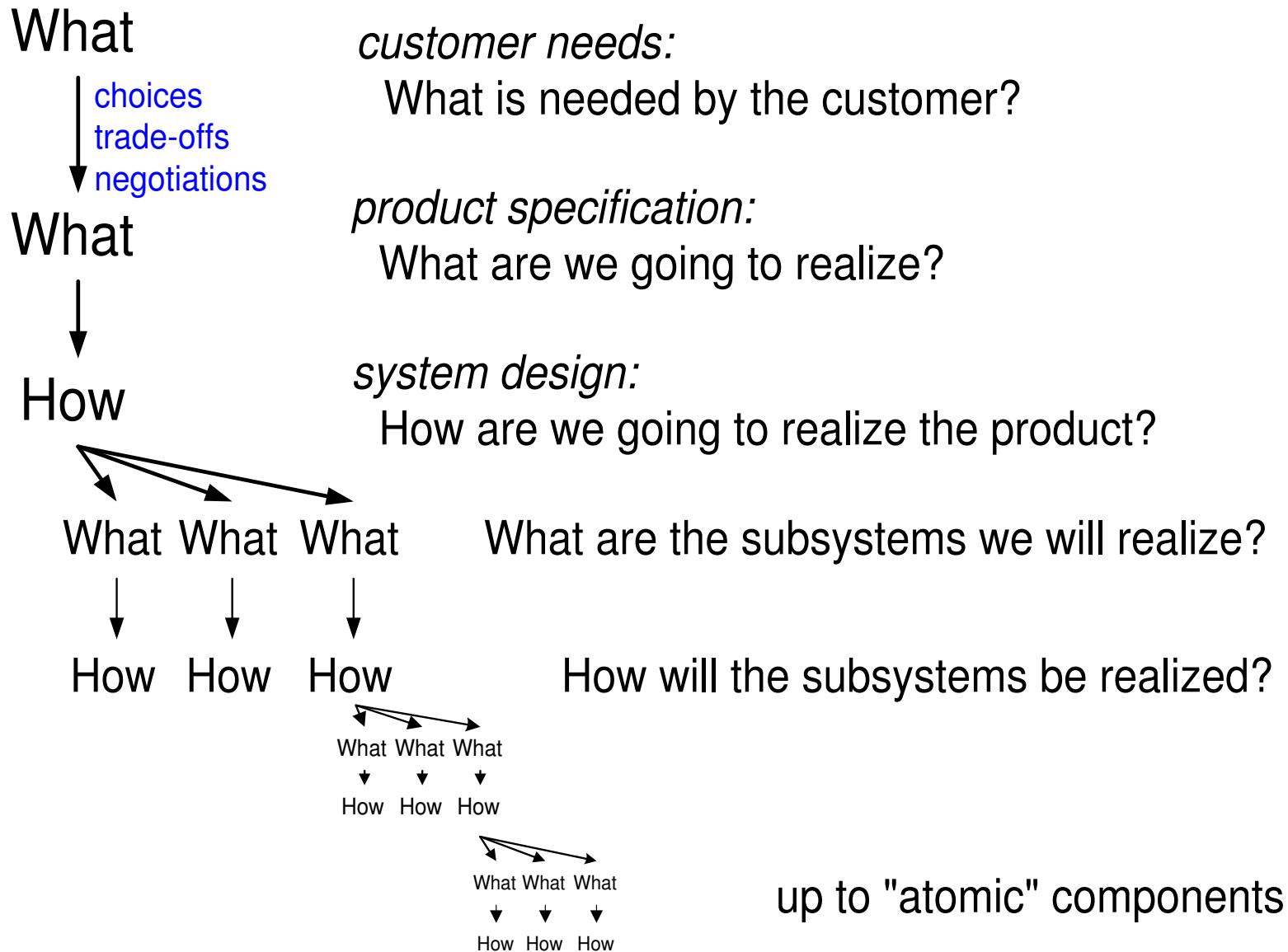
Requirements describing the needs of the customer:
Customer Needs

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

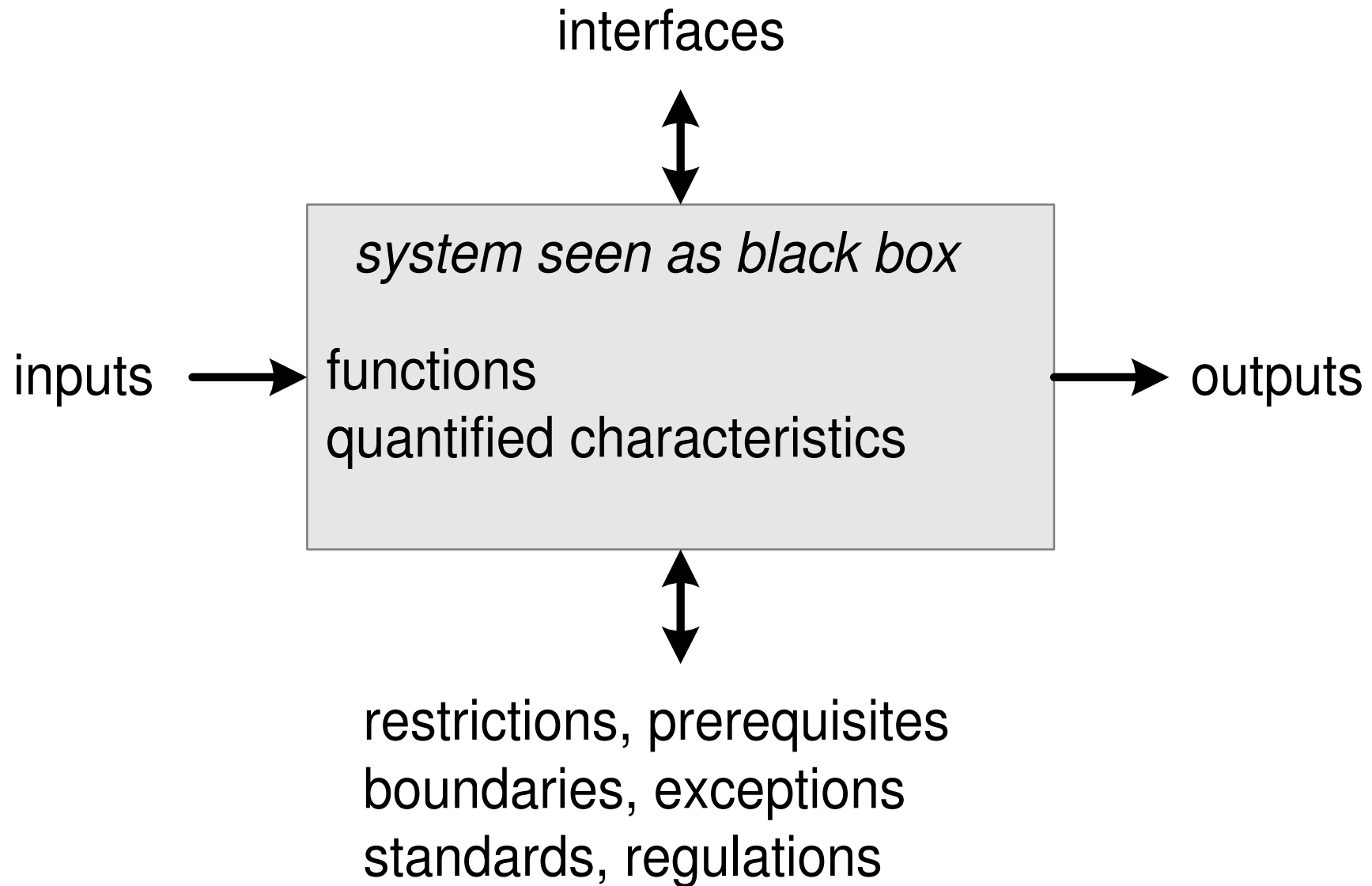
The requirements management process recursively applies definition 2 for every level of decomposition.

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

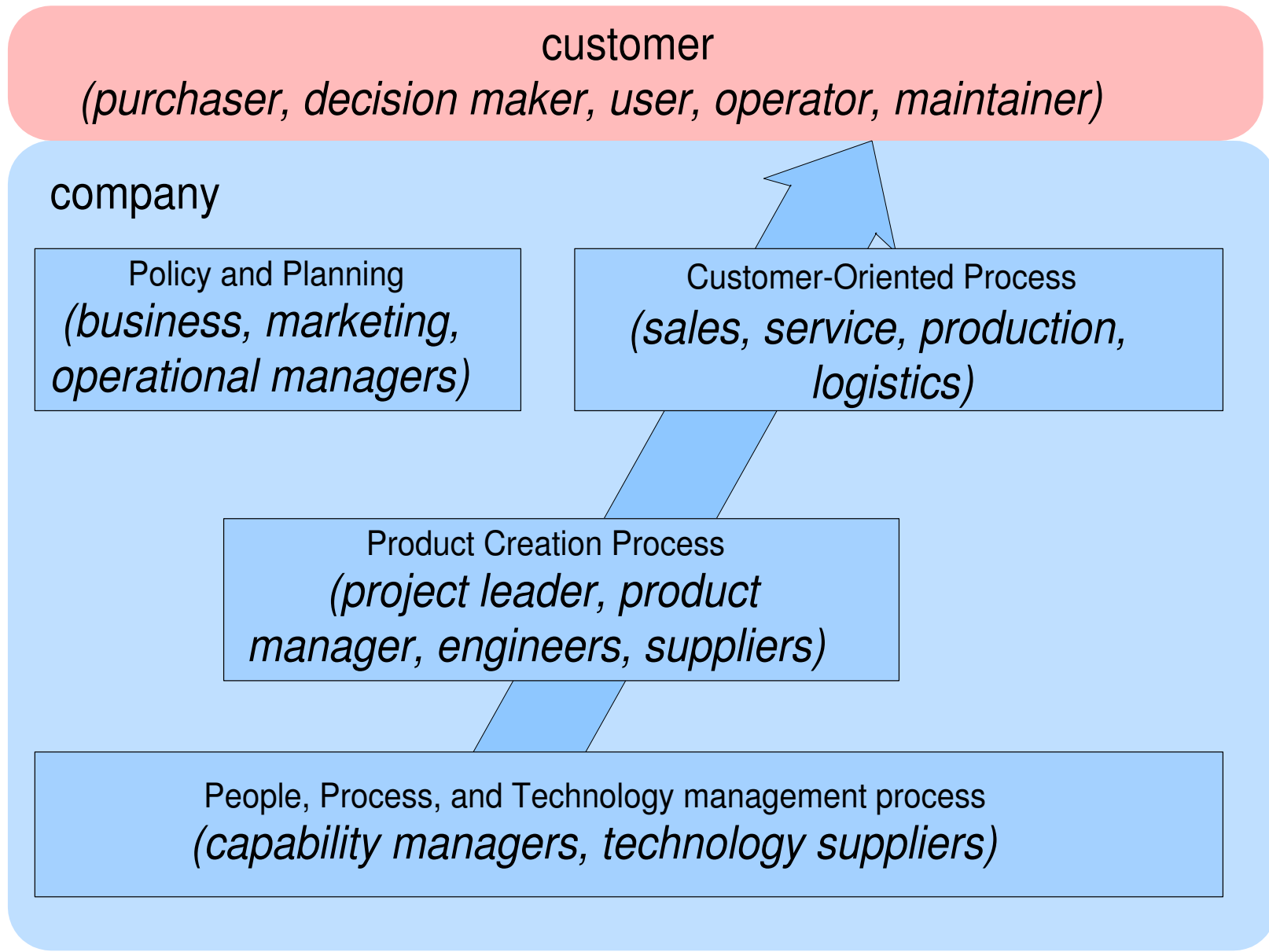
Flow of Requirements



System as a Black Box



Stakeholders w.r.t. Requirements



The “Formal” Requirements for Requirements

Specific

Unambiguous

Verifiable

Quantifiable

Measurable

Complete

Traceable

The Requirements to Enable Human Use

Accessible

Understandable

Low threshold

Short introduction to basic “CAFCR” model

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

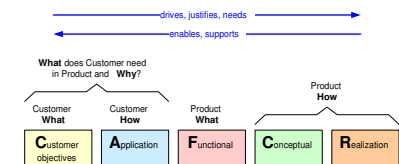
Abstract

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

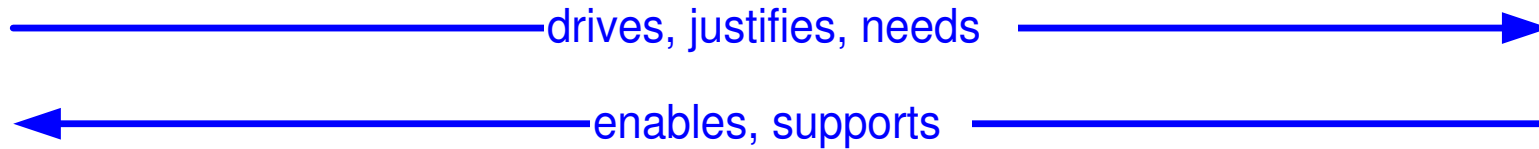
Distribution

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

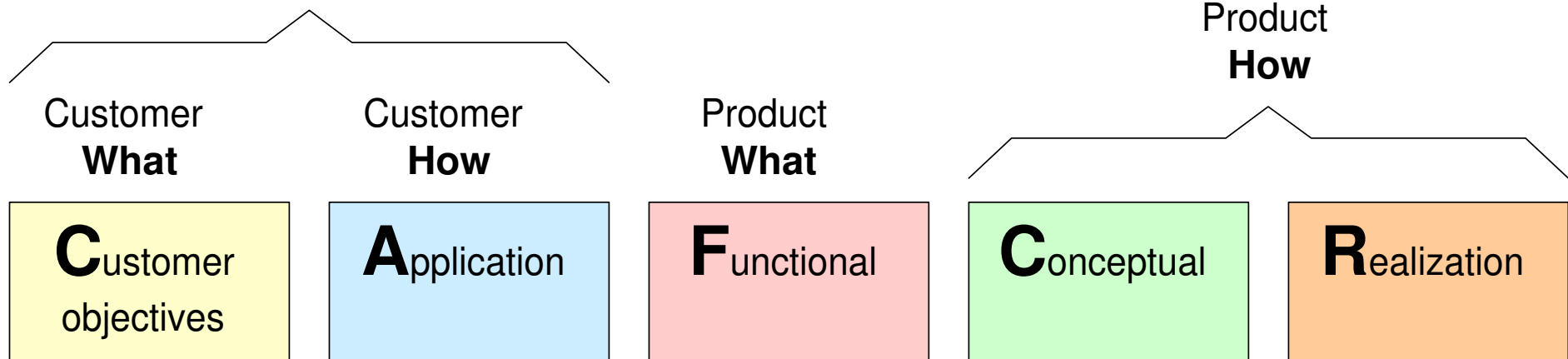
July 20, 2011
status: draft
version: 0.4



