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.
<table>
<thead>
<tr>
<th>time</th>
<th>subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session 1</td>
<td>What are requirements, black box, SMART</td>
</tr>
<tr>
<td>Session 2</td>
<td>Customer and Application view, Story telling</td>
</tr>
<tr>
<td>Session 3</td>
<td>Discussion of requirement specification per team</td>
</tr>
<tr>
<td>Session 4</td>
<td>Financial viewpoint, Presentation to management</td>
</tr>
<tr>
<td>Session 5</td>
<td>Documentation How-to, Coaching and discussion session</td>
</tr>
<tr>
<td>Session 6</td>
<td>Presentation of project case to management team</td>
</tr>
</tbody>
</table>
### Schedule

**block 1; teacher provides case**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are requirements, black box, SMART</td>
<td>1/2 day</td>
</tr>
<tr>
<td>homework: make requirement specification</td>
<td>1/2 day</td>
</tr>
<tr>
<td>Customer and Application view, Story telling</td>
<td>1/2 day</td>
</tr>
<tr>
<td>homework: improve requirement specification</td>
<td>1/2 day</td>
</tr>
<tr>
<td>Discussion of requirement specification per team</td>
<td>1/2 day</td>
</tr>
</tbody>
</table>

**block 2; actual current case of OOTI education**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial viewpoint, Presentation to management</td>
<td>1/2 day</td>
</tr>
<tr>
<td>homework: make presentation outline</td>
<td>1/2 day</td>
</tr>
<tr>
<td>Documentation How-to</td>
<td>Coaching and discussion</td>
</tr>
<tr>
<td>homework: make presentation</td>
<td>1/2 day</td>
</tr>
<tr>
<td>Presentation project case to management team</td>
<td>1/2 day</td>
</tr>
<tr>
<td>individual report</td>
<td>1/2 day</td>
</tr>
</tbody>
</table>
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.
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 criterions 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  Buskerud University College

e-mail: gaudisite@gmail.com

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 31, 2014
status: concept
version: 1.3
Fundamentals of Requirements Engineering

by Gerrit Muller    Buskerud University College

e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

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

What

- customer needs:
  - What is needed by the customer?

- product specification:
  - What are we going to realize?

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

How

- What are the subsystems we will realize?
- How will the subsystems be realized?

- up to "atomic" components
System as a Black Box

- interfaces
- system seen as black box
  - inputs
  - functions
  - quantified characteristics
  - outputs
- restrictions, prerequisites
- boundaries, exceptions
- standards, regulations

Fundamentals of Requirements Engineering
12  Gerrit Muller
Stakeholders w.r.t. Requirements

- **Policy and Planning**
  - (business, marketing, operational managers)

- **Customer-Oriented Process**
  - (sales, service, production, logistics)

- **Product Creation Process**
  - (project leader, product manager, engineers, suppliers)

- **People, Process, and Technology management process**
  - (capability managers, technology suppliers)
Specific
Unambiguous
Verifiable
Quantifiable
Measurable
Complete
Traceable
The Requirements to Enable Human Use

Accessible
Understandable
Low threshold
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.
Short introduction to basic “CAFCR” model

17  Gerrit Muller

version: 0.4
July 31, 2014
CAFCRannotated

The “CAFCR” model

What does Customer need in Product and Why?

- **Customer What**
- **Customer How**
- **Product What**
- **Product How**
- **Realization**
- **Conceptual**
- **Functional**
- **Application**
- **Customer objectives**

---
drives, justifies, needs
enables, supports
Integrating CAFCR

What does Customer need in Product and Why?

