Abstract

The Course System Architecture SARCH is a course for system architects, potential system architects and immediate stakeholders of the system architect, such as project leaders, designers and marketing managers.

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

Course System Architecting Introduction

Distribution
This article or presentation is written as part of the Gaudi project. The Gaudi 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

This article describes the course Systems Architecting. The course is set up to make the art of system architecting more accessible. The course will address a wide spectrum of issues in relation with system architecture, such as: Processes, Business, Role and task of the system architect (team), Roadmapping, System Architect toolkit, Technical, Skills, and Psycho Social
<table>
<thead>
<tr>
<th>Session 1</th>
<th>Positioning the System Architecture Process, Product Creation Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session 2</td>
<td>Role and Task of the System Architect</td>
</tr>
<tr>
<td>Session 3</td>
<td>Requirements Capturing</td>
</tr>
<tr>
<td>Session 4</td>
<td>System Architect Toolkit</td>
</tr>
<tr>
<td>Session 5</td>
<td>Roadmapping</td>
</tr>
<tr>
<td>Session 6</td>
<td>Product Families, generic developments</td>
</tr>
<tr>
<td>Session 7</td>
<td>Documentation, reviewing and other supportive processes; The role of Software in complex products</td>
</tr>
<tr>
<td>Session 8</td>
<td>BoM presentation</td>
</tr>
<tr>
<td>Session 9</td>
<td>Psycho Social side</td>
</tr>
<tr>
<td>Session 10</td>
<td>Wrap up, Expectations, How to continue, Evaluation</td>
</tr>
</tbody>
</table>
Structure

passive
Theory
dull

Insight
Practical Illustration
vivid

Interaction
vivid

Spin-off:
cross-fertilization

active
Abstraction
Exercise

Insight

Abstraction: 
Spin-off:
cross-fertilization

Theory:
dull

Interaction: 

Exercise

vivid
Timing Template of one subject

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00</td>
<td>interactive exploration</td>
</tr>
<tr>
<td>10:00</td>
<td>broadcast</td>
</tr>
<tr>
<td>11:00</td>
<td>broadcast</td>
</tr>
<tr>
<td>12:00</td>
<td>interactive discussion</td>
</tr>
<tr>
<td>12:30</td>
<td>break</td>
</tr>
<tr>
<td>13:30</td>
<td>groupwork</td>
</tr>
<tr>
<td>14:00</td>
<td>discussion</td>
</tr>
<tr>
<td>15:00</td>
<td>discussion</td>
</tr>
<tr>
<td>16:00</td>
<td>break</td>
</tr>
<tr>
<td>17:00</td>
<td>break</td>
</tr>
</tbody>
</table>
Rules of the Interactive Parts

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

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

by Gerrit Muller    HBV-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

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

• Consolidate existing Systems Architecting Methods
  \textit{evaluate, reflect, generalize}

• Make the Systems Architecting art more accessible
  \textit{case descriptions}

• Enable the education of (future) System Architects
  \textit{curriculum, course material}

• Research new or improved Systems Architecting Methods
  \textit{industry as laboratory}
Modular approach

- course
  - all slides
- module
- presentation
- reader
- book
- paper or chapter
- section
- figure

The Gaudí Project
11 Gerrit Muller
version: 3.1
March 9, 2015
GPdocumentModularity
Show Early to Get Feedback

leading principle:

show documents under construction but clearly show their status
Growth of the System Architect

- Root technical knowledge
- Generalist technical knowledge
- Business, application insight
- Process insight
- Psychosocial skills
Positioning Courses

root technical designer

Gaudi courses

Performance Architecting System Performance

Modeling System Modeling and Analysis

CAFCR Multi-Objective System Architecting and Design

SARCH System Architecting

Platforms and Evolvability Architecting Evolvable Product Families

business, application insight

process insight

psycho-social interests
Positioning Books

root technical designer

technical generalist

business, application insight

process insight

psycho-social interests

Architecting System Performance

System Modeling and Analysis: a Practical Approach

CAFCR PhD thesis

CAFCR Architectural Reasoning

Systems Architecting

Supporting Processes

Composable Architectures

Human Measure

The Gaudí Project
15 Gerrit Muller

version: 3.1
March 9, 2015
GPPositionBooks
Productivity: number of new entries

<table>
<thead>
<tr>
<th>Year</th>
<th>SARCH</th>
<th>Performance</th>
<th>CAFCR</th>
<th>Modeling</th>
<th>Evolvability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2003</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average productivity: 23.8

Philips and BUC

The Gaudí Project

version: 3.1
March 9, 2015
Gerrit Muller
Abstract

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

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

- **Policy and Planning Process**
  - Customer Roadmap
  - Business Drivers

- **Product Creation Process**
  - Technology, Process, and People roadmaps
  - Budgets
  - Product roadmap
  - Budget, plan

- **Customer-Oriented Process**
  - Information
  - Order
  - Product
  - Goods and feedback
  - Material