The “CAFCR” model

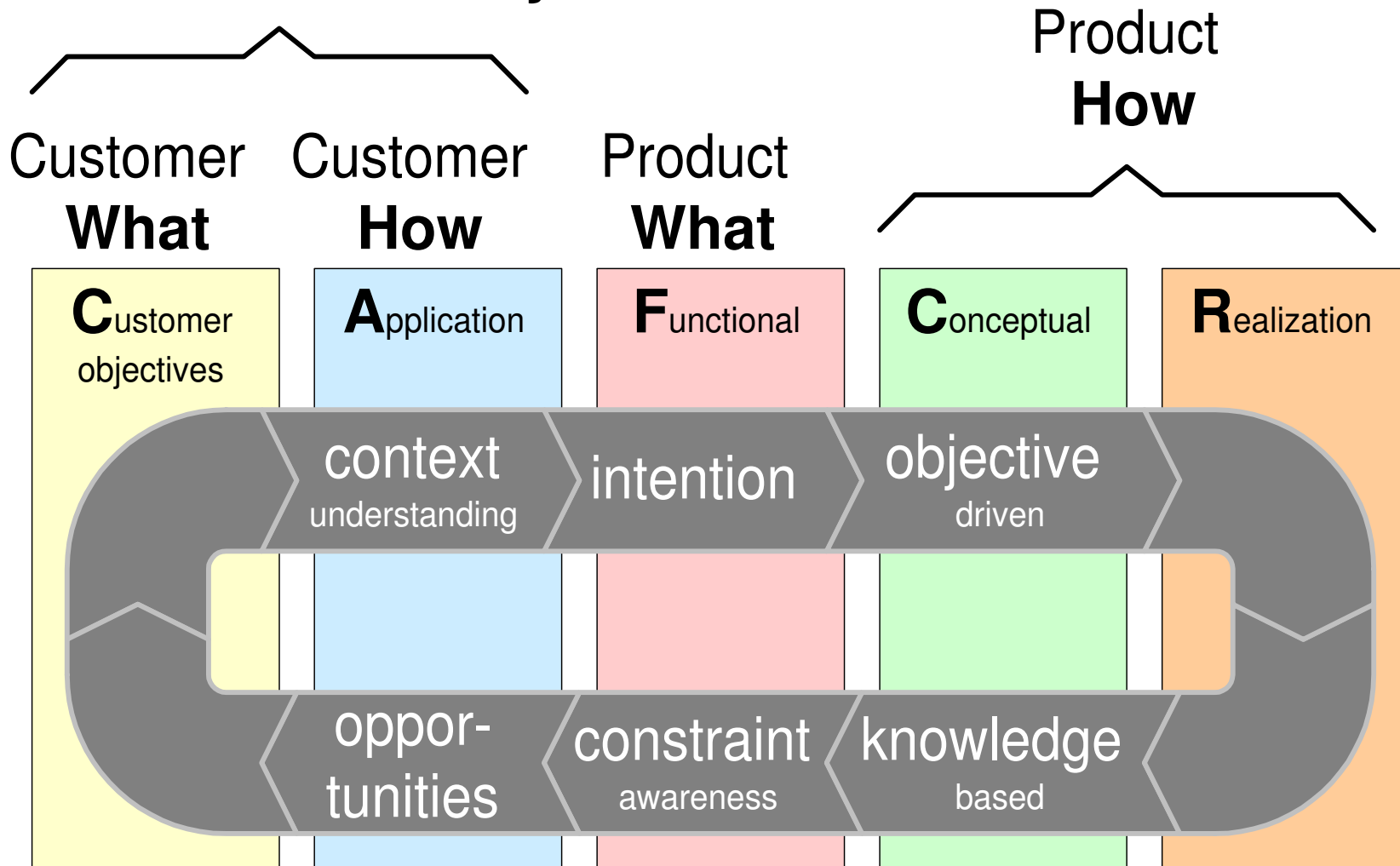


What does Customer need
in Product and **Why?**

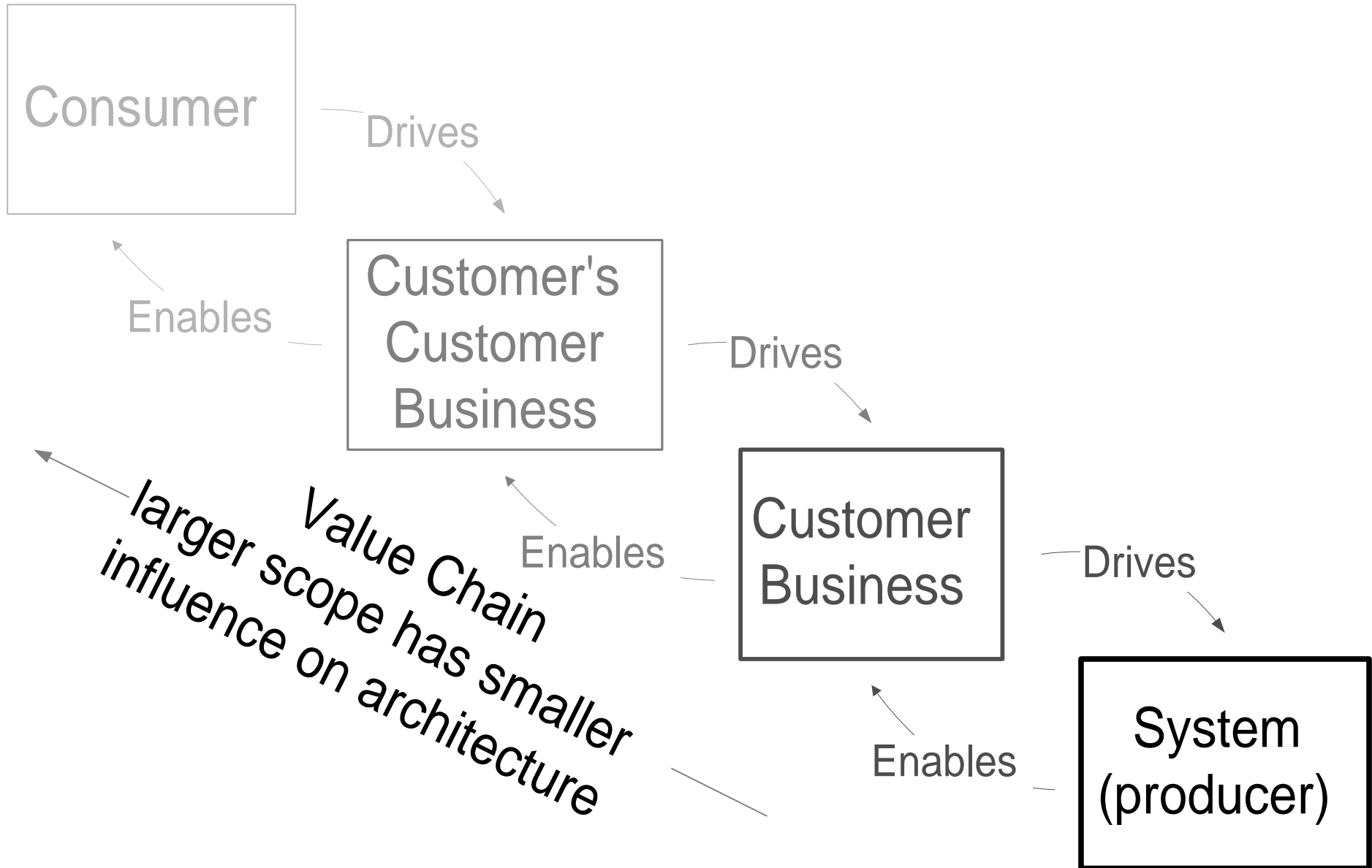


Integrating CAFCR

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



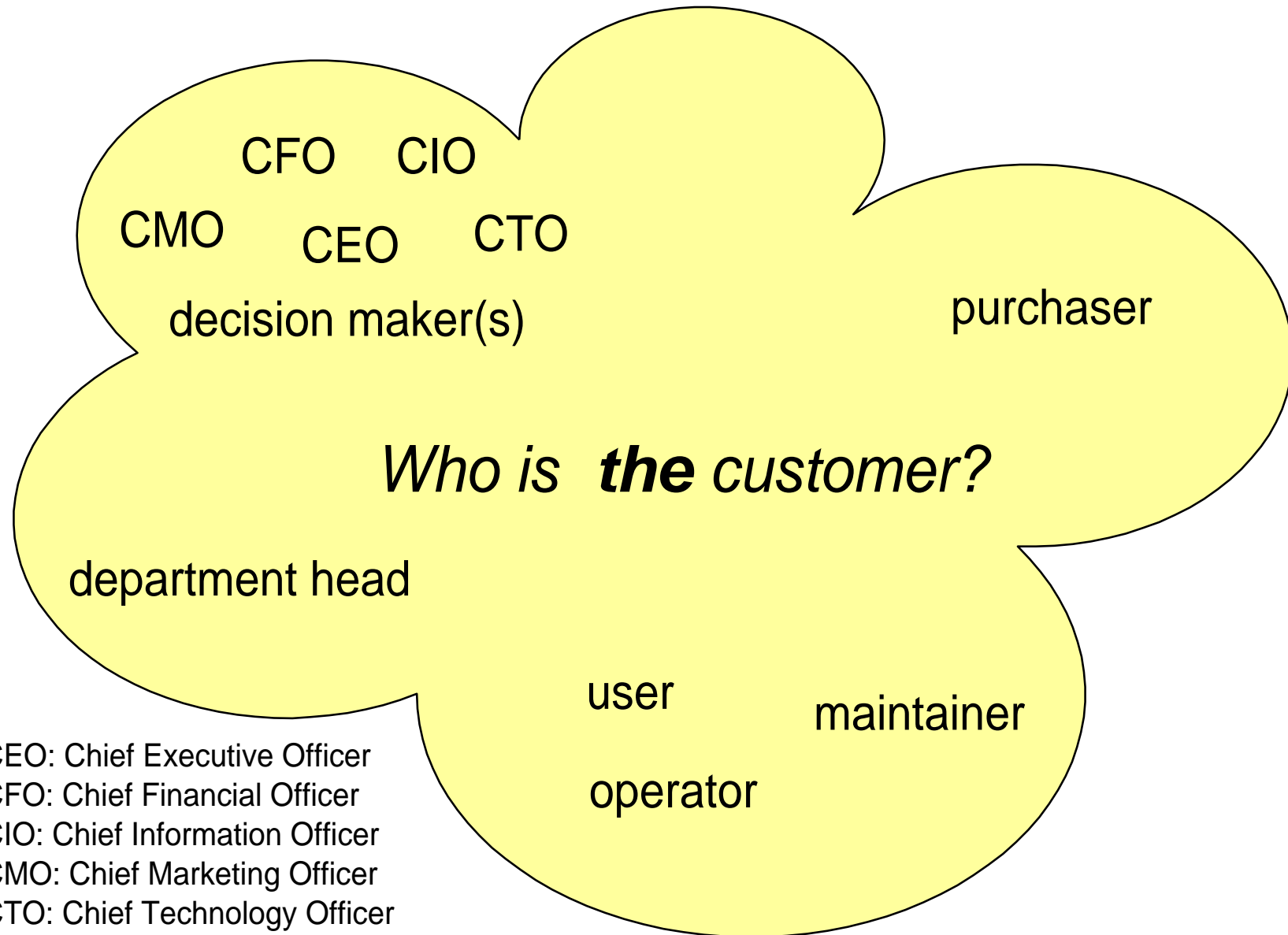
CAFCR can be applied recursively



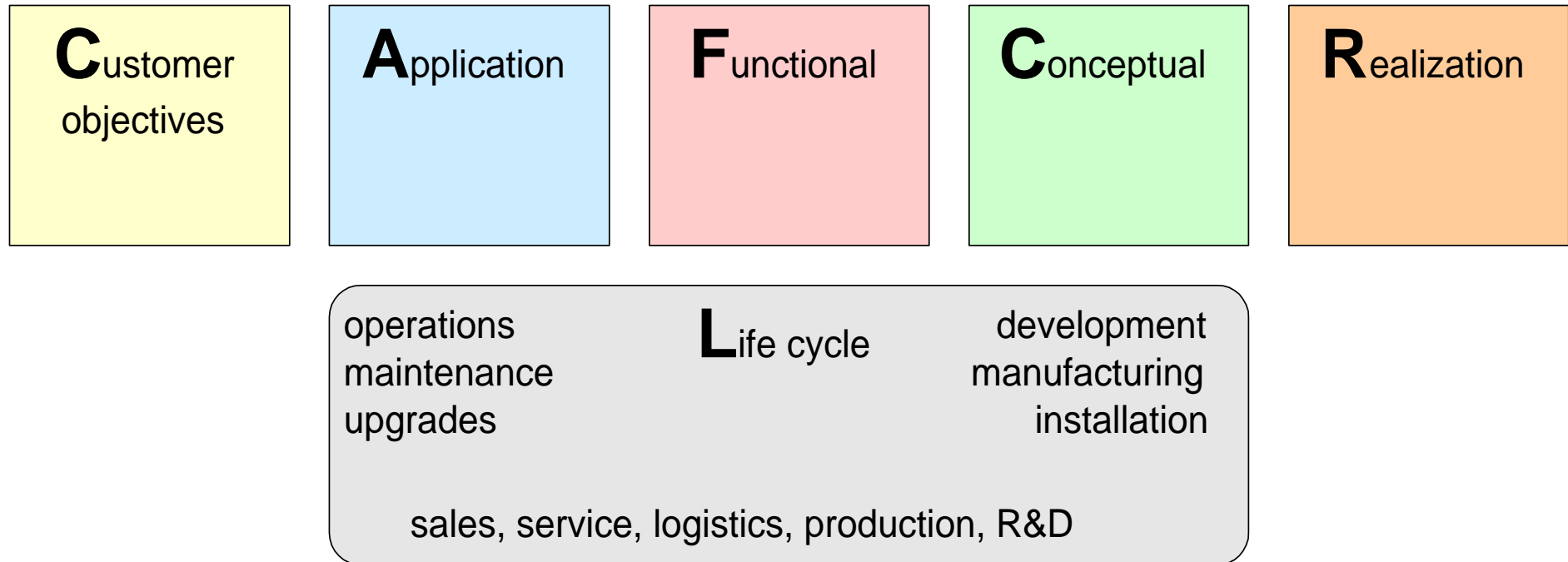
Market segmentation

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`

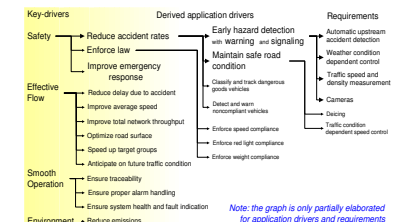
Abstract

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

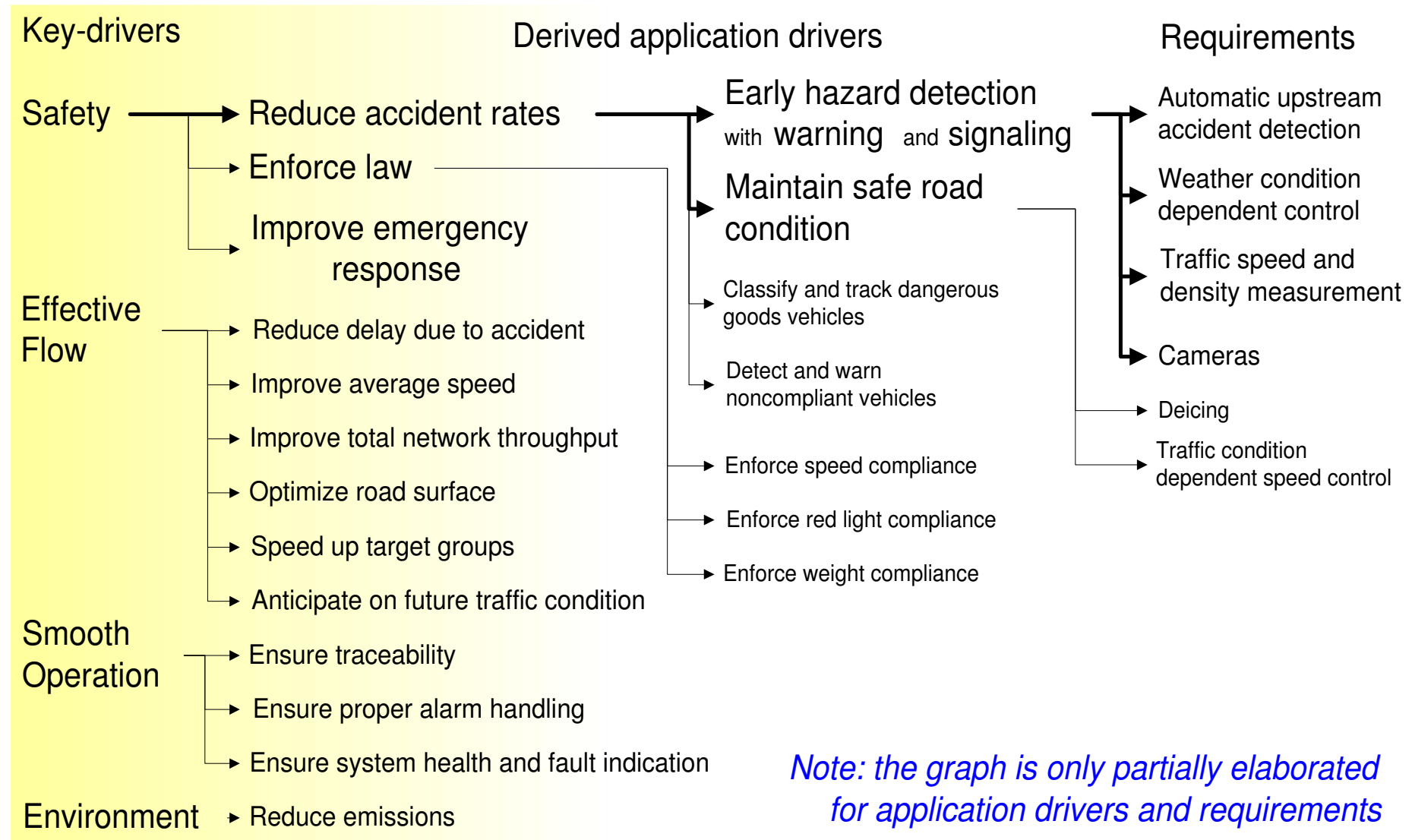
Distribution

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.

July 20, 2011
status: draft
version: 0.2



Example Motorway Management Analysis



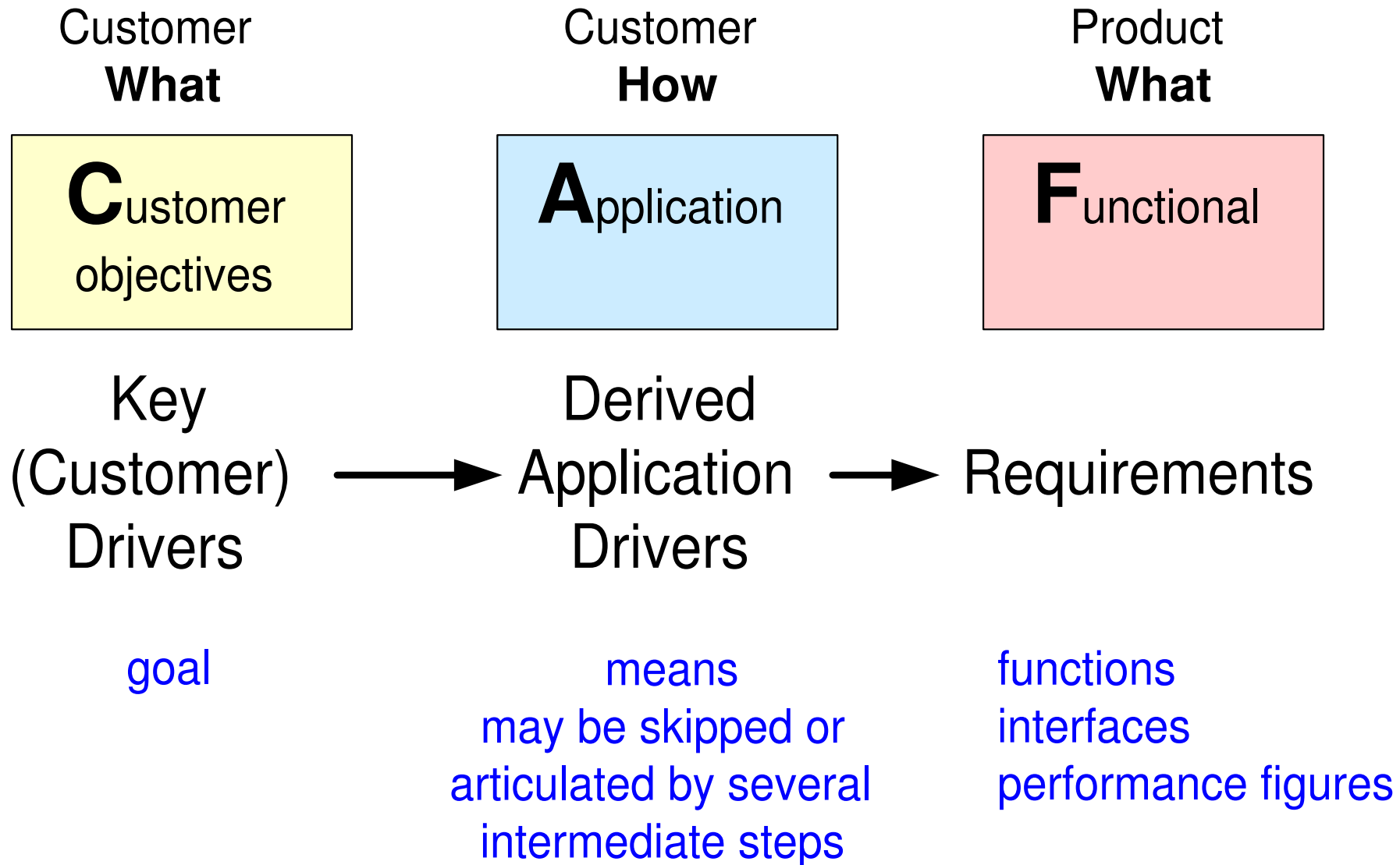
Method to create Key Driver Graph

- | | |
|--|--|
| • 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`

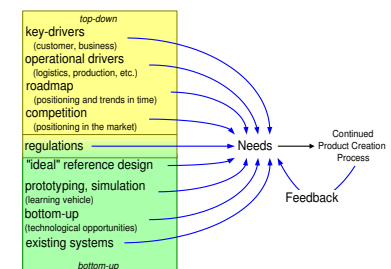
Abstract

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.

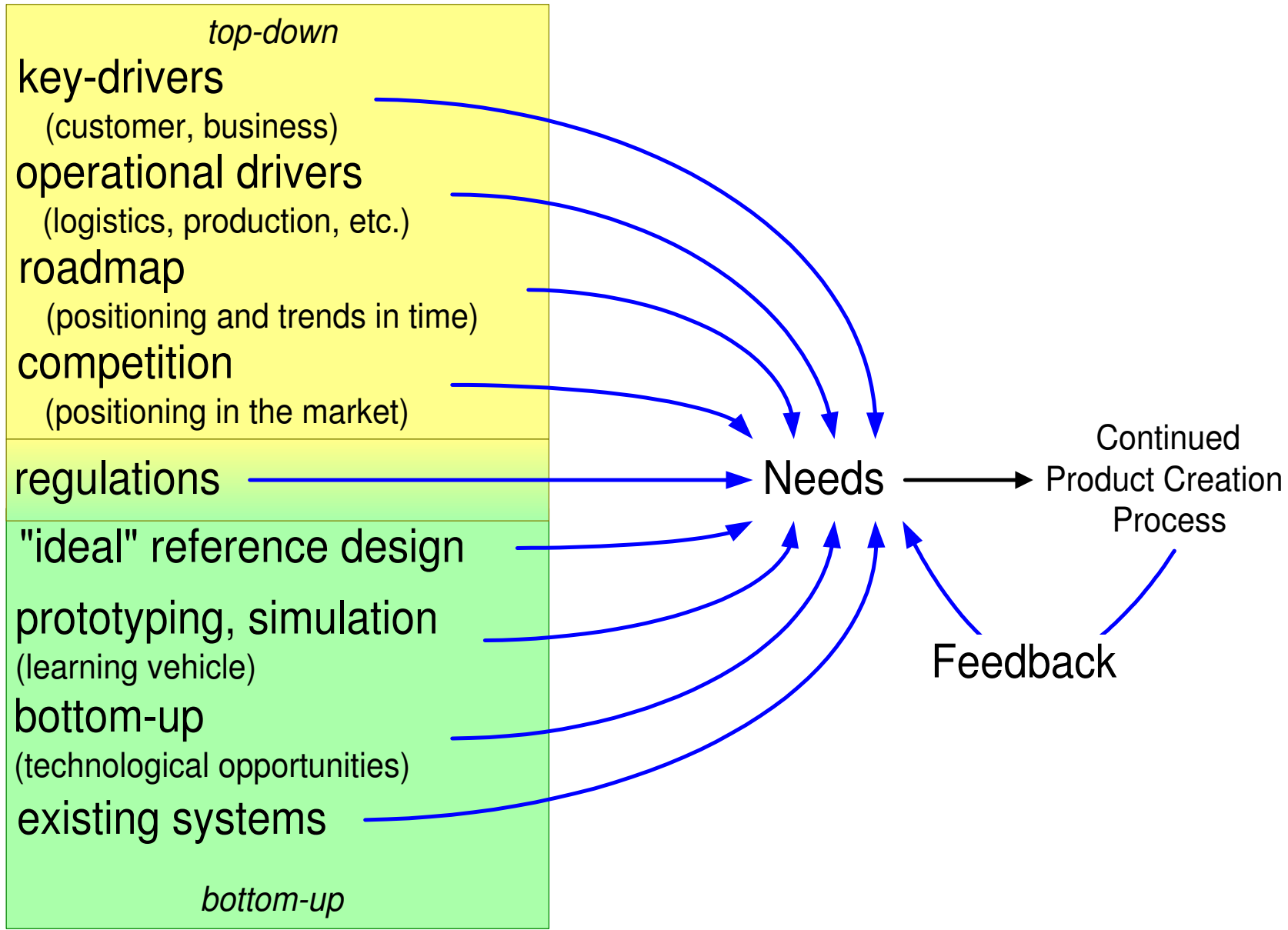
Distribution

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.