Customer What
Customer How
Product What
Product How

C
A
F
C
R

Customer objectives
Application
Functional
Conceptual
Realization

context understanding
intention
objective driven
opportunities
constraint awareness
knowledge based
CAFCR can be applied recursively

Consumer

Drives

Customer's Business

Enables

Customer Drives Enables Customer's Business

Value Chain

larger scope has smaller influence on architecture

Customer Business

Drives

Enables

System (producer)

Enables

Drives
<table>
<thead>
<tr>
<th>Segmentation Axis</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographical</td>
<td>USA, UK, Germany, Japan, China</td>
</tr>
<tr>
<td>Business Model</td>
<td>Profit, Non Profit</td>
</tr>
<tr>
<td>Economics</td>
<td>High end versus cost constrained</td>
</tr>
<tr>
<td>Consumers</td>
<td>Youth, Elderly</td>
</tr>
<tr>
<td>Outlet</td>
<td>Retailer, Provider, OEM, Consumer Direct</td>
</tr>
</tbody>
</table>
Example of a small buying organization

Who is the customer?

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

decision maker(s)
purchaser
user
maintainer
operator
department head
CAFCR+ model; Life Cycle View

Customer objectives

Application

Functional

Conceptual

Realization

Life cycle

operations
maintenance
upgrades

development
manufacturing
installation

sales, service, logistics, production, R&D

Short introduction to basic “CAFCR” model

version: 0.4
July 31, 2014
BCAFCRplusLifeCycle

Gerrit Muller
Abstract

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

Key-drivers

Safety
- Reduce accident rates
- Enforce law
- Improve emergency response

Effective Flow
- Reduce delay due to accident
- Improve average speed
- Improve total network throughput
- Optimize road surface
- Speed up target groups
- Anticipate on future traffic condition

Smooth Operation
- Ensure traceability
- Ensure proper alarm handling
- Ensure system health and fault indication

Environment
- Reduce emissions

Derived application drivers

Early hazard detection with warning and signaling
Maintain safe road condition
- Classify and track dangerous goods vehicles
- Detect and warn noncompliant vehicles
- Enforce speed compliance
- Enforce red light compliance
- Enforce weight compliance

Requirements

Automatic upstream accident detection
Weather condition dependent control
Traffic speed and density measurement
Cameras
Deicing
Traffic condition dependent speed control

Note: The graph is only partially elaborated for application drivers and requirements.
Method to create Key Driver Graph

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

Key Drivers How To
25 Gerrit Muller

version: 0.2
July 31, 2014
TCAFkeyDriverSubmethod
# 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

Customer
What
Customer objectives

Key Drivers

Derived Application Drivers

Product
What
Functional

Application

requirements

Customer
How

means
may be skipped or
articulated by several
intermediate steps

Functional

functions
interfaces
performance figures

Customer
What

goal

Derived Application Drivers

Requirements

Key (Customer) Drivers

Functional
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.
Complementary Viewpoints to Capture Requirements

**Top-down**
- Key drivers (customer, business)
- Operational drivers (logistics, production, etc.)
- Roadmap (positioning and trends in time)
- Competition (positioning in the market)
- Regulations

**Bottom-up**
- "Ideal" reference design (learning vehicle)
- Prototyping, simulation (technological opportunities)
- Existing systems

Needs

Feedback

Continued Product Creation Process
Requirement Selection Process

- Customer needs
- Operational needs
- Roadmap
- Competition
- Strategy
- Selection process
- Product specification
- Need characterization
- Requirement phasing
- Technology, People, Process
- Costs and constraints
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.
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.
From story to design

What does Customer need in Product and Why?

Customer What
Customer How
Application
Functional
Conceptual
Realization

market vision

story
analyze design
case
analyze design
design

a priori solution knowledge
A day in the life of Bob

bla blah bla, rabarber music
bla bla composer bla bla
qwwwesty30 zeps.
nja nja njet njippie est quo
vaudo? Pjotr jaleski bla bla
bla bree tfgf sg fhr g
mimm bas angel heeft een
interessant excusus, 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 forever because 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 solution for an indirection problem. The invention of Bob will
obsolete all of this in one incredible move, which will make him famous forever.