- **People, Process, and Technology Management Process**
  - Needs and Feedback
  - People Technology Process
  - Technical Product Documentation
  - Product-related processes

**Business Drivers**

- **Customer Roadmap**
- **Budget**
- **Product roadmap**
- **Needs and feedback**
- **Technical Product Documentation**
- **Product-related processes**
- **Support**

**Presales**, **Sales**, **Logistics**, **Production**, **Service**
Financial Characterization of Decomposition

Customer Oriented Process

Product Creation Process

Policy and Planning Process

People and Technology Management Process

Customer

Sales
Logistics
Production
Service
Presales

Management

Cashflow Generation

Tomorrow's Cashflow

Assets

People and Technology Management Process

Customer Roadmap
Business Drivers
Product Roadmap
Budget, plan
Product Needs and feedback
Material
Presales
Sales
Logistics
Production
Service
Support

Information
Order
Product

Product Needs and Feedback
Technical Product Documentation
Product-related processes

Budget

Technology, Process, and People roadmaps
Bodies

Version: 1.1
March 9, 2015
Multiple Instances per Process

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

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

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

**Policy and Planning Process:** One per business. This is the pro-active integrating process.
The Value Chain and the Opposite Feedback Flow

- Policy and Planning Process
- Customer-Oriented Process
- Product Creation Process
- People, Process, and Technology Management Process

[Diagram showing the flow of information and feedback from customer to various processes and back]
Decomposition of the Customer Oriented Process

Process Decomposition of a Business
23    Gerrit Muller
version: 1.1
March 9, 2015
PDBcustomerOriented
Abstract

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

Customer

Policy and Planning Process

Customer Oriented Process

Product Creation Process

People and Technology Management Process

Customer Roadmap

Business Drivers

Budget, plan

Product Requirements and feedback

Technical Product Documentation

Product related processes

People Technology Process

Support

Product

Order

Feedback

Information

presales sales logistics production service

Product Requirements and feedback

People Technology Management Process

Technology, Process and People roadmaps

Budgets

Product roadmap

Business Drivers

Customer Roadmap

Product Creation Process

version: 2.2
March 9, 2015
PCPcontext
Phasing of the PCP at Business Level

0. feasibility
1. definition
2. system design
3. engineering
4. integration & test
5. field monitoring

- sales
- logistics
- production
- service
- development & engineering: marketing, project management, design
Phasing the Design Control Process

Legend:
- **core information in draft**
- **50%**
- **most information available in concept**
- **information is stable enough to use heavier change control**
- **full under development**
- **preparing or updating work**

The Product Creation Process

Gerrit Muller

version: 2.2
March 9, 2015
PCPdesignPhases
Advantages and Disadvantages of a Phased Process

**benefits**
- blueprint: how to work
- reuse of experience
- employees know *what* and *when*
- reference for management

**disadvantages**
- following blueprint blindly
- too bureaucratic
- transitions treated black and white
Characteristics of a Phase Model

- Large impact decisions
- Phase transitions
- Check points
- Needs
- Specification
- Design
- Verification
- Engineering

Concurrent process:

0. Feasibility
1. Definition
2. System design
3. Engineering
4. Integration & test
5. Field monitoring

Iteration:

Order:
- Long-lead items
- High-cost items
- Product announcement

Decisions:
- Large impact decisions
Define a minimal set of *large-impact* decisions.

Define the mandatory and supporting information required for the decision.

Schedule a decision after the appropriate phase transition.

Decide explicitly.

Communicate the decision clearly and widely.
Evolutionary PCP model

- test and evaluate
- requirements specification
- design
- build

2% of budget (EVO)
2 weeks (XP)
up to 2 months per cyclus
Decomposition of the Product Creation Process

Product Creation Process

Operational Management
- specification
- budget
- time
  - planning
  - progress control
  - resource management
  - risk management
  - project log

Design Control
- technical
  - needs
  - what is needed
  - specification
  - what will be realized
  - design
  - how to realize
  - verification
  - meeting specs
  - following design
  - engineering
  - how to produce
  - and to maintain

Marketing
- profitability
- saleability
  - customer input
  - customer expectations
  - commercial structure
  - product pricing
  - market introduction
  - introduction at customer
  - feedback
Operational Organization of the PCP

- **Entire Portfolio**
  - **Operational Manager**
  - **Technical Portfolio Architect**
  - **Commercial Portfolio Marketing Manager**

- **Product Family**
  - **Family Operational Manager**
  - **Family Marketing Manager**

- **Single Product**
  - **Product Architect**
  - **Product Manager**

- **Subsystem**
  - **Subsystem Architect**

- **Module**
  - **Developers**
Prime Responsibilities of the Operational Leader

- **Specification**
- **Quality**
- **Resources**
- **Time**
The Rules of the Operational Game

- define project
- update project
- assess risks
- determine feasibility
- execution within normal quality rules

business management

project leader

- specification, resources, time
- accept or reject
Operational Teams

Sales Manager

Application Manager

Requirements Analyst

Test Engineer

Service

Development support

Technology-Specific Architects

Subsystem Architects

Subsystem Operational Leaders

Operational Support (project manager)

Operational Leader (project leader)

Quality Assurance

The Product Creation Process

version: 2.2
March 9, 2015
PCPconcentricTeams
The System Architecture Process

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

Abstract

The System Architecture Process is positioned in the business context. This process bridges the gap between the Policy and Planning Process and the Product Creation Process.