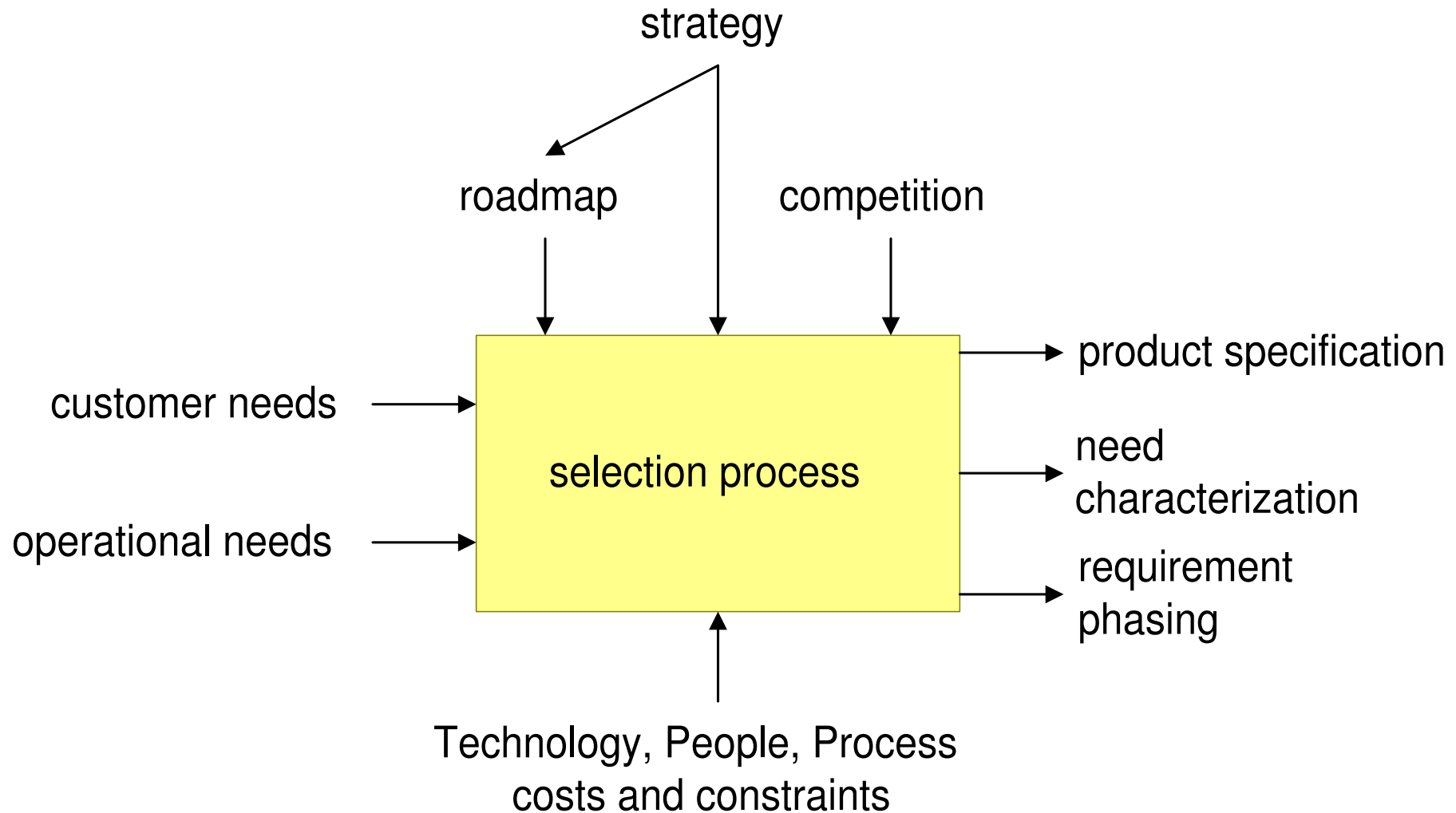
July 20, 2011
status: draft
version: 0



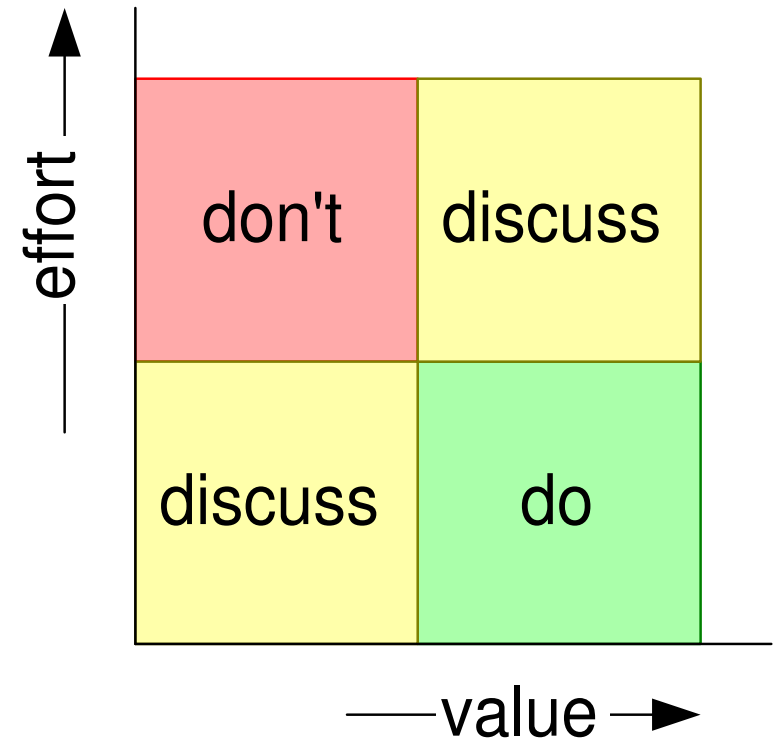
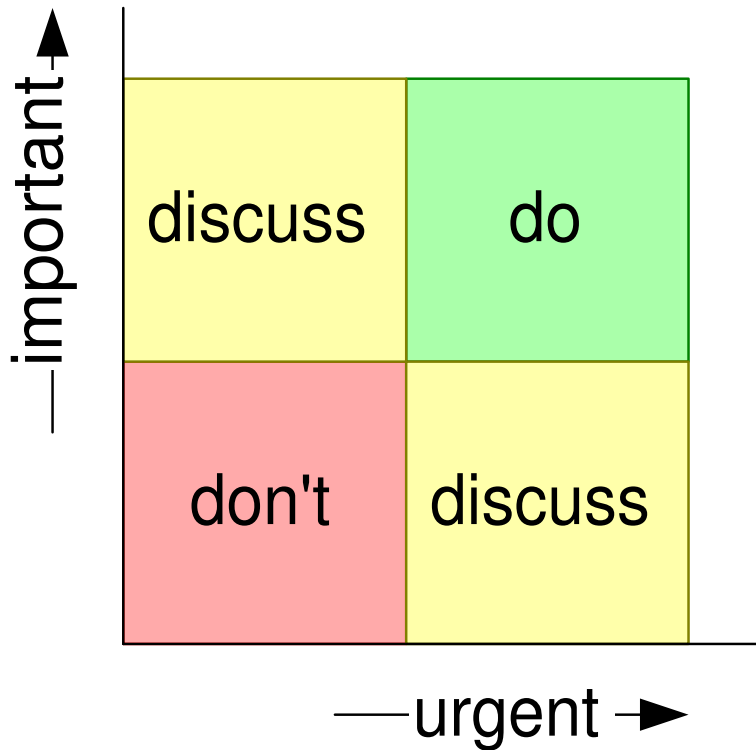
Complementary Viewpoints to Capture Requirements



Requirement Selection Process



Simple Qualification Method



Examples of Quantifiable Aspects

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

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

Ask several knowledgeable people to score

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

Exercise Requirements Capturing

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

Story How To

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

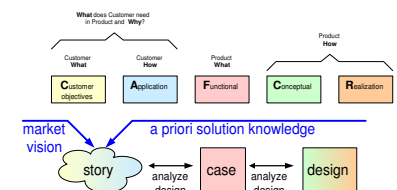
Abstract

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

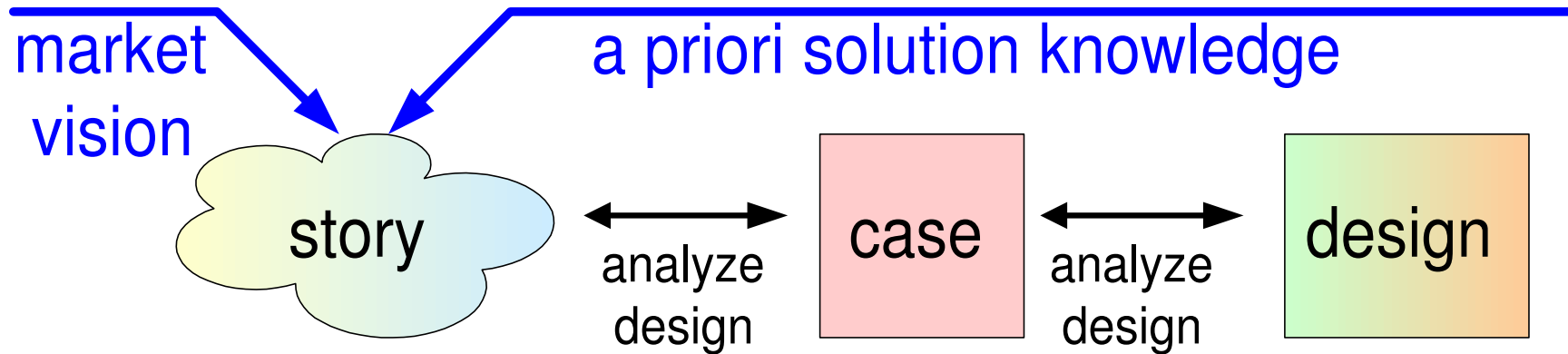
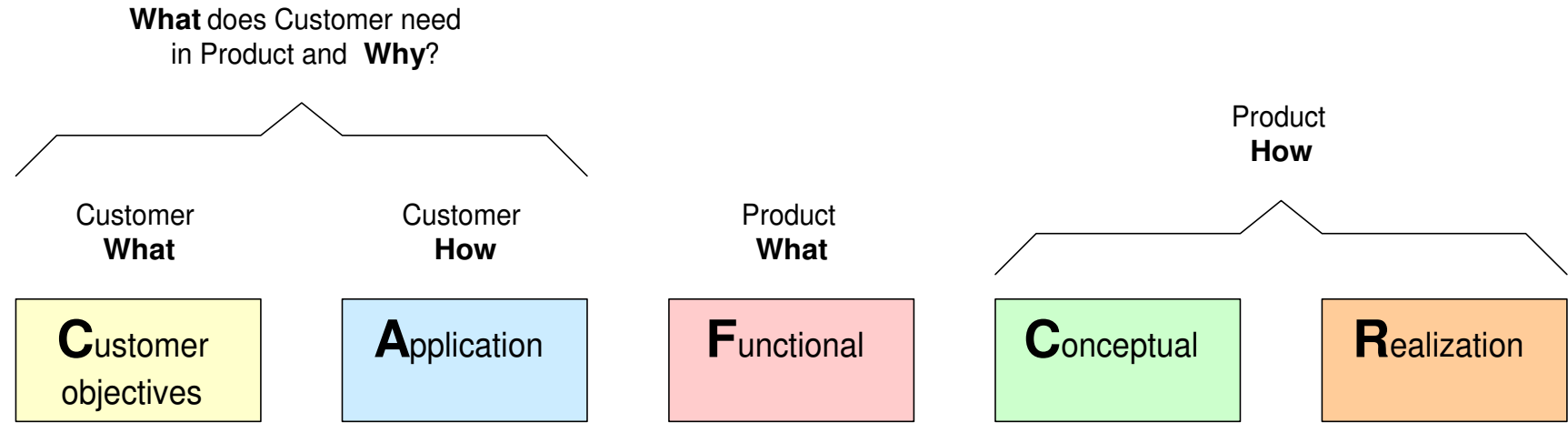
Distribution

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

July 20, 2011
status: concept
version: 1.1



From story to design



Example story layout

ca. half a page of
plain English text

A day in the life of Bob

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

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

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

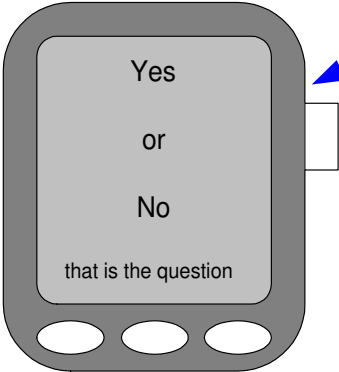
In the middle of the night he
is awake and decides to
change the world forever.

The next hour the great
event takes place:

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

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

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



draft or sketch of
some essential
appliance

Points of attention

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

Criteria for a good story

Customer objectives

- accessible, understandable

Application

"Do you see it in front of you?"

Customer objectives

- valuable, appealing

Application

attractive, important

"Are customers queuing up for this?"

Conceptual

- critical, challenging

Realization

"What is difficult in the realization?"

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

Application

- frequent, no exceptional niche

"Does it add significantly to the bottom line?"

Application

- specific

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

Functional

Example of a story

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

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

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

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



source: Roland Mathijssen
Embedded Systems Institute
Eindhoven

Value and Challenges in this story

Customer
objectives

Application

Value proposition in this story:

quality of life:

active participation in different social settings

usability for nontechnical elderly people:

"intelligent" system is simple to use

loading of batteries

Conceptual

Realization

Challenges in this story:

Intelligent hearing instrument

Battery life —at least 1 week

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

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

Instrument can be adapted to the hearing loss of the user

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

Recognition of sound environments and automatic adaptation (adaptive
filtering)

source: Roland Mathijssen, Embedded Systems Institute, Eindhoven

Module Management Presentation

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

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

`www.gaudisite.nl`

Abstract

This module addresses the presentation of architectural issues to higher 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.

July 20, 2011
status: draft
version: 1.0



Embedded Systems
INSTITUTE

Simplistic Financial Computations for System Architects.

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

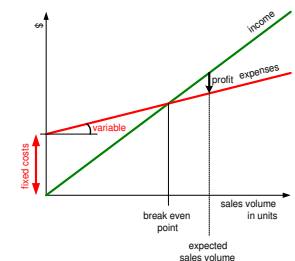
Abstract

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

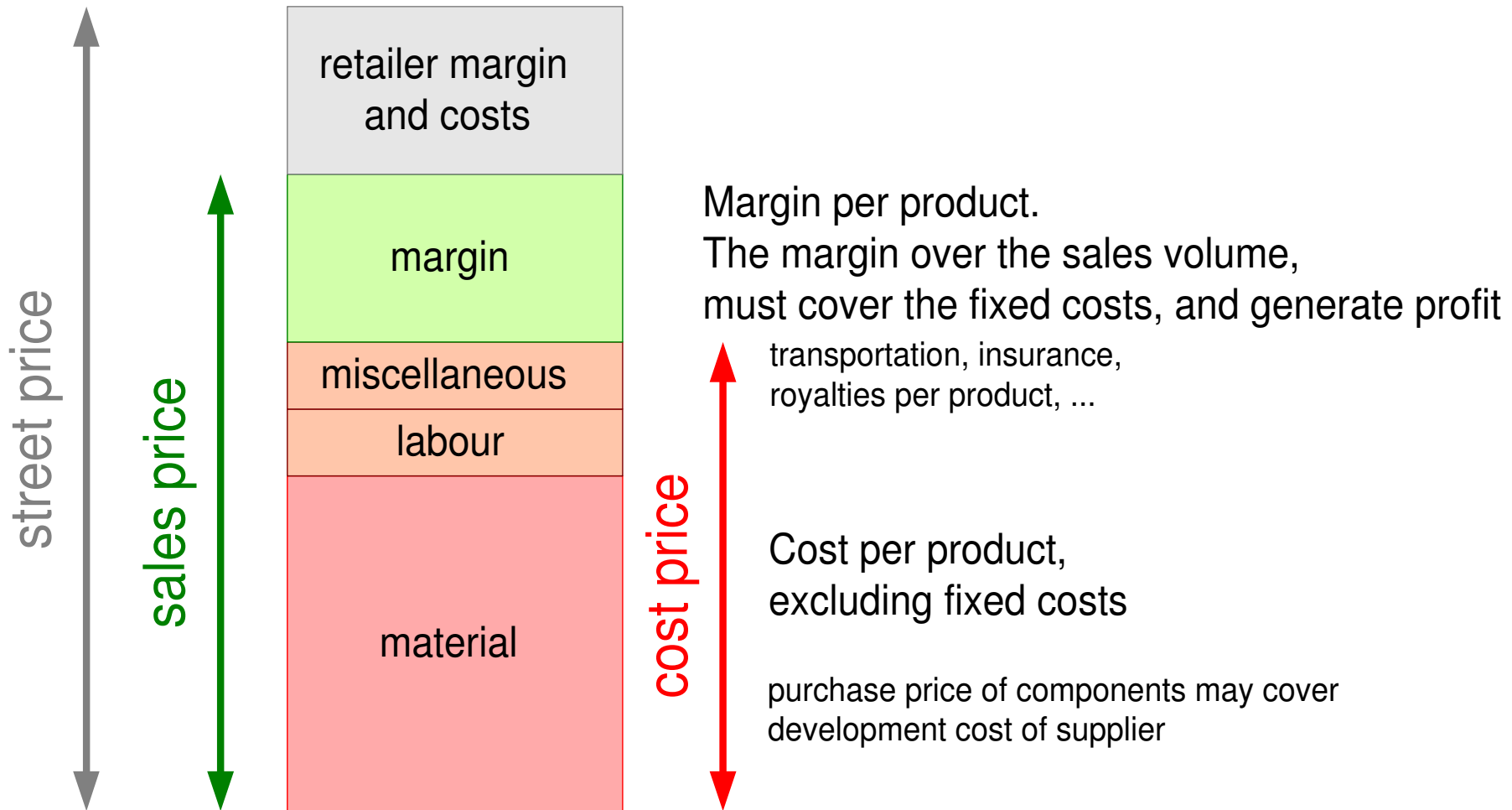
Distribution

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

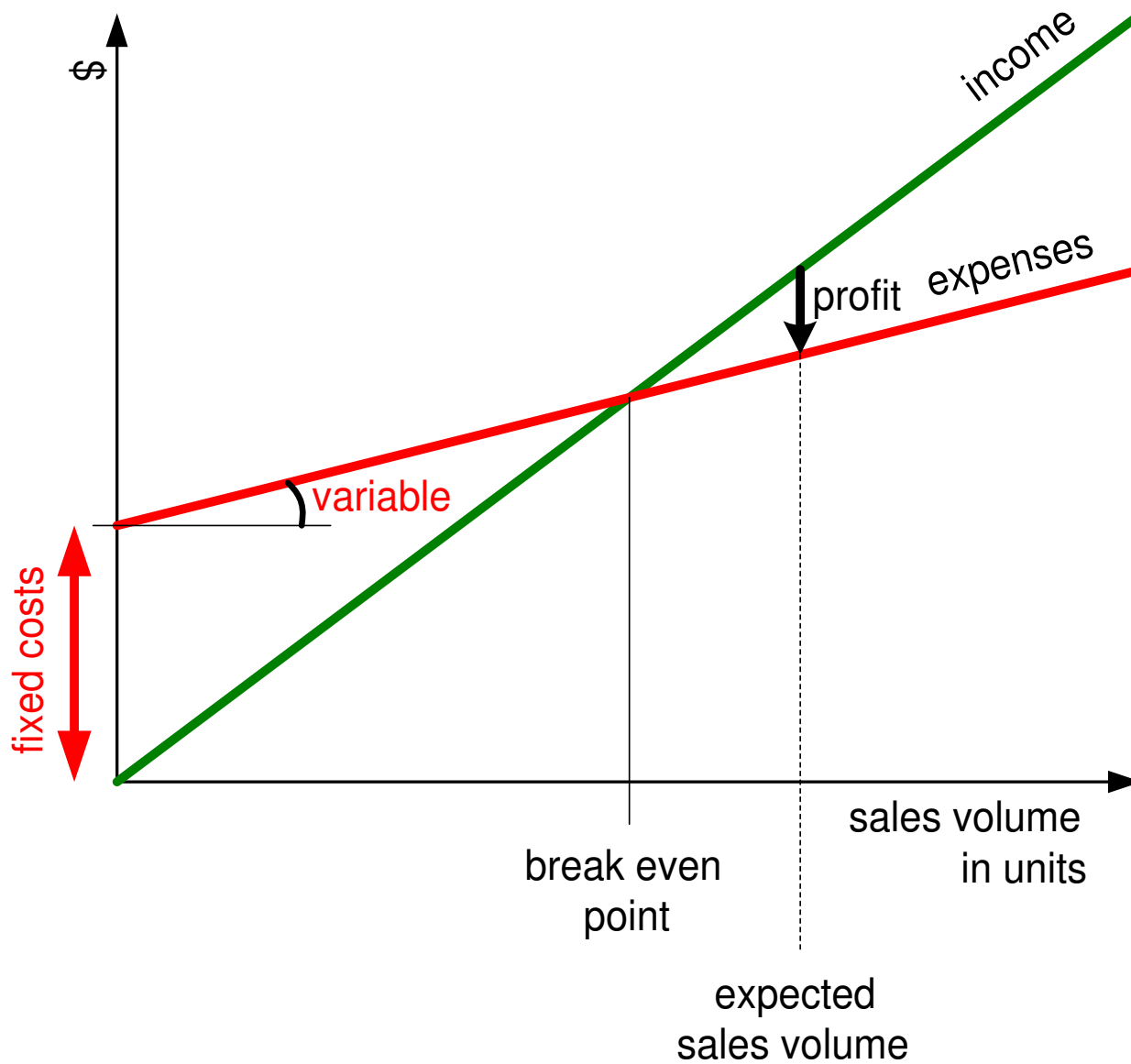
July 20, 2011
status: preliminary
draft
version: 1.2