He opens his PDA, logs in and enters his private secure unique 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.
Points of attention

- purpose
- scope
- viewpoint, stakeholders
- visualization
- size (max 1 A4)
- recursive decomposition, refinement
Criteria for a good story

• accessible, understandable
  "Do you see it in front of you?"

• valuable, appealing
  attractive, important
  "Are customers queuing up for this?"

• critical, challenging
  "What is difficult in the realization?"
  "What do you learn w.r.t. the design?"

• frequent, no exceptional niche
  "Does it add significantly to the bottom line?"

• specific
  names, ages, amounts, durations, titles, ...
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.
Value and Challenges in this story

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

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
Abstract
This module addresses the presentation of architectural issues to higher management teams.
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.
Product Margin = Sales Price - Cost

Margin per product. The margin over the sales volume, must cover the fixed costs, and generate profit:
- transportation, insurance, royalties per product, ...

Cost per product, excluding fixed costs:
- purchase price of components may cover development cost of supplier

- material
- labour
- miscellaneous
- margin
- retailer margin and costs
Profit as function of sales volume

- Sales volume (in units)
- Income
- Expenses
- Break-even point
- Profit
- Expected sales volume
- Fixed costs
- Variable costs
Investments, more than R&D

<table>
<thead>
<tr>
<th>financing</th>
</tr>
</thead>
<tbody>
<tr>
<td>marketing, sales</td>
</tr>
<tr>
<td>training sales &amp; service</td>
</tr>
<tr>
<td>NRE: outsourcing, royalties</td>
</tr>
<tr>
<td>research and development</td>
</tr>
</tbody>
</table>

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

strategic choice: NRE or per product

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

often a standard staffing rate is used that covers most costs above: R&D investment = Effort * rate
Income, more than product sales only

\[
\sum_{\text{services}} \text{income}_{\text{service}} \\
\sum_{\text{options}} \text{sales price}_{\text{option}} \times \text{volume}_{\text{option}} \\
\text{sales price}_{\text{product}} \times \text{volume}_{\text{product}}
\]

- Other recurring income
- Services
- Options, accessories
- Products

License fees, pay per movie, content, portal updates, maintenance
### The Time Dimension

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>investments</strong></td>
<td>100k$</td>
<td>400k$</td>
<td>500k$</td>
<td>100k$</td>
<td>100k$</td>
<td>60k$</td>
<td>20k$</td>
</tr>
<tr>
<td><strong>sales volume (units)</strong></td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td><strong>material &amp; labour costs</strong></td>
<td>-</td>
<td>-</td>
<td>40k$</td>
<td>200k$</td>
<td>400k$</td>
<td>600k$</td>
<td>600k$</td>
</tr>
<tr>
<td><strong>income</strong></td>
<td>-</td>
<td>-</td>
<td>100k$</td>
<td>500k$</td>
<td>1000k$</td>
<td>1500k$</td>
<td>1500k$</td>
</tr>
<tr>
<td><strong>quarter profit (loss)</strong></td>
<td>(100k$)</td>
<td>(400k$)</td>
<td>(440k$)</td>
<td>200k$</td>
<td>500k$</td>
<td>840k$</td>
<td>880k$</td>
</tr>
<tr>
<td><strong>cumulative profit</strong></td>
<td>(100k$)</td>
<td>(500k$)</td>
<td>(940k$)</td>
<td>(740k$)</td>
<td>(240k$)</td>
<td>600k$</td>
<td>1480k$</td>
</tr>
</tbody>
</table>

*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

Simplistic Financial Computations for System Architects.