Product Margin = Sales Price - Cost



Profit as function of sales volume



Investments, more than R&D



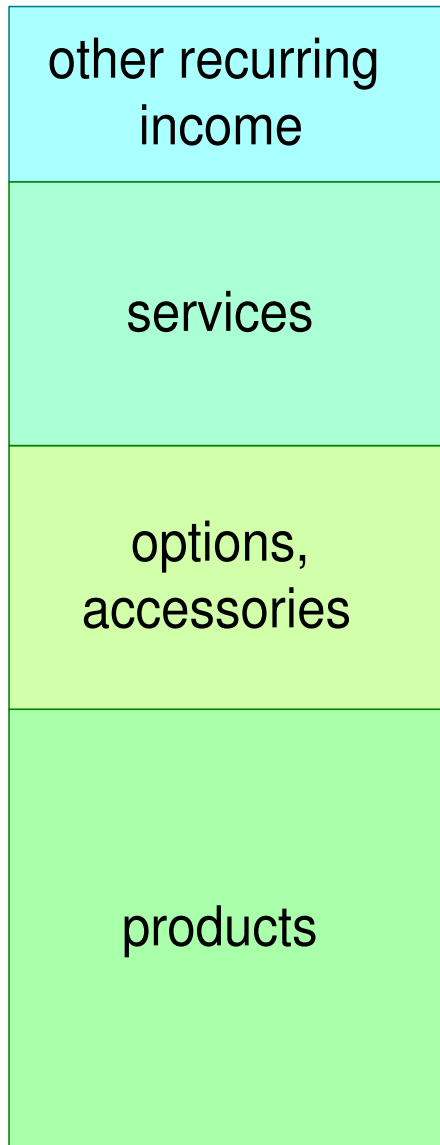
business dependent:
pharmaceutics industry
sales cost >> R&D cost

strategic choice:
NRE or per product

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

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

Income, more than product sales only



license fees
pay per movie

content, portal
updates
maintenance

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

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

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

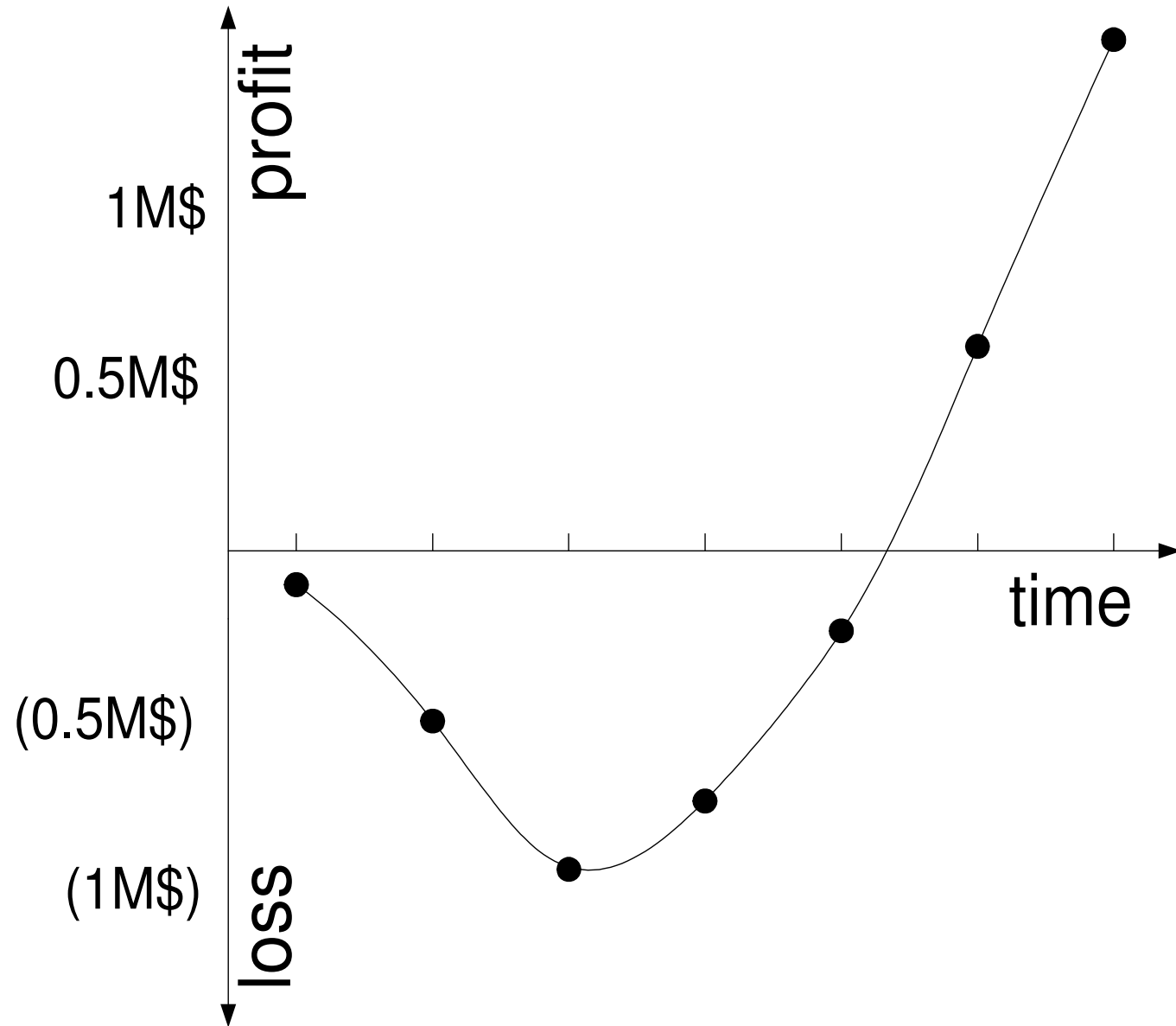
The Time Dimension

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

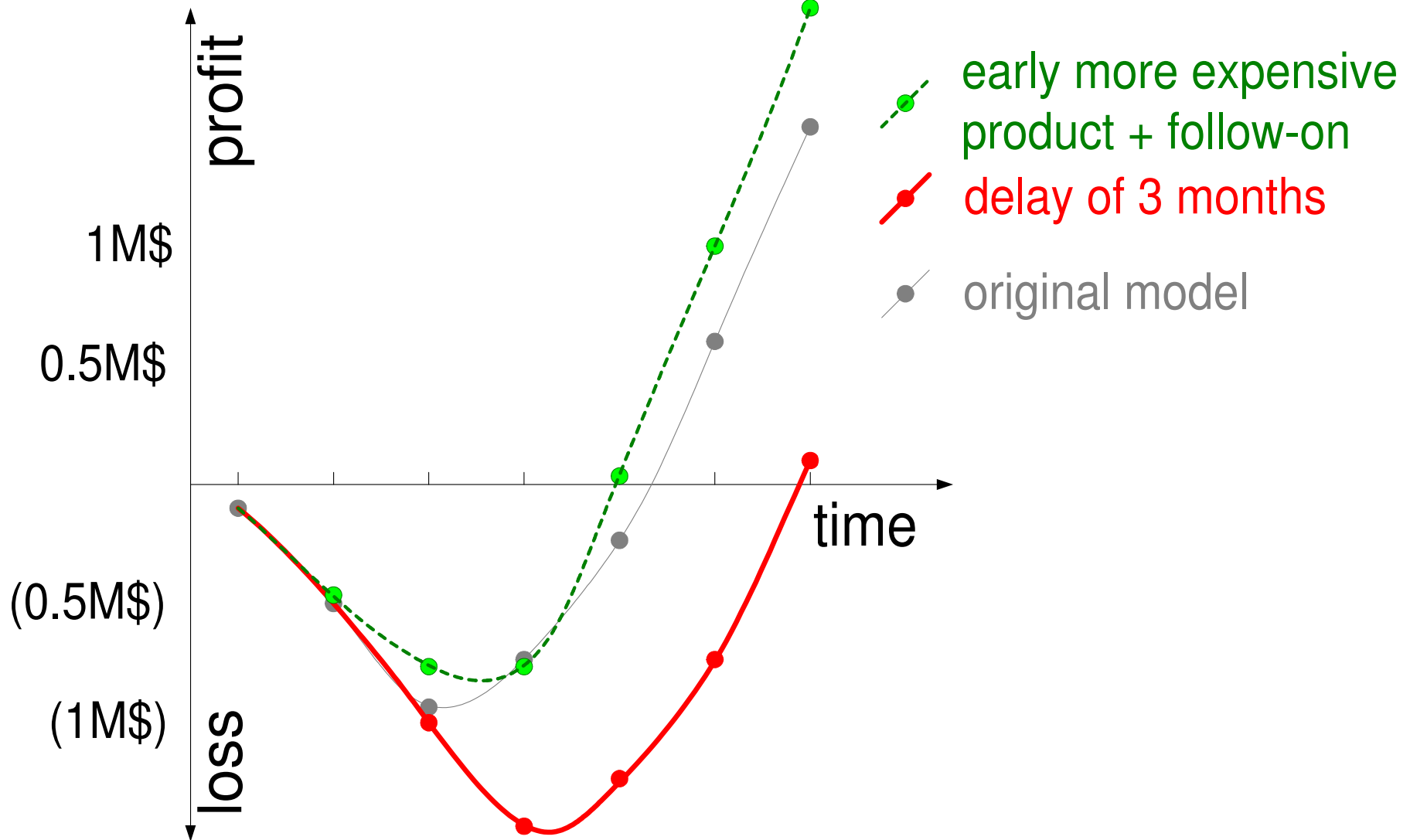
cost price / unit = 20k\$
sales price / unit = 50k\$

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

The "Hockey" Stick



What if ...?



Fashionable financial yardsticks

Return On Investments (ROI)

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

turnover / fte outsourcing reduces headcount, improves this ratio

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

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

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

How to present architecture issues to higher management

by *Gerrit Muller* Buskerud University College

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

`www.gaudisite.nl`

Abstract

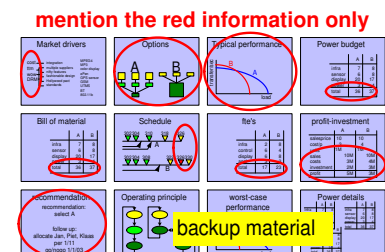
Architects struggle with their visibility at higher management echelons. The introvert nature of architects is a severe handicap. Participation of architects in management teams is important for balanced technical sound decisions and strategy. Improved managerial communication skills of architects are required.

This article describes how to give a more effective presentation to higher management teams. Subjects discussed are the preparation, content and form, do and don't advise.

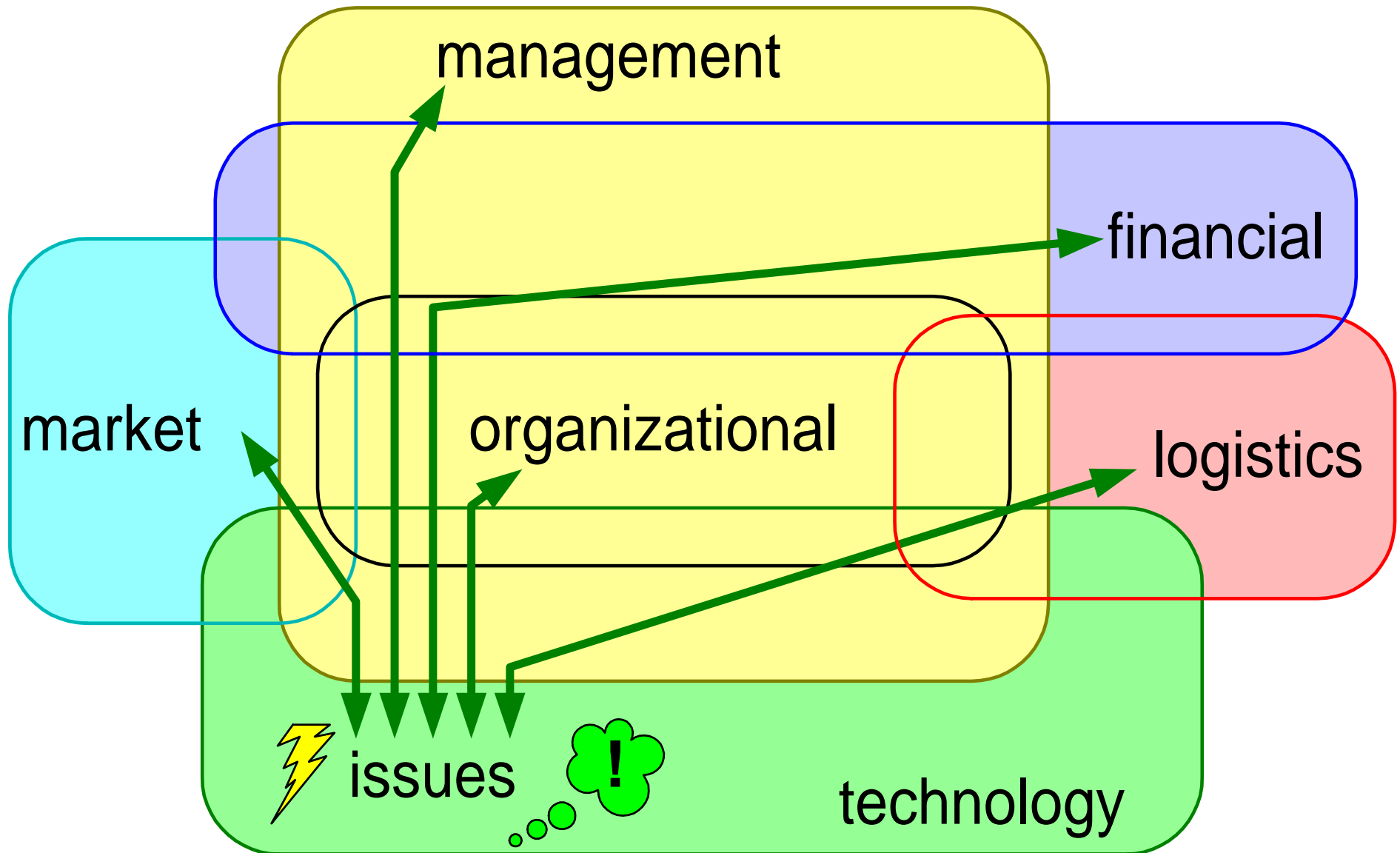
Distribution

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.

July 20, 2011
status: concept
version: 0.1



Architectural issues related to managerial viewpoints



Characteristics of managers in higher management teams

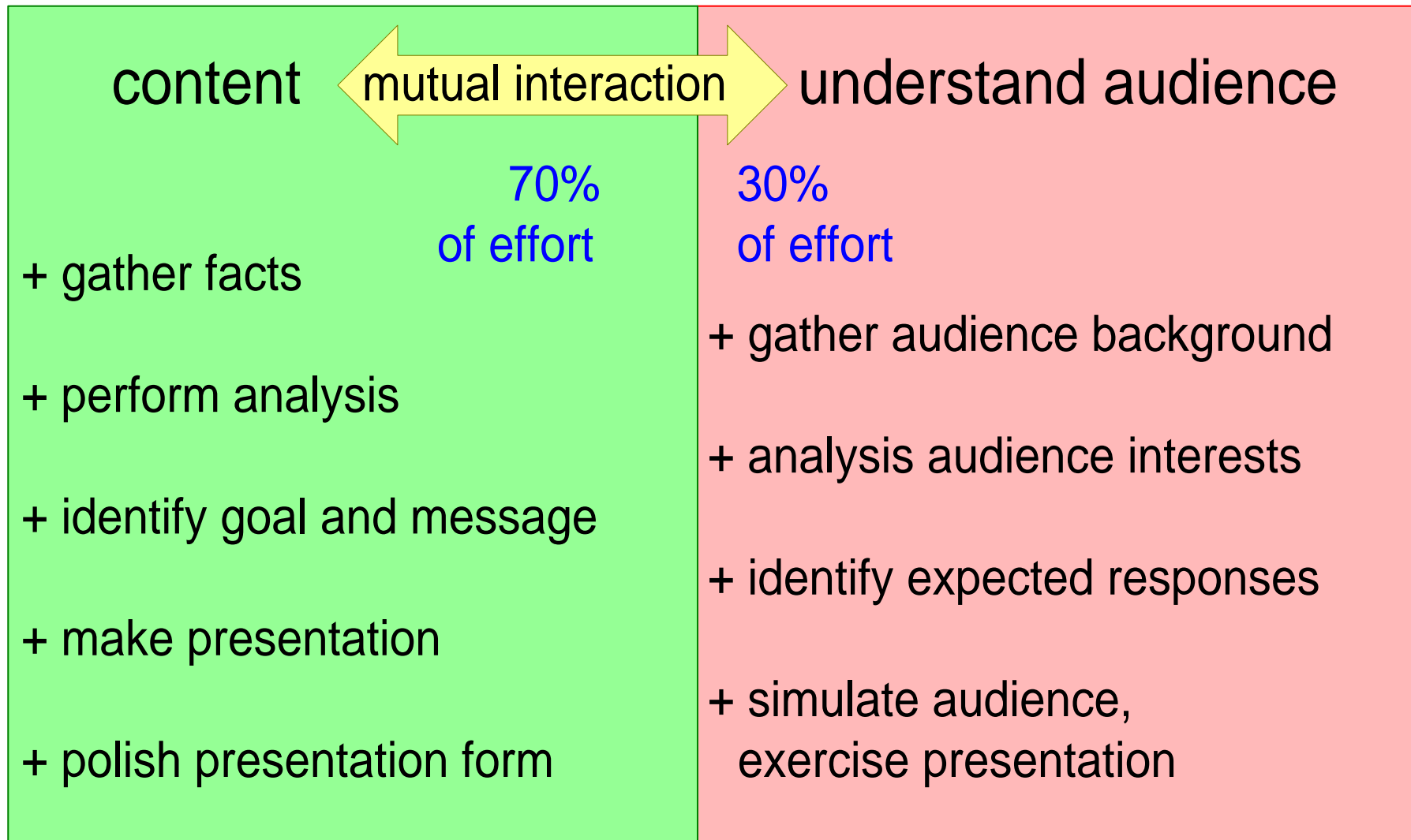
common characteristics

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

highly variable characteristics

- ? technology knowledge
from extensive to shallow
- ? style from power play to
inspirational leadership

Always prepare with small team!



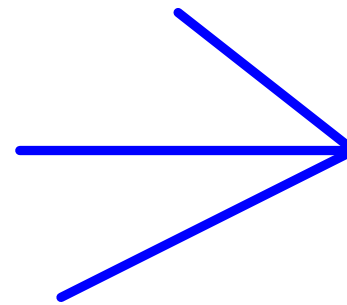
Recommended content

+ clear problem statement (what, why)

+ solution exploration (how)

+ options, recommendations

+ expected actions or decisions



supported by
facts and figures

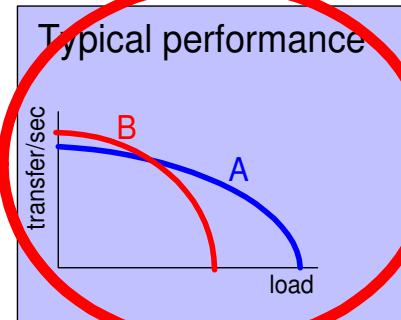
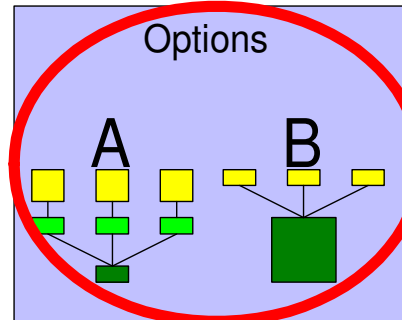
mention the red information only

Market drivers

- Cost
- ttm
- WOW
- DRM

- integration
- multiple suppliers
- nifty features
- fashionable design
- Hollywood pact
- standards

- MPEG4
- MP3
- color display
- ePen
- GPS sensor
- GSM
- UTMS
- BT
- 802.11b

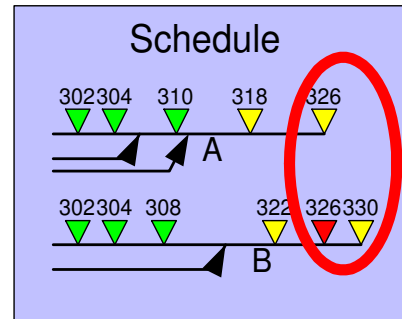


Power budget

	A	B
infra	7	8
sensor	6	8
display	20	17
power	3	4
total	36	37

Bill of material

	A	B
infra	7	8
sensor	6	8
display	20	17
power	3	4
total	36	37



fte's

	A	B
infra	2	8
control	6	4
display	6	8
and	3	3
total	17	23

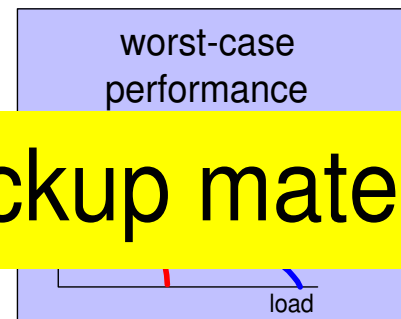
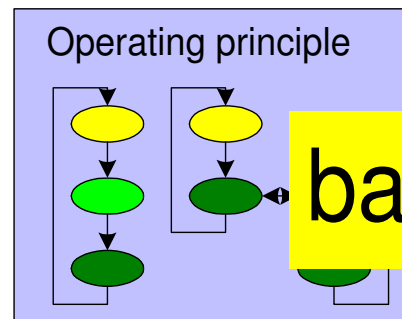
profit-investment

	A	B
salesprice	10	10
cost/p	3	4
units	1M	1M
sales	10M	10M
costs	3M	4M
investment	2M	3M
profit	5M	3M

recommendation

recommendation:
select A

follow up:
allocate Jan, Piet, Klaas
per 1/11
go/nogo 1/1/03



Power details

	A	B
infra	7	8
sensor	6	8
display	20	17
power	3	4
total	36	37

backup material

Form is important

poor form can easily distract from purpose and content

presentation material

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

presenter's appearance

- + well dressed
- + self confident but open

but stay yourself,
stay authentic

Don't force your opinion, understand the audience

do not

- preach beliefs
- underestimate technology knowledge of managers
- tell them what they did wrong
- oversell

do

- + quantify, show figures and facts
- + create faith in your knowledge
- + focus on objectives
- + manage expectations

How to cope with managerial dominance

do not

- let one of the managers hijack the meeting
- build up tensions by withholding facts or solutions
- be lost or panic at unexpected inputs or alternatives

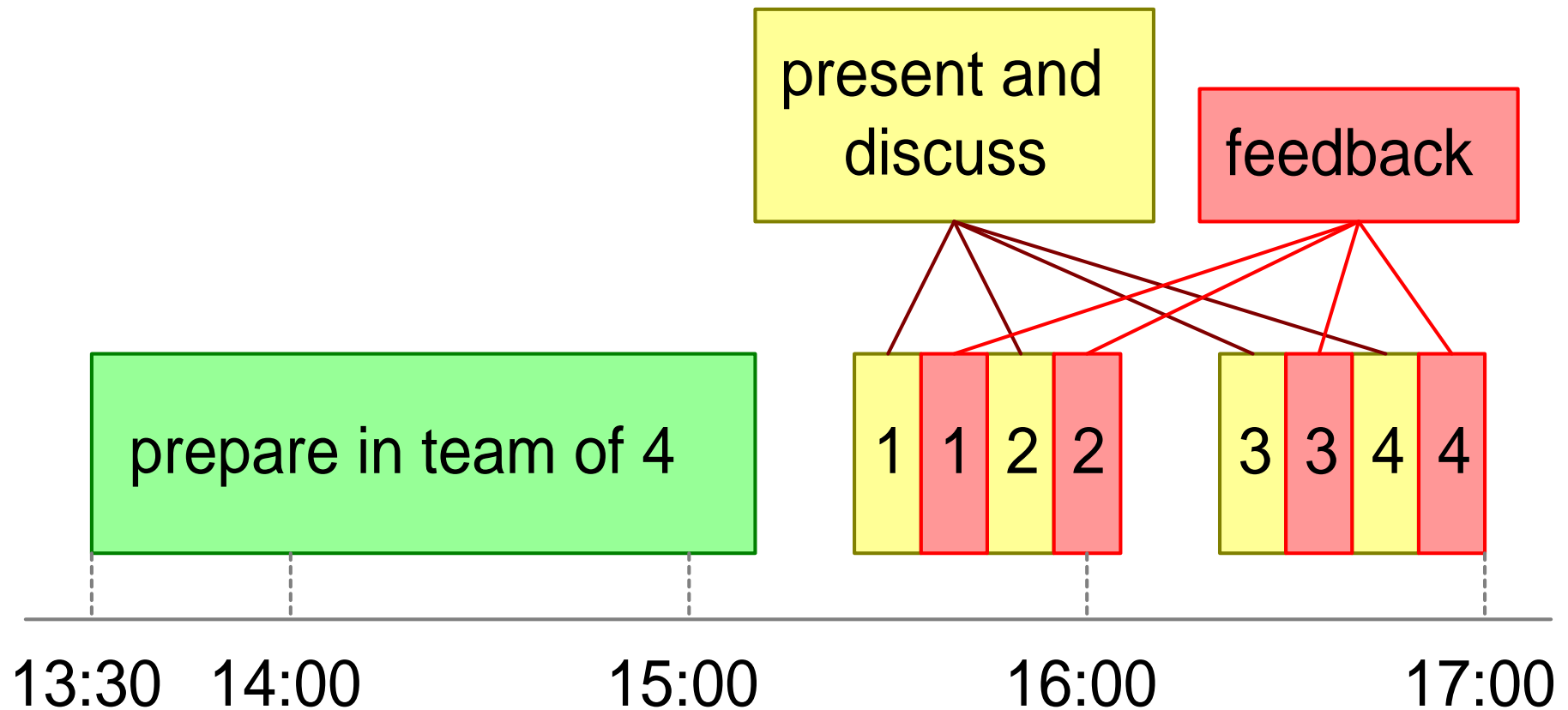
do

- + maintain the lead
- + be to the point and direct
- + acknowledge input, indicate consequences (facts based)

Exercise presentation to higher management

- + Bring a clear **architecture message** to
- + a **Management team** at least 2 hierarchical levels higher
- + with **10 minutes** for **presentation including discussion**
(no limitation on number of slides)
- * architecture message =
technology options in relation with **market/product**
- * address the **concerns** of the **management stakeholders** :
translation required from **technology** issues into
business consequences (months, fte's, turnover, profit, investments)

Exercise schedule



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.

July 20, 2011
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`

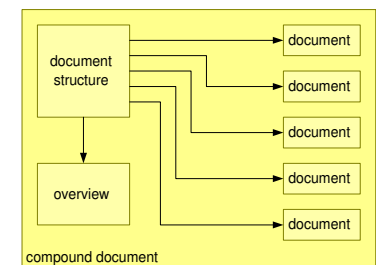
Abstract

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.

July 20, 2011
status: concept
version: 1.2



Requirements for the Entire Documentation Structure

Accessibility for the readers

Low threshold for the readers

Low threshold for the authors

Completeness

Consistency

Maintainability

Scalability

Evolvability

Process to ensure the quality of the information

Requirements from Reader Point of View

Convenient *easy*
 fast

viewing

printing

searching

Requirements per Document

High cohesion (within the unit)

Low coupling (outside of the unit)

Accessibility for the readers

Low threshold for the reader

Low threshold for the author

Manageable steps to create, review, and change

Clear responsibilities

Clear position and relation with the context

Well-defined status of the information

Timely availability

Accessibility Requirements

Ease of reading, “juiciness”

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

Understandability

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

Reachability in a limited number of steps

Responsibility Requirements

single author

limited amount of reviewers

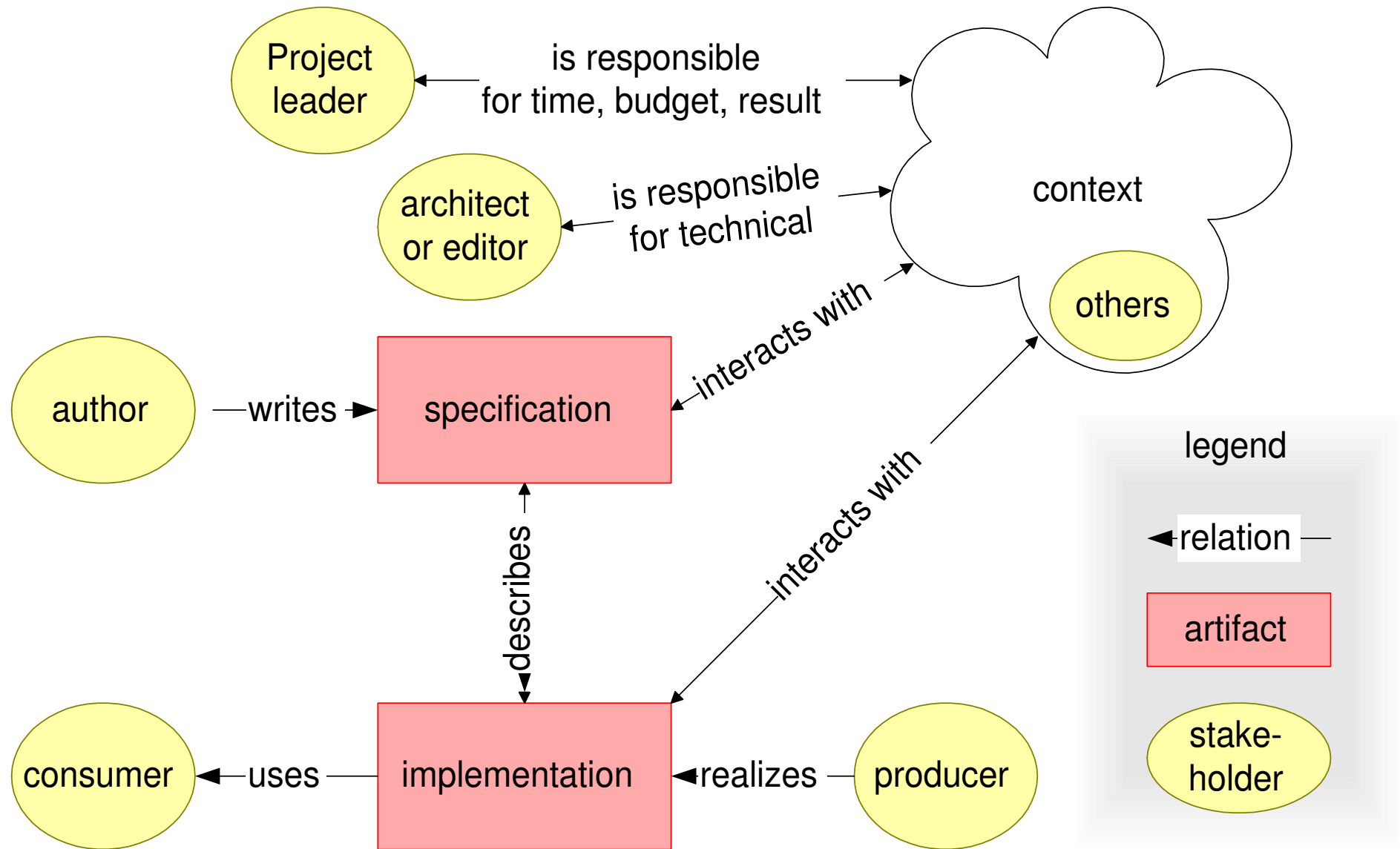
well defined documentation structure

overview specifications at higher
aggregation levels

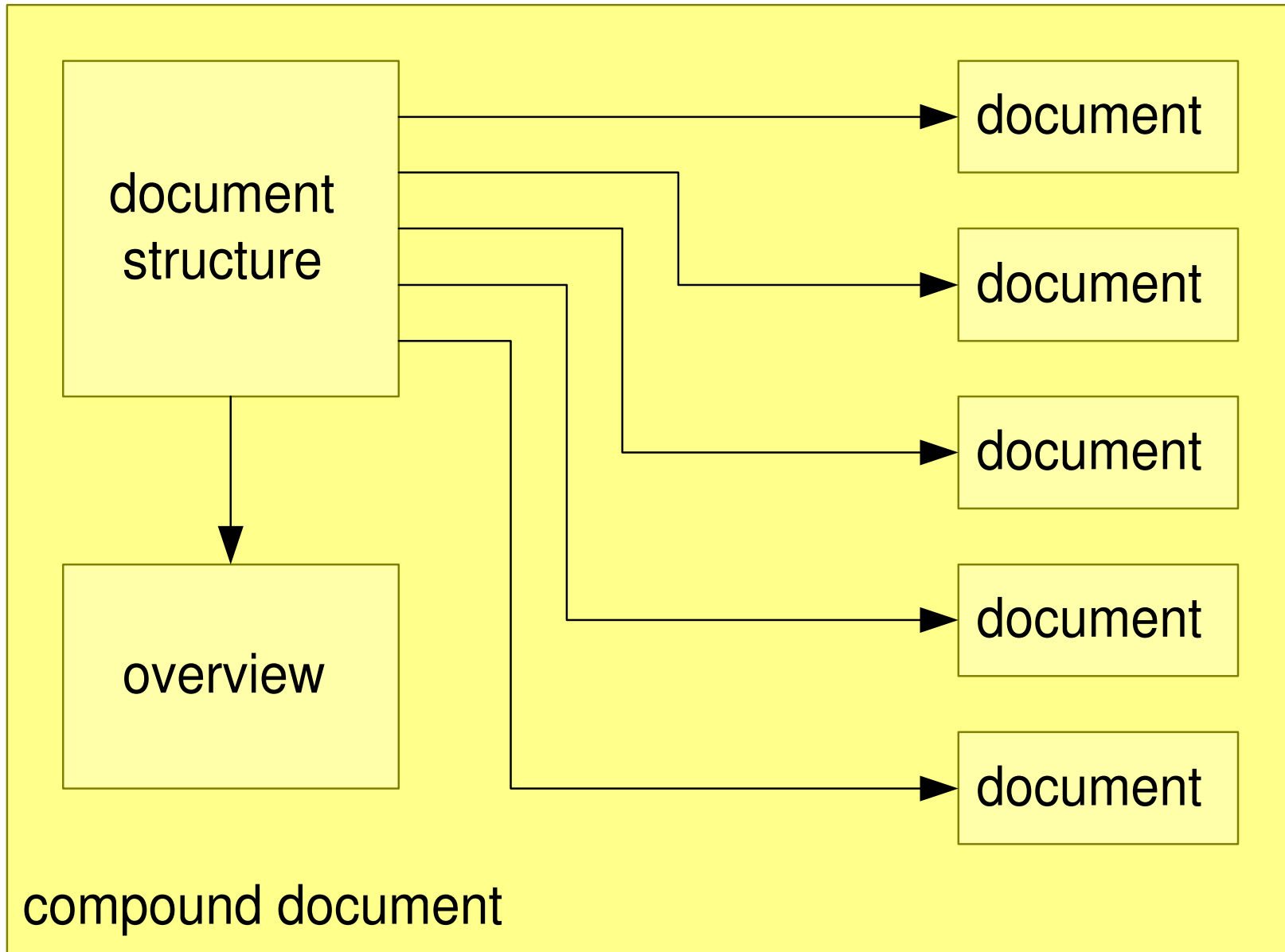
recursive application of structure and
overview