Gerrit Muller
What if ...?

early more expensive product + follow-on

delay of 3 months

original model
Stacking Multiple Developments

-2
-1
0
1
2
3
4
5
6
7
8
9
1
2
3
4
5
6
7
8
9
10
11
12
13
14
cumulative 1
cumulative 2
cumulative 3
cumulative 4
cumulative total
M€

Simplistic Financial Computations for System Architects.

version: 1.3
July 31, 2014
SFCmultipleDevelopments
Fashionable financial yardsticks

Return On Investments (ROI)

Net Present Value

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

turnover / fte outsourcing reduces headcount, improves this ratio

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

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

cash-flow fast growing companies combine profits with negative cash-flow, risk of bankruptcy
How to present architecture issues to higher management

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

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

This article describes how to give a more effective presentation to higher management teams. Subjects discussed are the preparation, content and form, do and don’t advise.
Architectural issues related to managerial viewpoints

How to present architecture issues to higher management

version: 0.1
July 31, 2014
AMIintroduction
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!

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

- understand audience
  - gather audience background
  - analyze audience interests
  - identify expected responses
  - simulate audience, exercise presentation

70% of effort
30% of effort

How to present architecture issues to higher management

Gerrit Muller
Recommended content

+ clear problem statement (what, why)
+ solution exploration (how)
+ options, recommendations
+ expected actions or decisions

supported by
facts and figures
How to present architecture issues to higher management

57  Gerrit Muller

version: 0.1  
July 31, 2014  
AMInfoTypes

**Market drivers**
- cost
ttm
wow
DRM
- MPEG4
MP3
color display
ePen
GPS sensor
UTMS
BT
802.11b
- integration
multiple suppliers
nifty features
Hollywood pact
standards

**Options**
- A
- B

**Typical performance**
- load
- transfer/sec

**Bill of material**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>infra</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>sensor</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>display</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>total</td>
<td>36</td>
<td>37</td>
</tr>
</tbody>
</table>

**Schedule**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>302304</td>
<td>310</td>
<td>318</td>
</tr>
<tr>
<td>326</td>
<td></td>
<td></td>
</tr>
<tr>
<td>308</td>
<td>322</td>
<td>326330</td>
</tr>
</tbody>
</table>

**ffe's**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>infra</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>control</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>display</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>total</td>
<td>17</td>
<td>23</td>
</tr>
</tbody>
</table>

**Profit-investment**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>salesprice</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>cost/p</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>units</td>
<td>1M</td>
<td>1M</td>
</tr>
<tr>
<td>sales</td>
<td>10M</td>
<td>10M</td>
</tr>
<tr>
<td>costs</td>
<td>3M</td>
<td>4M</td>
</tr>
<tr>
<td>investment</td>
<td>2M</td>
<td>3M</td>
</tr>
<tr>
<td>profit</td>
<td>5M</td>
<td>3M</td>
</tr>
</tbody>
</table>

**Power details**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>infra</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>sensor</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>display</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>power</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>total</td>
<td>36</td>
<td>37</td>
</tr>
</tbody>
</table>

**Operating principle**

**worst-case performance**

**recommendation**
- recommendation: select A
- follow up: allocate Jan, Piet, Klaas per 1/11
go/nogo 1/1/03

**backup material**

**version**
version: 0.1
July 31, 2014

**Embedded Systems INSTITUTE**
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)
+ 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

13:30  14:00  15:00  16:00  17:00

prepare in team of 4

1 1 2 2
3 3 4 4

present and discuss

feedback
Module Supporting Processes

by Gerrit Muller 
Buskerud University College

e-mail: gaudisite@gmail.com
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.
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.
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
Ease of reading, “juiciness”

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

Understandability

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

Reachability in a limited number of steps
Responsibility Requirements

- single author
- limited amount of reviewers
Scalability Requirements

- well defined documentation structure
- overview specifications at higher aggregation levels
- recursive application of structure and overview
- delegation of review process
The Stakeholders of a Single Document

- **Project leader**
  - is responsible for time, budget, result

- **architect or editor**
  - is responsible for technical
  - interacts with others