delegation of review process

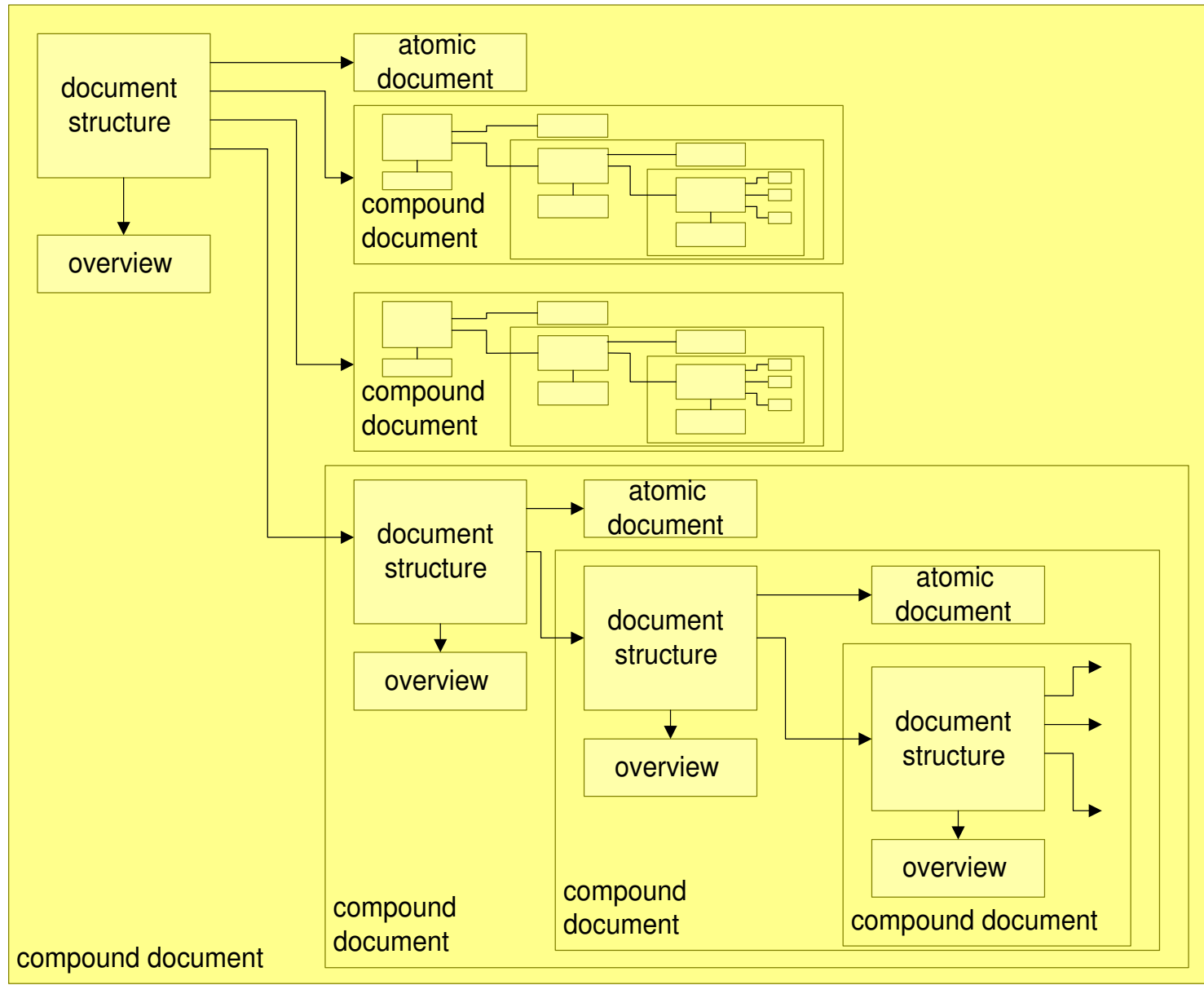
The Stakeholders of a Single Document



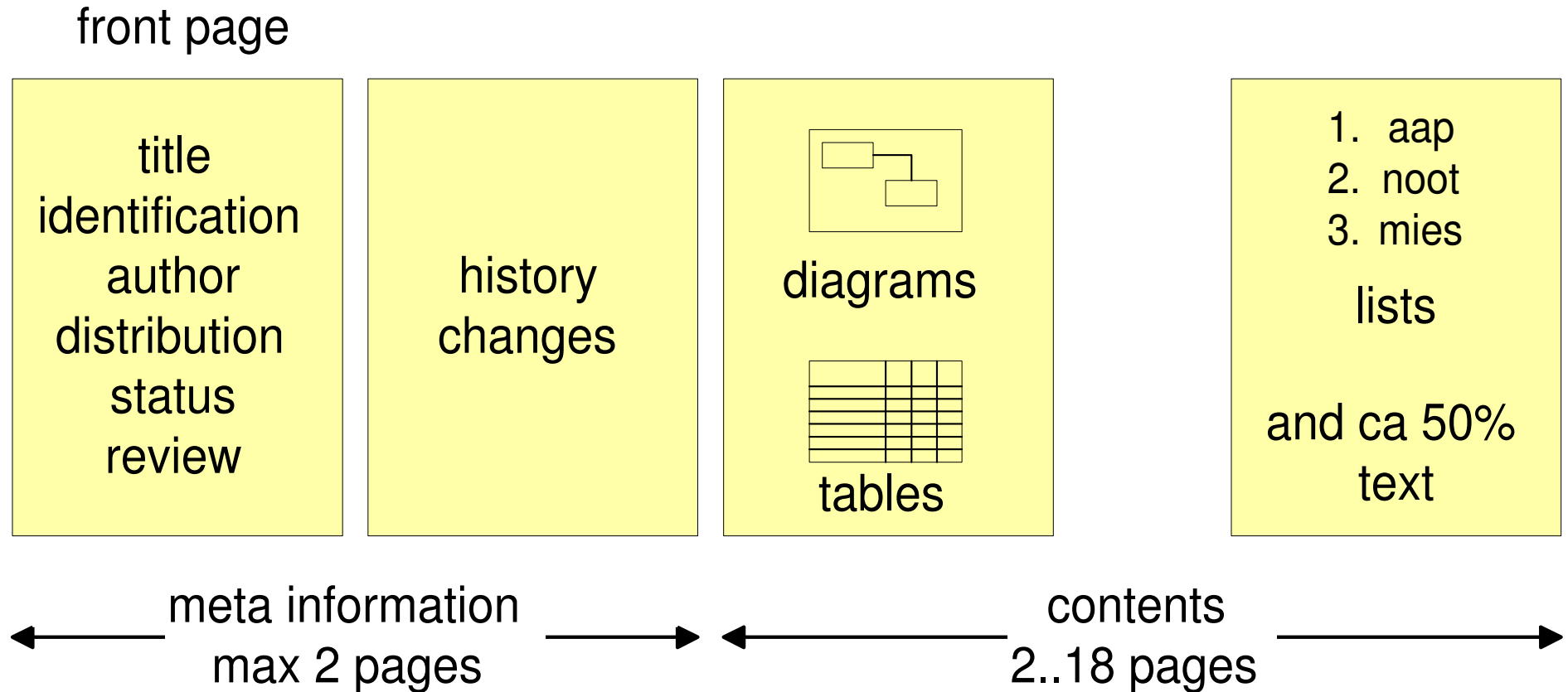
Decomposition of Large Documents



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`

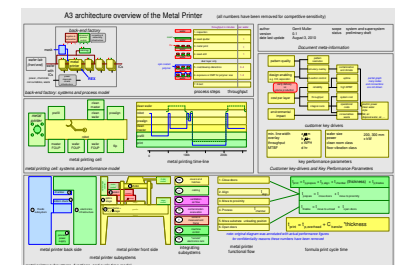
Abstract

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.

July 20, 2011
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.

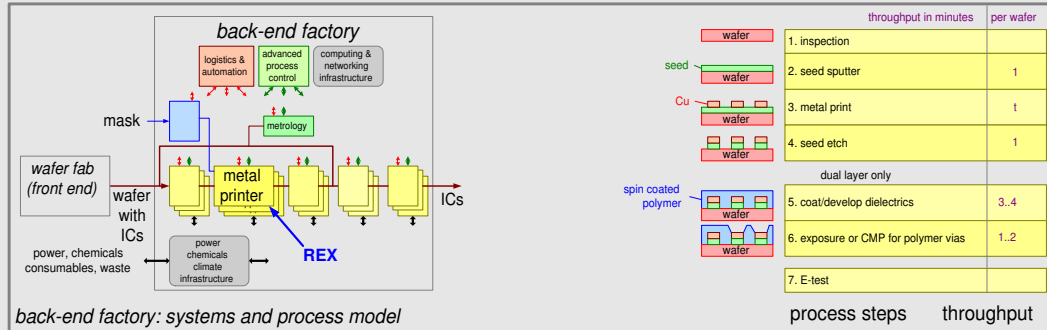
Continuous improvement and learning as cultural value .

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

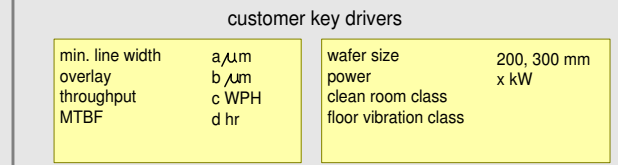
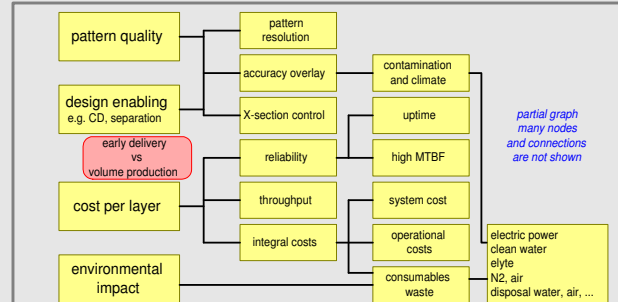
(all numbers have been removed for competitive sensitivity)



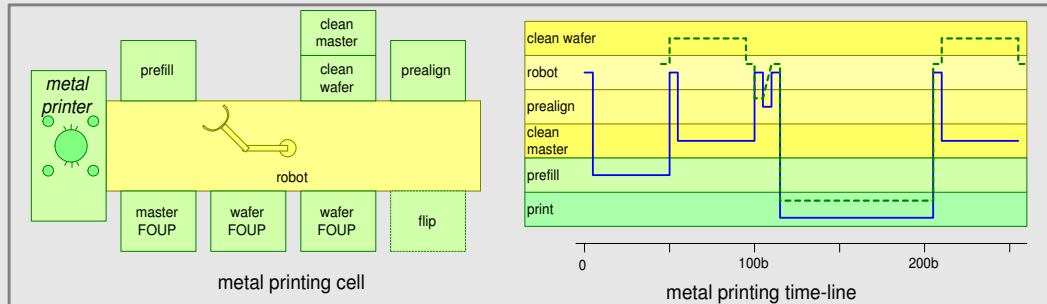
back-end factory: systems and process model

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

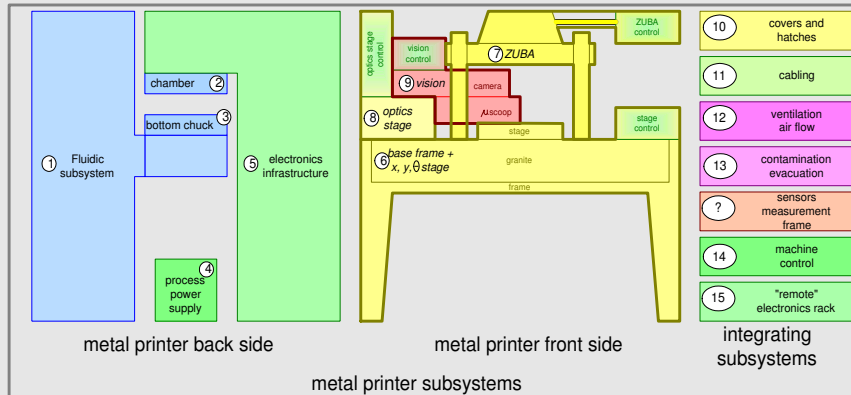
Document meta-information



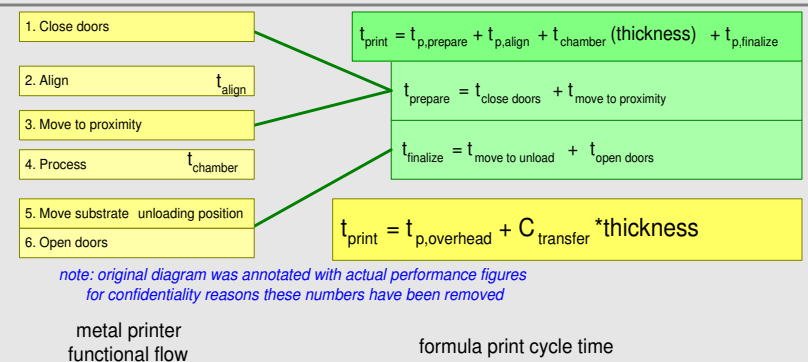
Customer key-drivers and Key Performance Parameters



metal printing cell: systems and performance model



metal printer subsystems, functions, and cycle time model



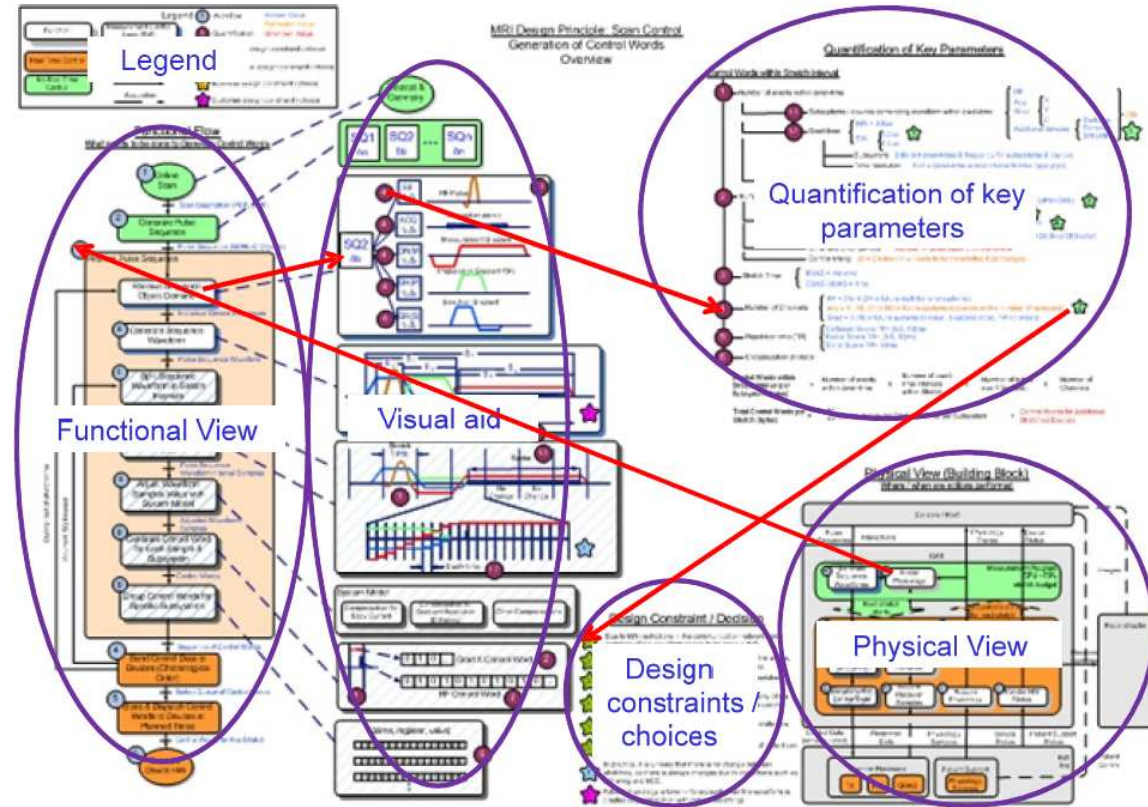
note: original diagram was annotated with actual performance figures for confidentiality reasons these numbers have been removed

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*.

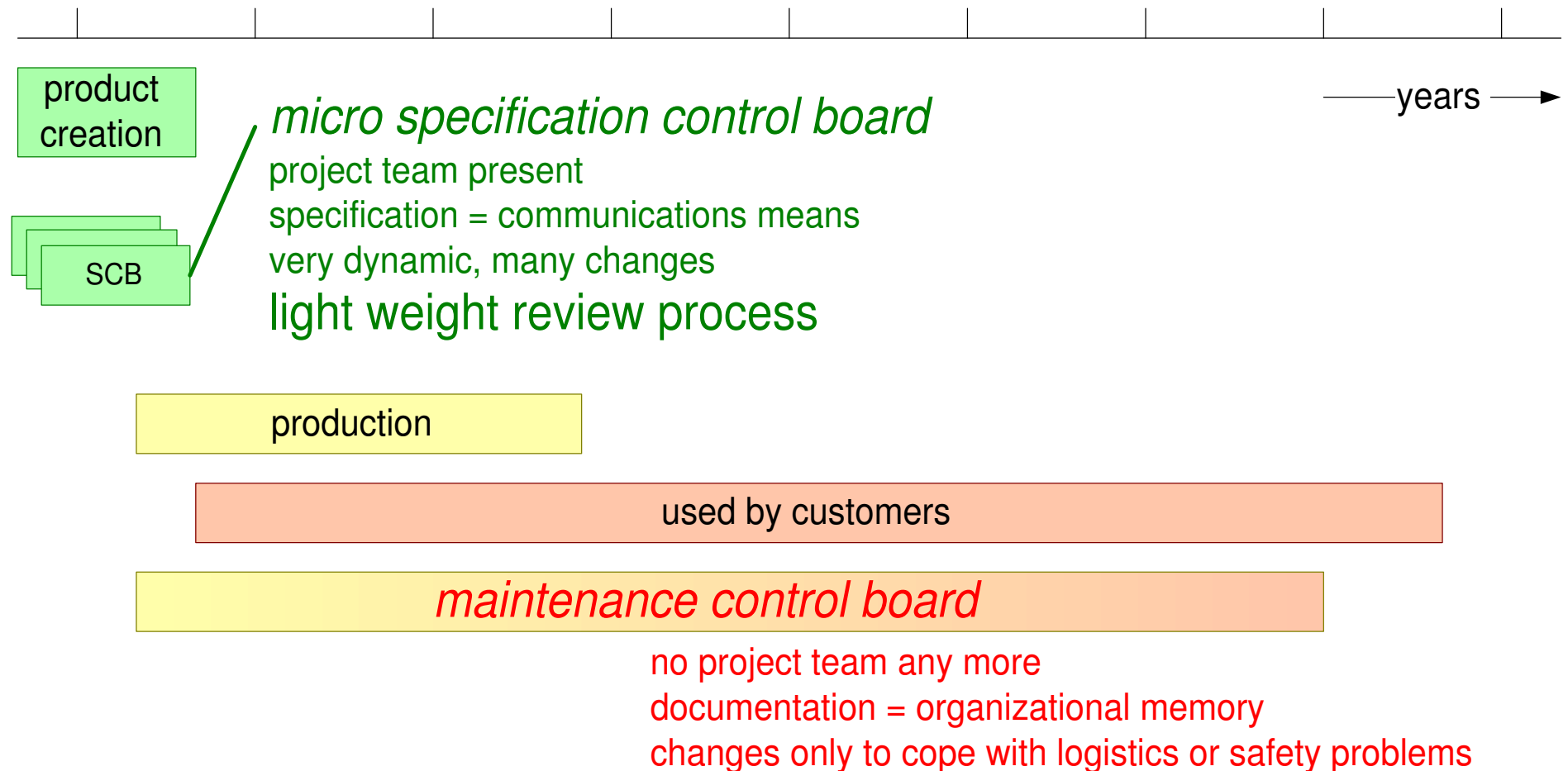
Distribution

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.

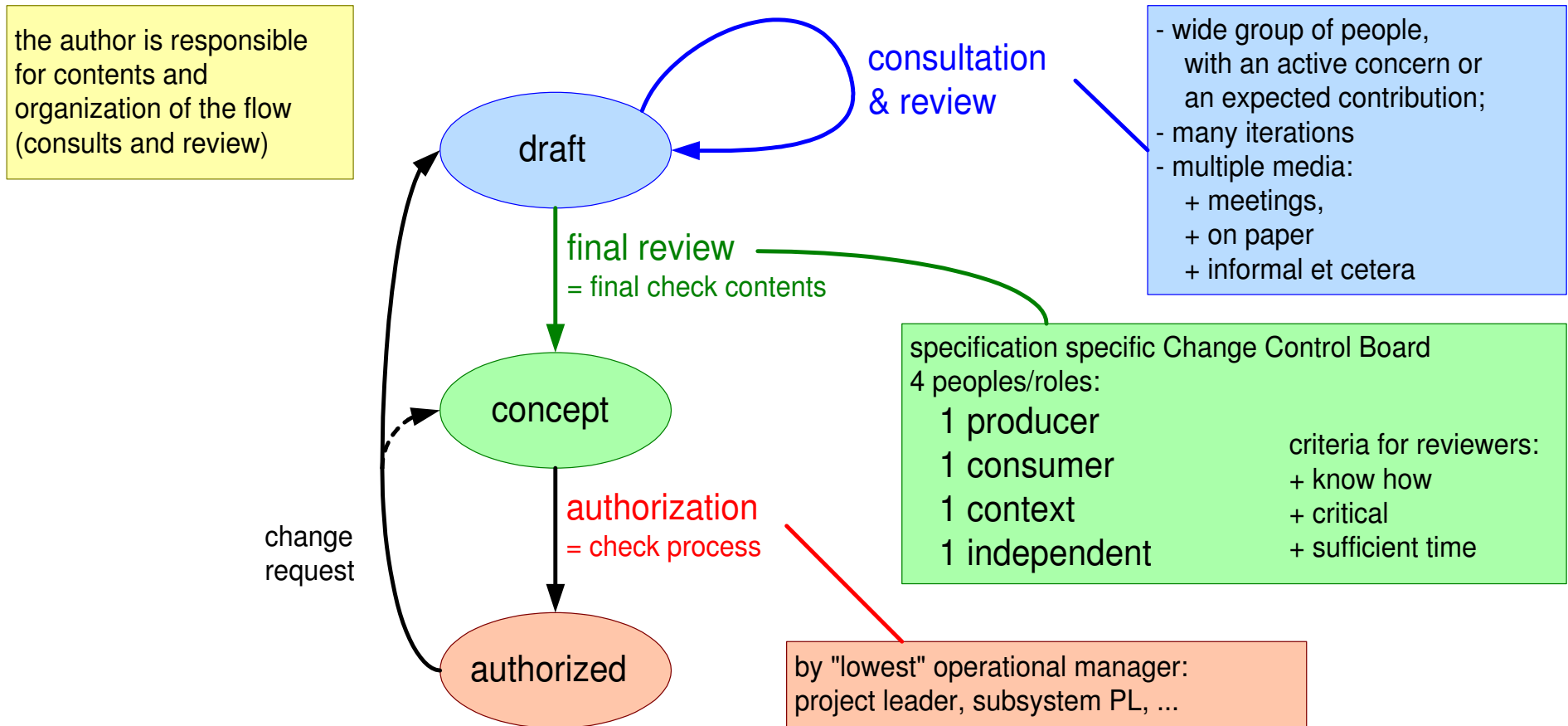
July 20, 2011
status: preliminary
draft
version: 0



Product Life Cycle and Change Management



Light Weight Specification Review Process



Template How To

by *Gerrit Muller* Embedded Systems Institute

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

`www.gaudisite.nl`

Abstract

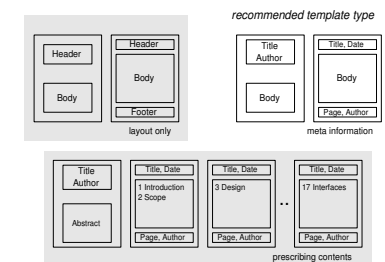
The introduction of a new process (way of working) is quite often implemented by supplying ready-to-go tools and templates. This implementation mainly serves the purpose of a smooth introduction of the new process.

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.

July 20, 2011
status: draft
version: 1.6



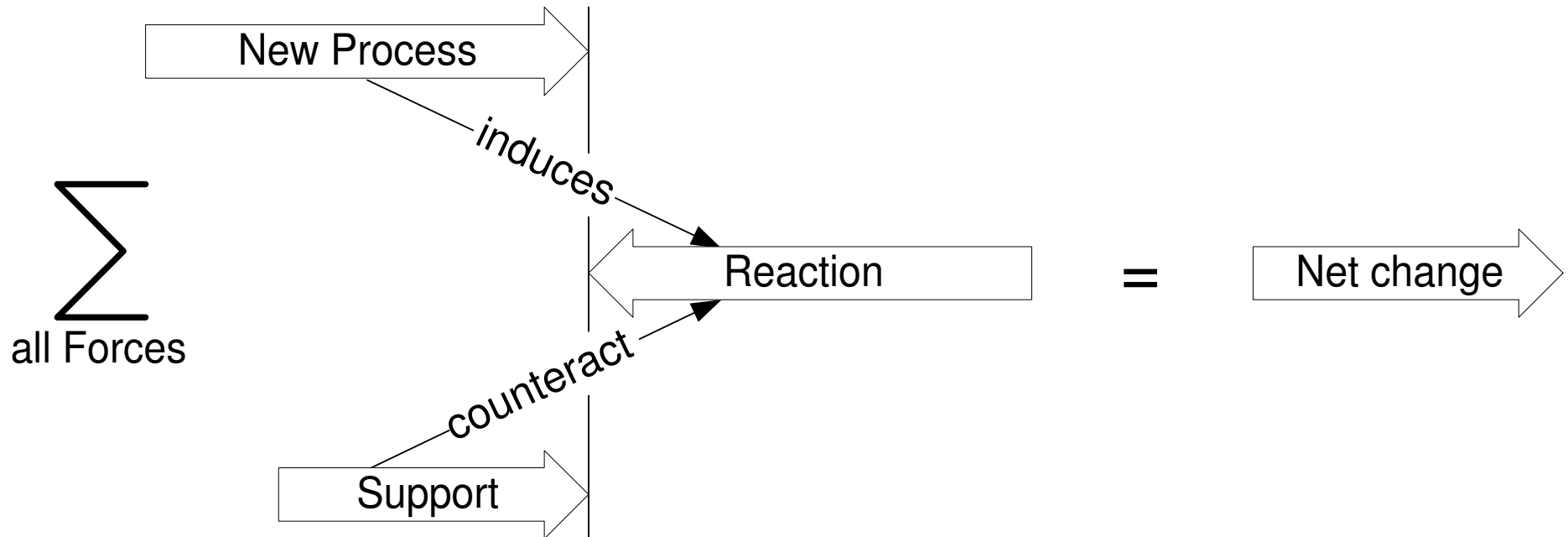
Rationale for Templates

- 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)

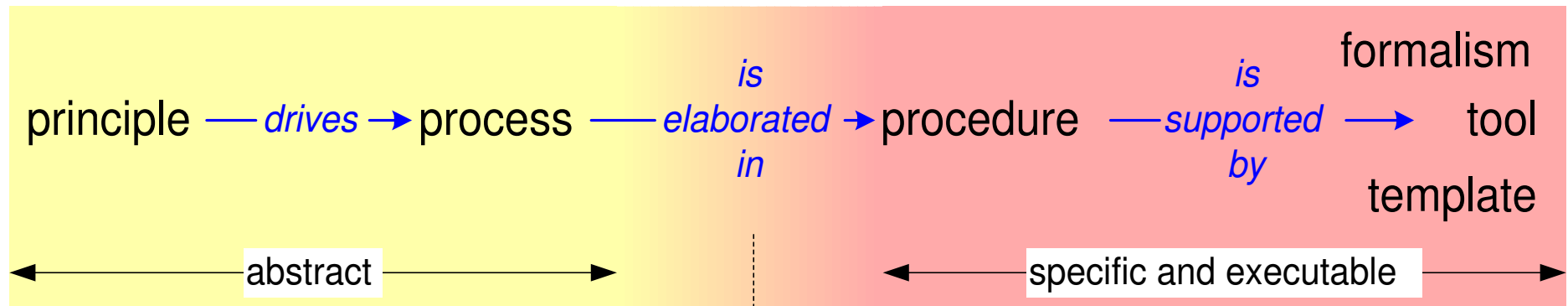
Bogus Arguments for Templates

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

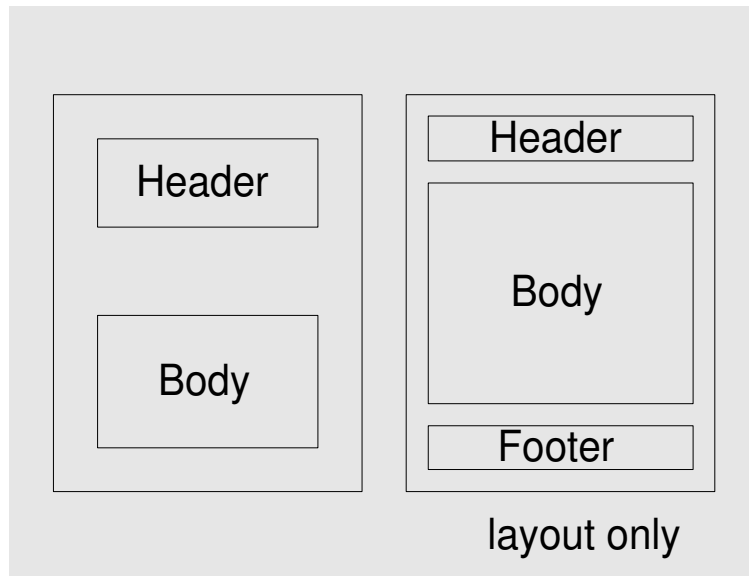
Forces of Change: Action = - Reaction



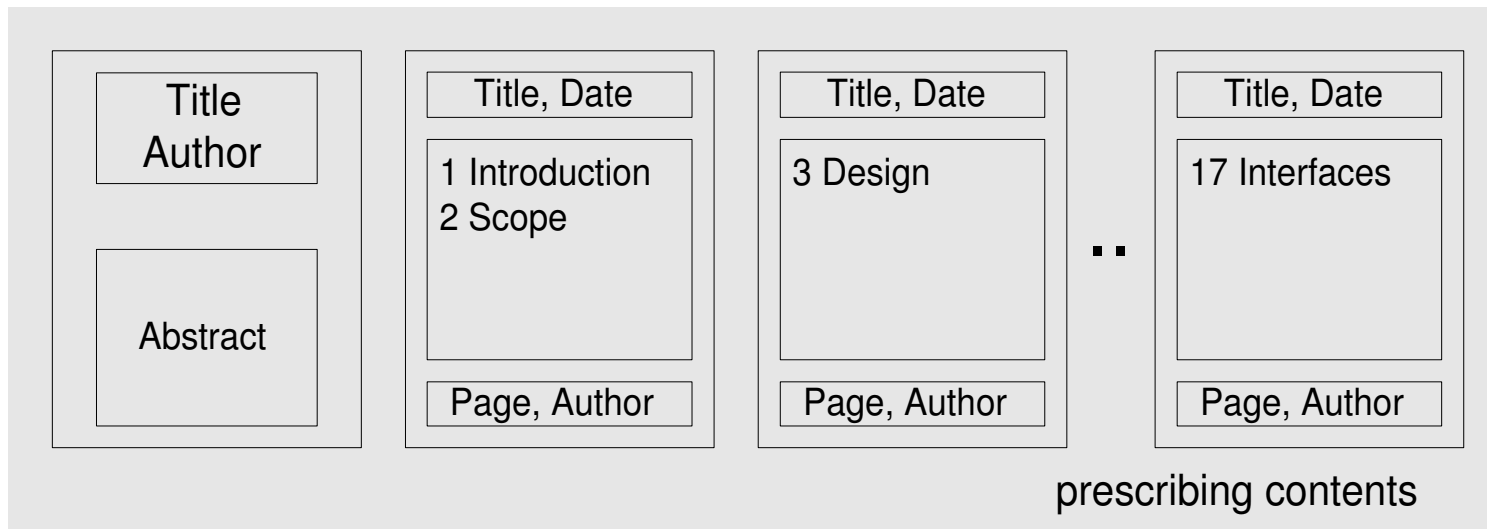
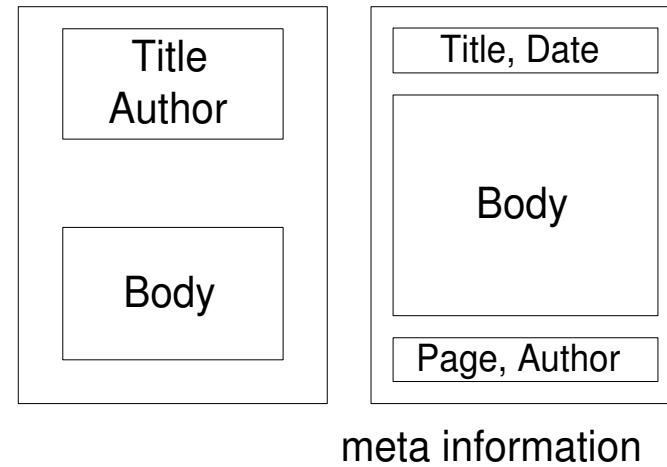
Template as Support for Process



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

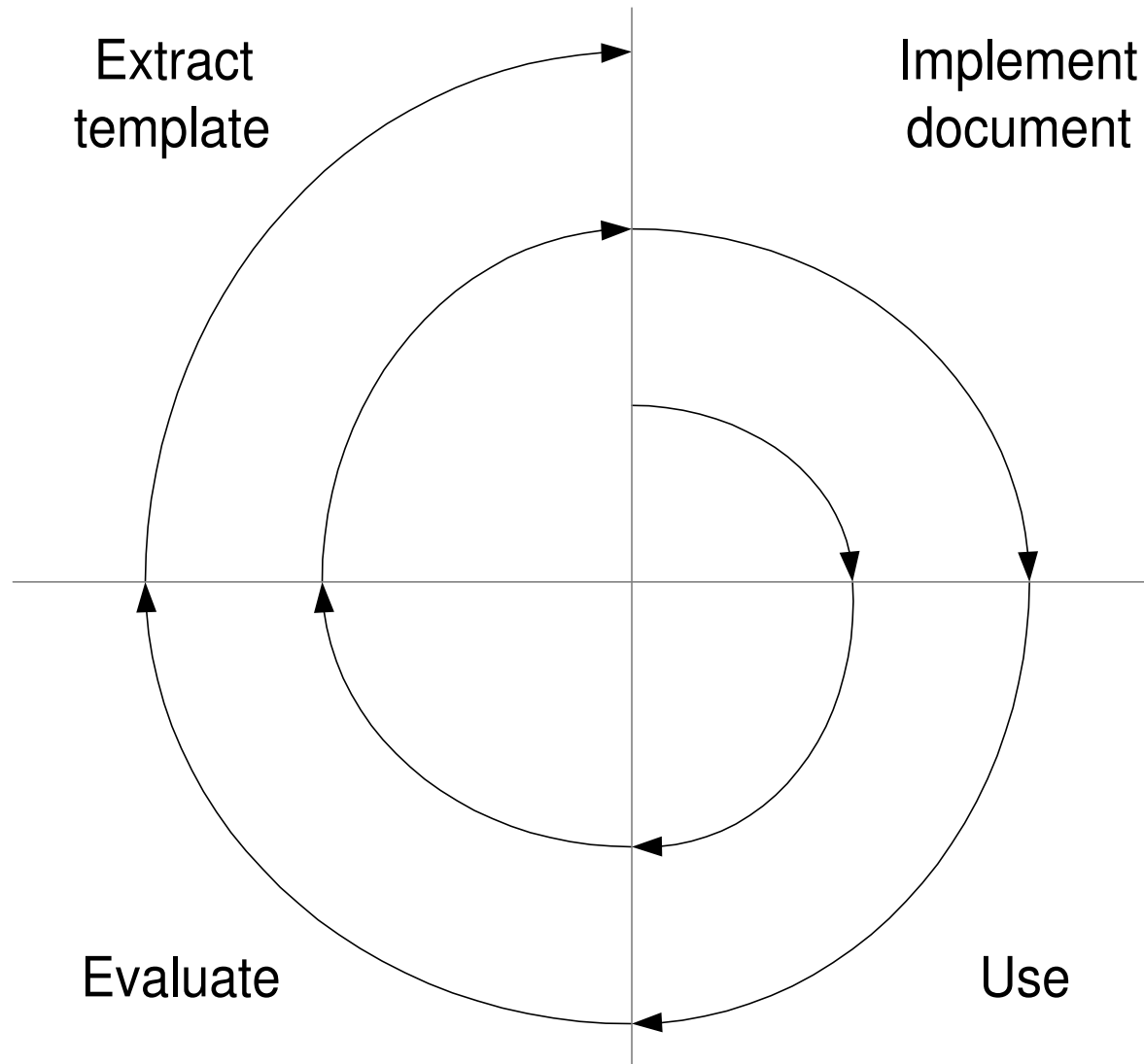
- Use templates for meta-information.
- Use checklists for structure and contents.

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`

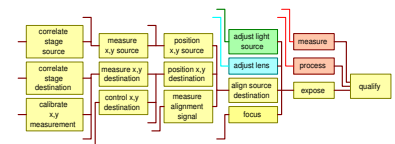
Abstract

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.

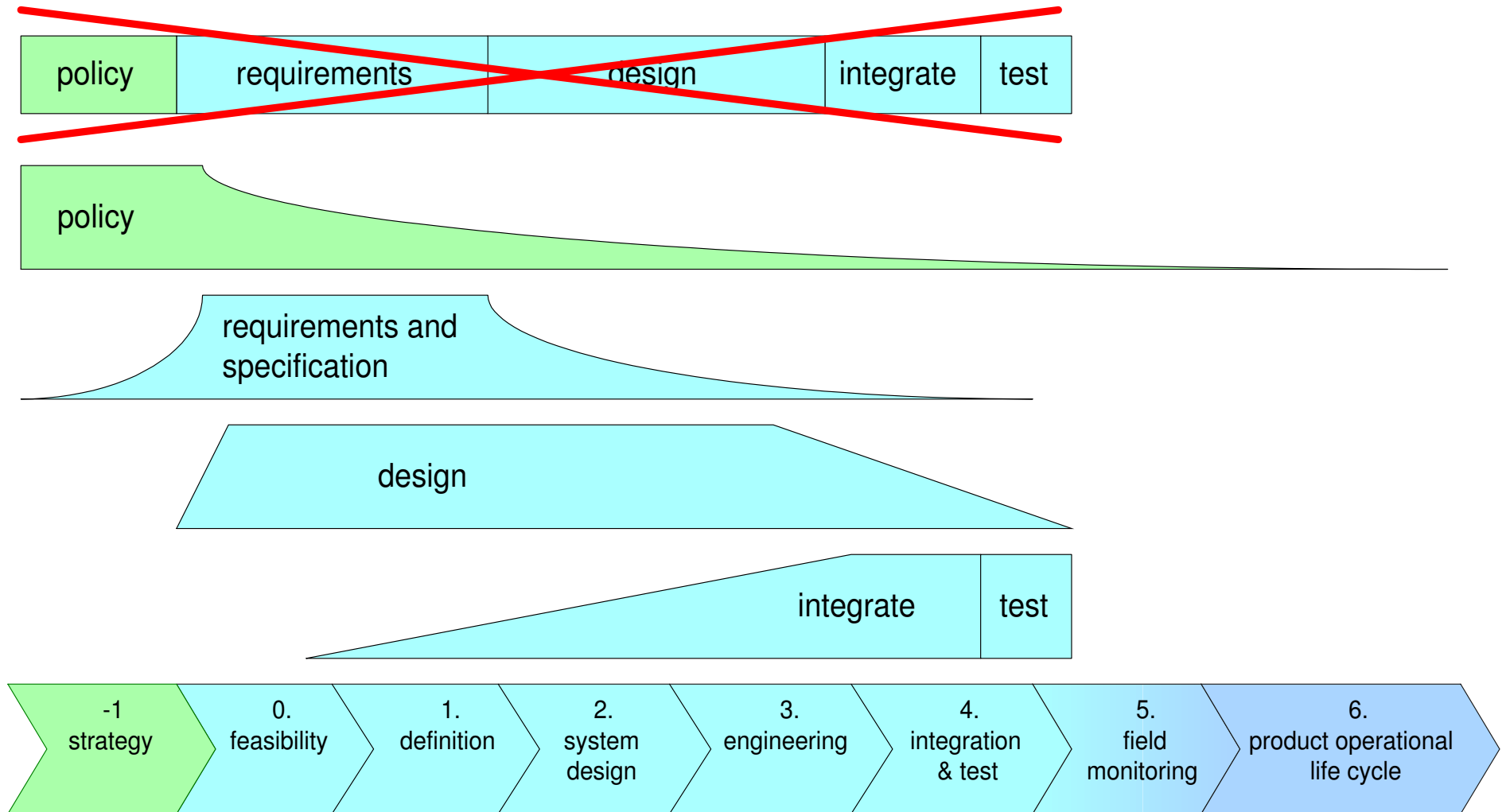
Distribution

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.

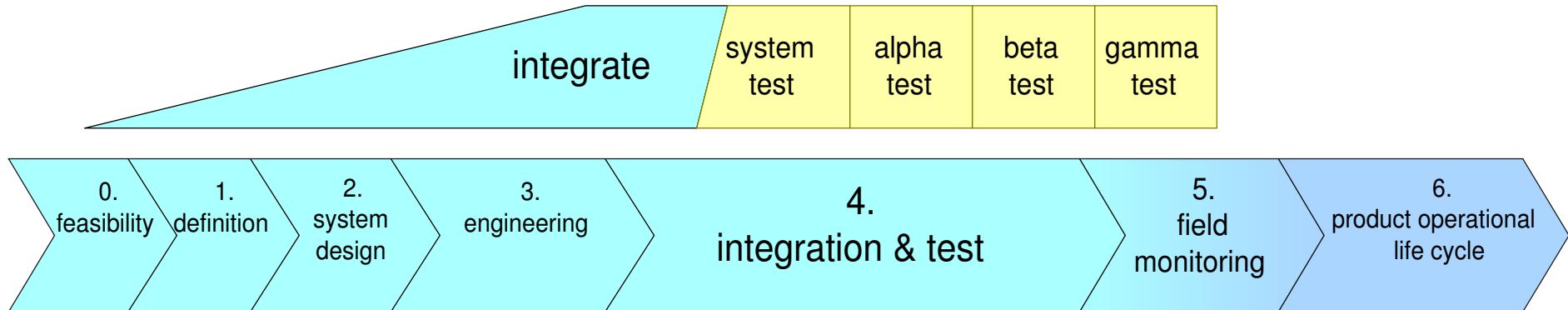
July 20, 2011
status: concept
version: 0.2