- **author**
  - writes
  - interacts with specification

- **consumer**
  - uses
  - interacts with implementation

- **implementation**
  - realizes
  - interacts with producer

- **specification**
  - describes
  - interacts with context

- **context**
  - others

Legend:
- **relation**
- **artifact**
- **stakeholder**

Granularity of Documentation

version: 1.2
July 31, 2014
DGdocumentationRoles

Gerrit Muller
Documentation Tree by Recursive Decomposition

Granularity of Documentation
Gerrit Muller

version: 1.2
July 31, 2014
DGdocumentRecursion
Payload: the Ratio between Content and Overhead

front page

- title
- identification
- author
- distribution
- status
- review

history
changes

1. aap
2. noot
3. mies

lists
and ca 50%
text

meta information
max 2 pages

contents
2..18 pages

max 2 pages
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.
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 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

metal printing cell: systems and performance model

Customer key-drivers and Key Performance Parameters

Lean and A3 Approach to Supporting Processes
version: 0.1
July 31, 2014
Gerrit Muller
multiple related views

quantifications

one topic
per A3

close to stakeholder experience

capture
"hot" topics

digestable
(size limitation)

practical

source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/
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*.
**Product Life Cycle and Change Management**

*Micro Specification Control Board*

- Project team present
- Specification = communications means
- Very dynamic, many changes
- Light weight review process

*Maintenance Control Board*

- No project team any more
- Documentation = organizational memory
- Changes only to cope with logistics or safety problems
Light Weight Specification Review Process

- final review = final check contents
- authorization = check process
- consultation & review

draft

concept

authorized

change request

specification specific Change Control Board
4 peoples/roles:
1 producer
1 consumer
1 context
1 independent

criteria for reviewers:
+ know how
+ critical
+ sufficient time

by "lowest" operational manager: project leader, subsystem PL, ...

- wide group of people, with an active concern or an expected contribution;
- many iterations
- multiple media:
  + meetings,
  + on paper
  + informal et cetera

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

\[ \sum \text{all Forces} \quad \text{induces} \quad \text{New Process} \quad \text{reaction} \quad \text{Net change} = \text{Support} \]
Template as Support for Process

principle — *drives* — process — *elaborated in* — procedure — *supported by* — tool

*formalism*

abstract — *specific and executable* — template
Types of Templates

- **Recommended template type**
  - **Layout only**
  - **Meta information**
  - **Prescribing contents**

**Types:**
- **Header**
- **Body**
- **Footer**

**Templates:**
- **Title, Date**
- **Page, Author**

**Content Blocks:**
- **Title**
- **Abstract**
- **Introduction** (1)
- **Scope** (2)
- **Design** (3)
- **Interfaces** (17)

---

*Template How To*  
Gerrit Muller

*version: 1.6*  
*July 31, 2014*  
*THTypes*
## Recommendation

<table>
<thead>
<tr>
<th>template type</th>
<th>context knowhow</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>layout only</td>
<td>no</td>
<td>low</td>
</tr>
<tr>
<td>meta information</td>
<td>process</td>
<td>high</td>
</tr>
<tr>
<td>prescribing content</td>
<td>process and domain</td>
<td>constraining</td>
</tr>
</tbody>
</table>

- 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

Extract template

Implement document

Evaluate

Use

Template How To
90      Gerrit Muller

version: 1.6
July 31, 2014
THTdevelopmentSpiral
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
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
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.
Typical Concurrent Product Creation Process

- Strategy
  - 0. Feasibility
  - 1. Definition
  - 2. System Design
  - 3. Engineering
  - 4. Integration & Test
  - 5. Field Monitoring
  - 6. Product Operational Life Cycle
Zooming in on Integration and Tests

0. feasibility
1. definition
2. system design
3. engineering
4. integration & test
5. field monitoring
6. product operational life cycle

integrate

system test
alpha test
beta test
gamma test
Integration Takes Place in a Bottom-up Fashion

- Component
- Subsystem
- System function
- Product
- Context
- Integrate
- Alpha test
Transition from Previous System to New System