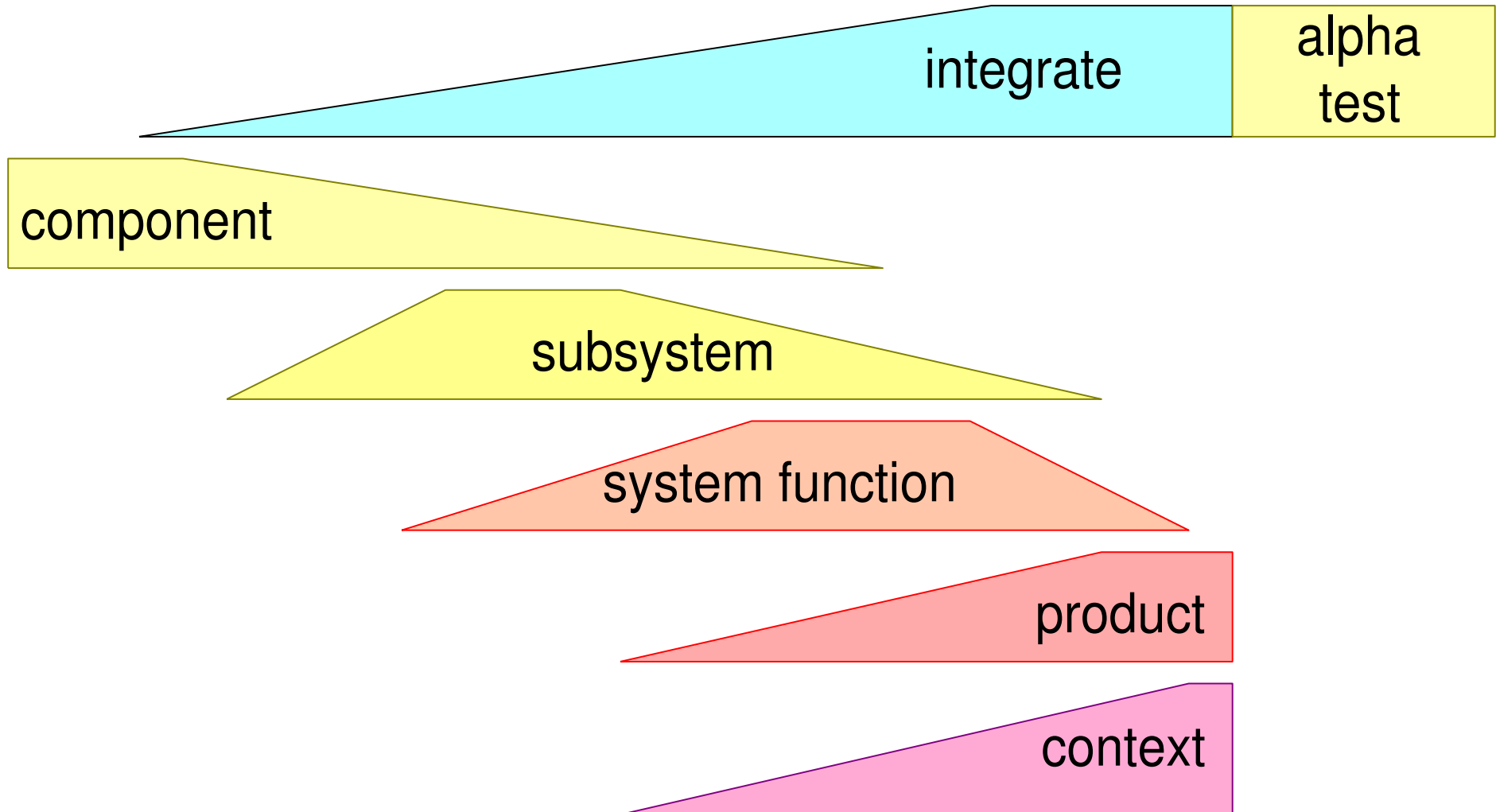
Typical Concurrent Product Creation Process



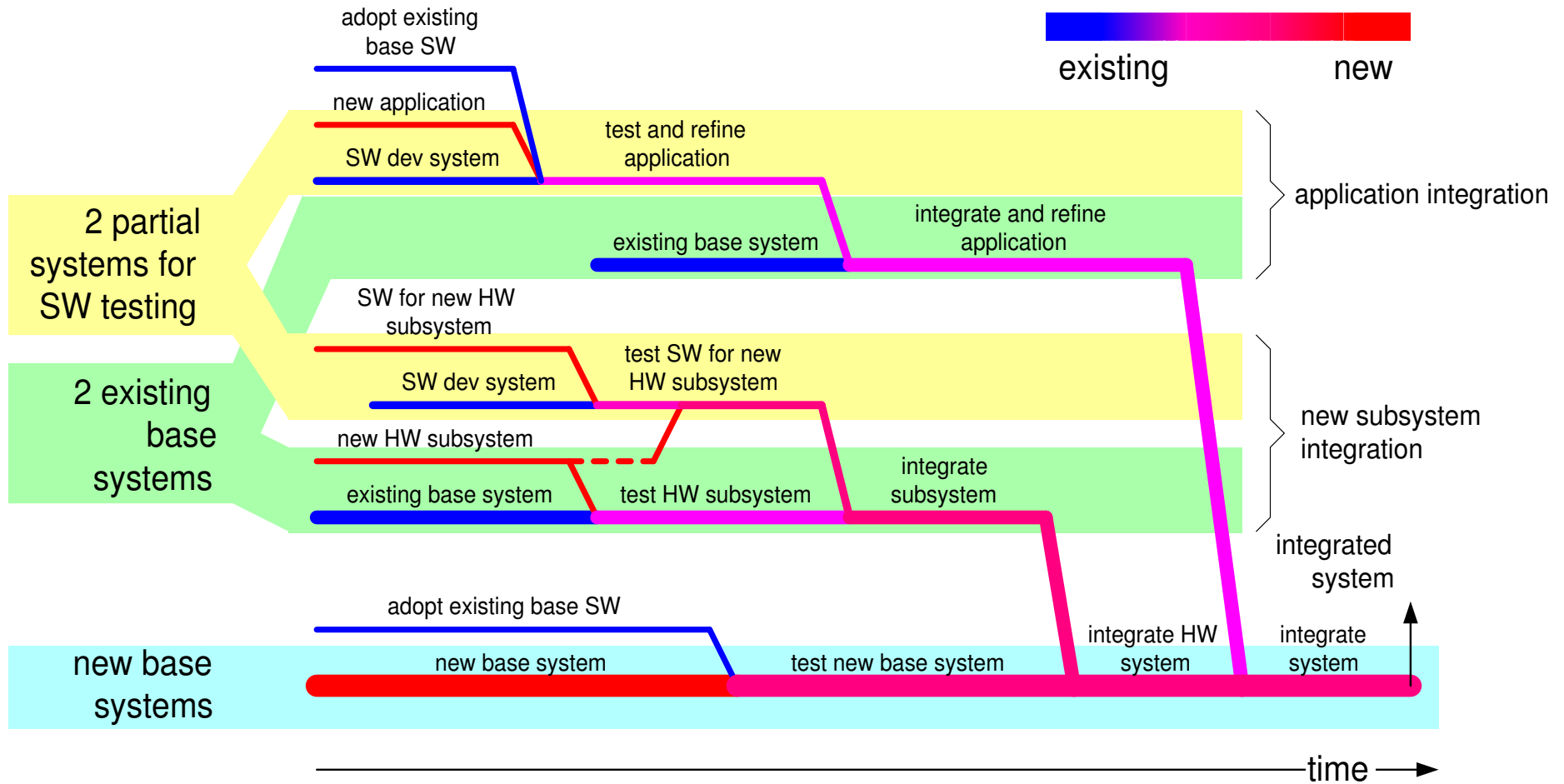
Zooming in on Integration and Tests



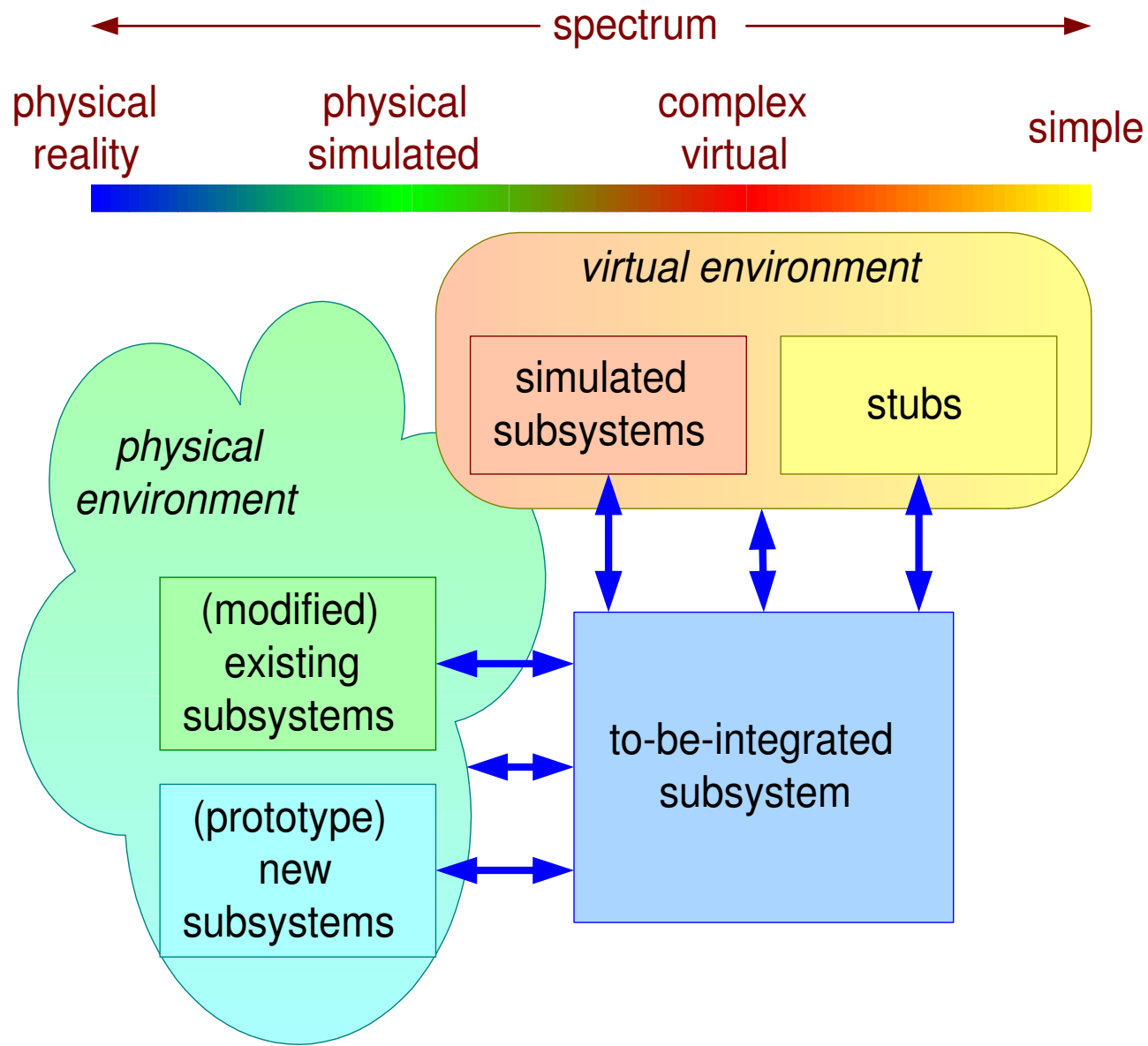
Integration Takes Place in a Bottom-up Fashion



Transition from Previous System to New System



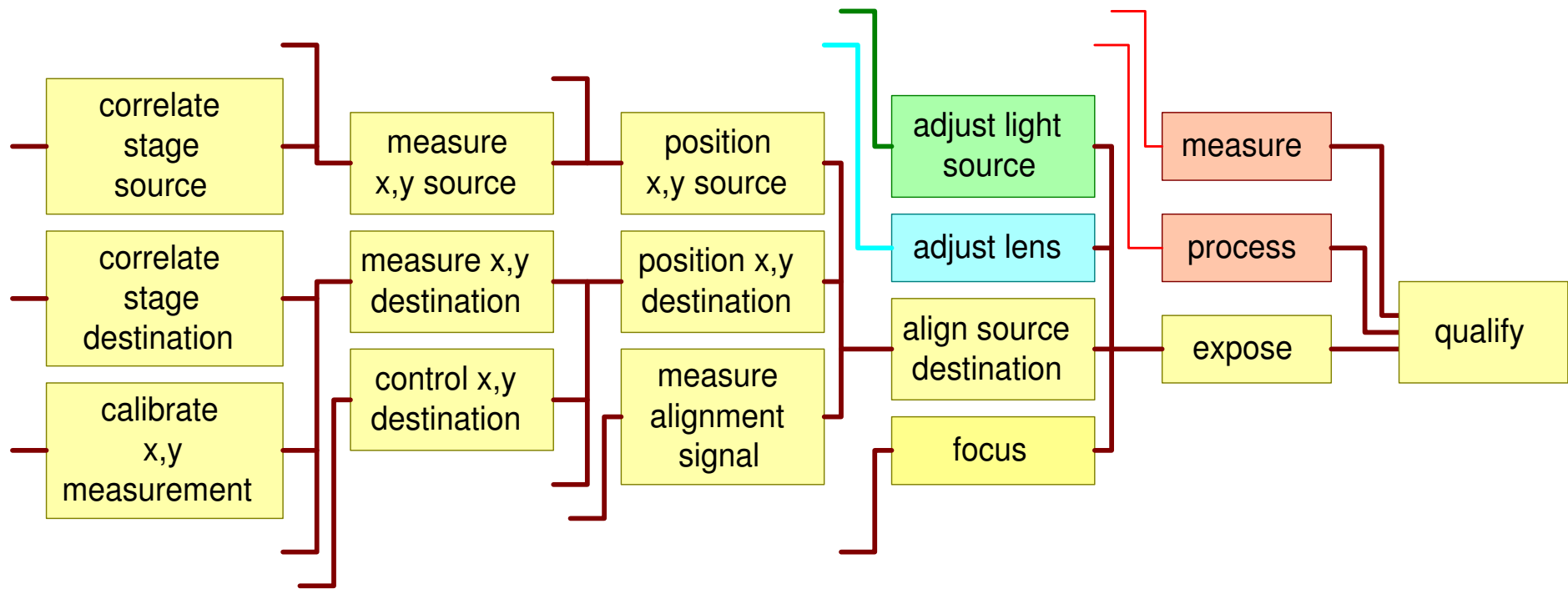
Alternatives to Integrate a Subsystem Early in the Project



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

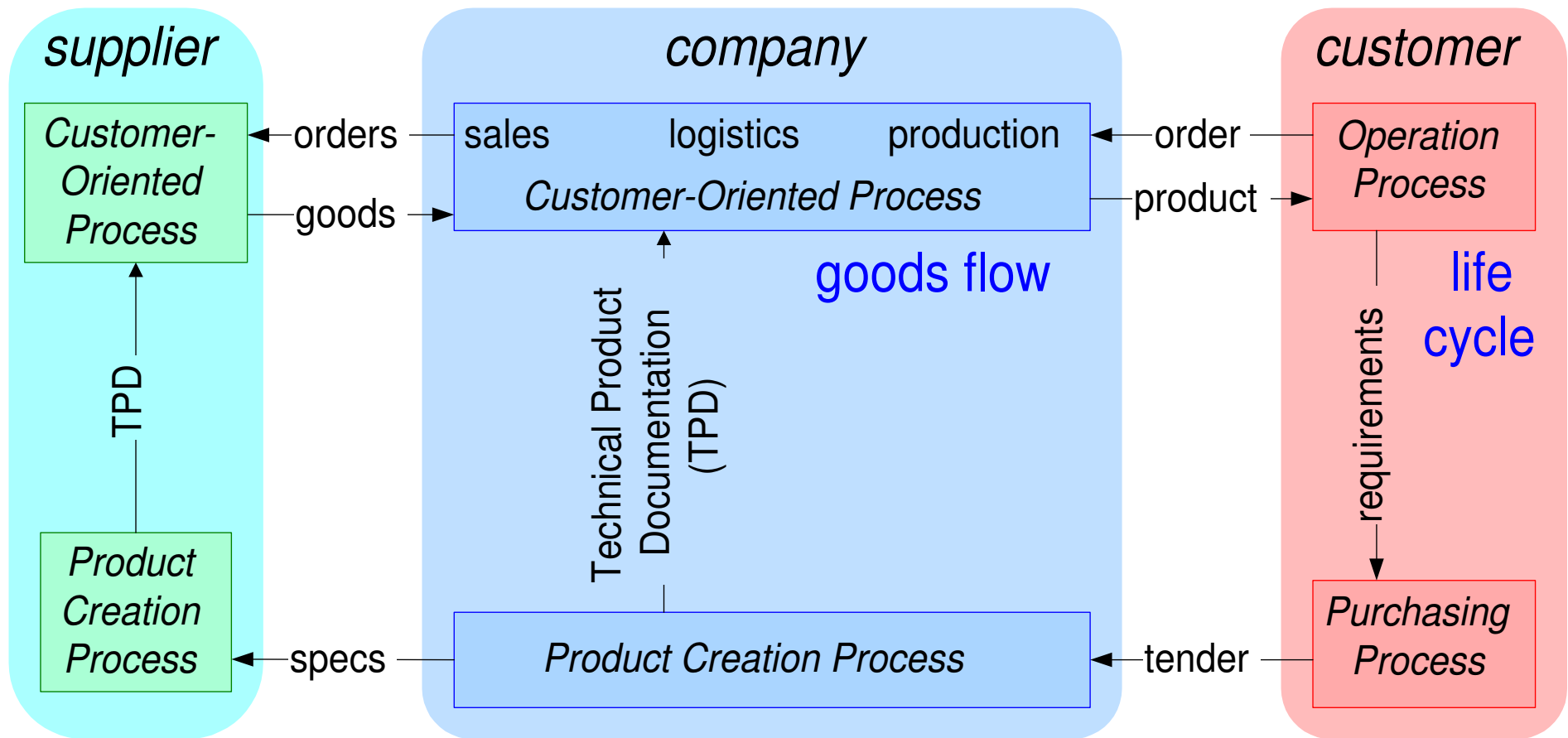
engineers

design
component test
troubleshooting
participate in test

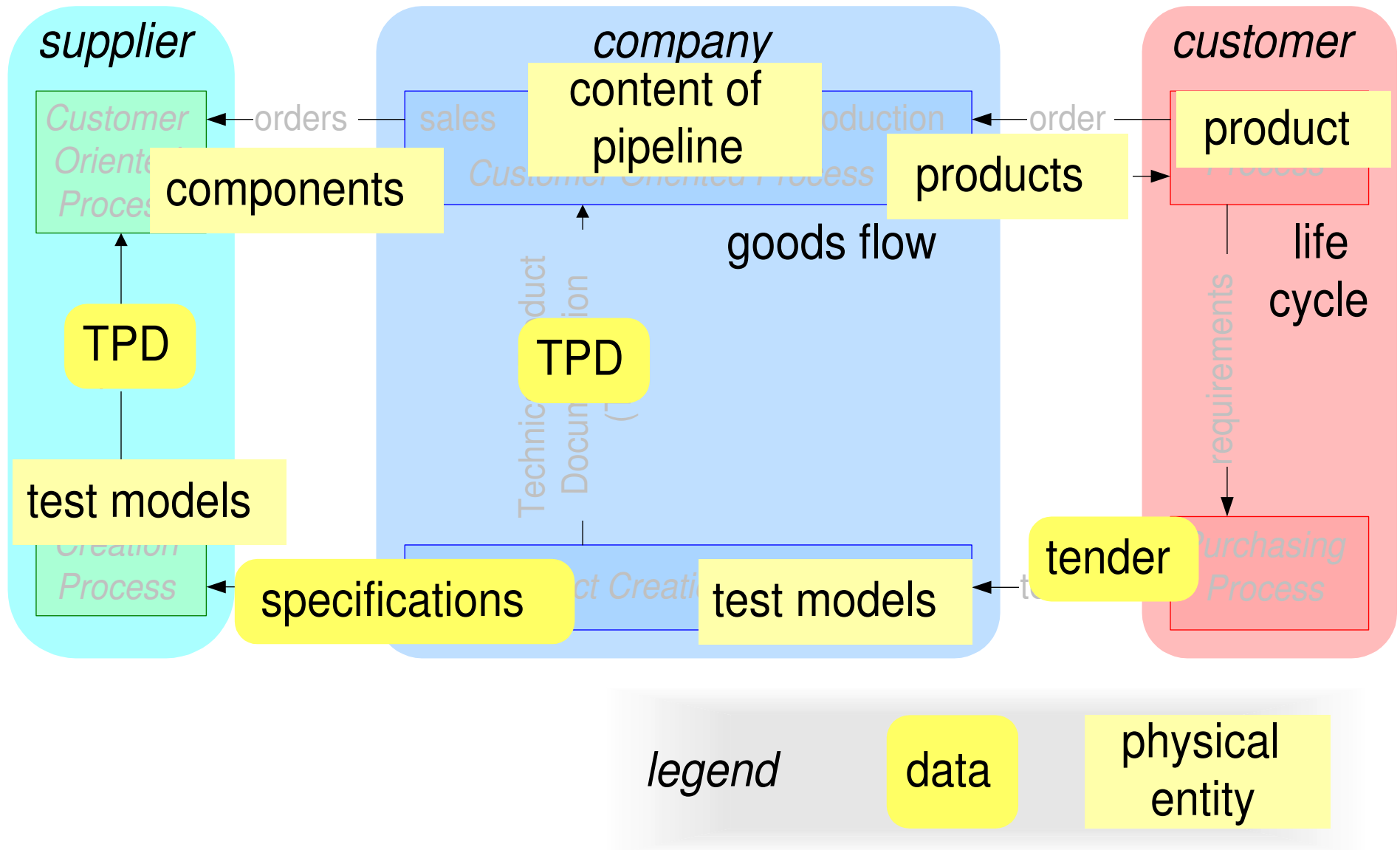
machine owner

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.