2 partial systems for SW testing

2 existing base systems

new base systems

adopt existing base SW

new application

SW dev system

SW for new HW subsystem

new HW subsystem

existing base system

test and refine application

test SW for new HW subsystem

test HW subsystem

integrate subsystem

integrate and refine application

integrate HW system

integrate system

application integration

new subsystem integration

integrated system

time
Alternatives to Integrate a Subsystem Early in the Project

- Physical reality
- Physical simulated
- Complex virtual
- Simple physical

- Virtual environment
  - Simulated subsystems
  - Stubs

- To-be-integrated subsystem

- (Modified) existing subsystems
- (Prototype) new subsystems

System Integration How-To
101 Gerrit Muller

version: 0.2
July 31, 2014
SINTenvironments
## Stepwise Integration Approach

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Determine most critical system performance parameters.</td>
</tr>
<tr>
<td>2</td>
<td>Identify subsystems and functions involved in these parameters.</td>
</tr>
<tr>
<td>3</td>
<td>Work towards integration configurations along these chains of subsystems and functions.</td>
</tr>
<tr>
<td>4</td>
<td>Show system performance parameter as early as possible; start with showing &quot;typical&quot; system performance.</td>
</tr>
<tr>
<td>5</td>
<td>Show &quot;worst-case&quot; and &quot;boundary&quot; system performance.</td>
</tr>
<tr>
<td>6</td>
<td>Rework manual integration tests in steps into automated regression tests.</td>
</tr>
<tr>
<td>7</td>
<td>Monitor regression results with human-driven analysis.</td>
</tr>
<tr>
<td>8</td>
<td>Integrate the chains: show system performance of different parameters simultaneously on the same system.</td>
</tr>
</tbody>
</table>
Order of Functions Required for the IQ of a Waferstepper

- Order of Functions:
  1. correlate stage source
  2. correlate stage destination
  3. calibrate x,y measurement
  4. measure x,y source
  5. measure x,y destination
  6. control x,y destination
  7. position x,y source
  8. position x,y destination
  9. measure alignment signal
  10. adjust light source
  11. adjust lens
  12. align source destination
  13. measure
  14. process
  15. expose
  16. qualify
  17. measure x,y
  18. destination
  19. correlate stage
  20. source
  21. correlate stage
  22. destination
  23. calibrate x,y
  24. measurement
  25. measure
  26. position x,y
  27. source
  28. position x,y
  29. destination
  30. measure alignment signal
  31. adjust light source
  32. adjust lens
  33. align source destination
  34. measure
  35. process
  36. expose
  37. qualify

System Integration How-To
103   Gerrit Muller

version: 0.2
July 31, 2014
SINTorder
### Roles and Responsibilities During the Integration Process

<table>
<thead>
<tr>
<th><strong>Role</strong></th>
<th><strong>Responsibilities</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Leader</strong></td>
<td>organization, resources, schedule, budget</td>
</tr>
<tr>
<td><strong>Systems Architect/Engineer/Integrator</strong></td>
<td>system requirements, design inputs, test specification, schedule rationale, troubleshooting, participate in test</td>
</tr>
<tr>
<td><strong>System Tester</strong></td>
<td>test, troubleshooting, report</td>
</tr>
<tr>
<td><strong>Logistics and Administrative Support</strong></td>
<td>configuration, orders, administration</td>
</tr>
<tr>
<td><strong>Engineers</strong></td>
<td>design, component test, troubleshooting, participate in test</td>
</tr>
<tr>
<td><strong>Machine Owner</strong></td>
<td>maintain test model, support test</td>
</tr>
</tbody>
</table>
Simplified Process Diagram

**Customer-Oriented Process**
- Orders from supplier to company
- Goods from supplier to company
- Specs from company to product creation process

**Product Creation Process**
- Technical Product Documentation (TPD)
- Goods flow from company to customer

**Operation Process**
- Order from customer to purchasing process
- Product from purchasing process to customer

**Purchasing Process**
- Tender from customer to purchasing process

**System Integration How-To**
105  Gerrit Muller
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.
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.