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

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

Customer-Oriented Process

Systems Architecting Process

People, Process, and Technology Management Process
System Architecting Relation between PPP and PCP

Policy and Planning Process

Context: Product Portfolio, Time

Vision, Policy, Intention

Practical Knowledge

Feedback from Reality

Product Creation Process
System Architecting Key Issues

**Key words**
- Balance
- Consistency
- Integrity
- Simplicity
- Elegance
- Stakeholder satisfaction

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

**Example trade-offs**
- Performance ↔ synergy
- Functionality ↔ specific solution
- Qualities ↔
1. Map operational organization.
2. Report on one flip the best case.
3. Identify the relationships of the core team: geographical, organizational, psychological, et cetera.
4. Report the result of 3 on one flip.
Process Decomposition of a Business

Importance in Financial terms

Value Chain and Feedback Flow

intentionally left blank
Product Creation Process

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

<table>
<thead>
<tr>
<th>0. feasibility</th>
<th>1. definition</th>
<th>2. system design</th>
<th>3. engineering</th>
<th>4. integration &amp; test</th>
<th>5. field monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>sales</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>logistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>production</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>service</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>development &amp; engineering: marketing, project management, design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Phased Process**

<table>
<thead>
<tr>
<th>0. feasibility</th>
<th>1. definition</th>
<th>2. system design</th>
<th>3. engineering</th>
<th>4. integration &amp; test</th>
<th>5. field monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>needs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>specification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>verification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>engineering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:

- [ ] core information in draft
- [ ] 50%
- [ ] most information available in concept
- [ ] information is stable enough to use heavier change control

**Incremental Development**

- test and evaluate
- requirements specification
- build
- design

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

intentionally left blank
PCP Decomposition and Operational Management

PCP decomposition

<table>
<thead>
<tr>
<th>Operational Management</th>
<th>Design Control</th>
<th>Marketing</th>
</tr>
</thead>
<tbody>
<tr>
<td>specification</td>
<td>technical</td>
<td>profitability sellability</td>
</tr>
</tbody>
</table>

Architecture at all levels; From portfolio to subsystem

<table>
<thead>
<tr>
<th>entire portfolio</th>
<th>product family</th>
<th>single product</th>
<th>subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>portfolio operational manager</td>
<td>family operational manager</td>
<td>(single product) project leader</td>
<td>subsystem project leader</td>
</tr>
<tr>
<td>portfolio architect</td>
<td>family architect</td>
<td>product architect</td>
<td>subsystem architect</td>
</tr>
<tr>
<td>portfolio marketing manager</td>
<td>family marketing manager</td>
<td>product manager</td>
<td>developers</td>
</tr>
</tbody>
</table>

Operational Commitment

<table>
<thead>
<tr>
<th>Specification</th>
<th>Quality</th>
<th>Resources</th>
<th>Time</th>
</tr>
</thead>
</table>

Core: Operational + Technical + Commercial

Exercise Product Creation Process

version: 2.3
March 9, 2015
System Architecture Process

In Business Context

Key Issues

- **key words**
  - balancing acts
  - balance
  - consistency
  - integrity
  - simplicity
  - elegance
  - stakeholder satisfaction

- **example trade-offs**
  - performance
  - functionality
  - synergy
  - qualities
  - specific solution

5 Views

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

### 5 Views Diagram

<table>
<thead>
<tr>
<th>Customer</th>
<th>Application</th>
<th>Functional</th>
<th>Conceptual</th>
<th>Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

intentionally left blank
Abstract

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

Customer and Life-Cycle Needs (*what is needed*)

System Specification (*what will be realized*)

Design Specification (*how the system will be realized*)

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

Verification Report (*the result of the verification*)

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

Roadmap
Responsibilities of the System Architect

Balance Consistency Decomposition Integration Overview

KISS Elegance Simple Integrity Fitting

satisfied stakeholders system context

The Role and Task of the System Architect

version: 2.0
March 9, 2015
RSAre responsibilities

Embedded Systems Innovation

The Role and Task of the System Architect

52 Gerrit Muller
## Examples of Secondary Responsibilities

<table>
<thead>
<tr>
<th>Responsibility</th>
<th>Primary Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business plan, profit</td>
<td>Business manager</td>
</tr>
<tr>
<td>Schedule, resources</td>
<td>Project leader</td>
</tr>
<tr>
<td>Market, salability</td>
<td>Marketing manager</td>
</tr>
<tr>
<td>Technology</td>
<td>Technology manager</td>
</tr>
<tr>
<td>Process, people</td>
<td>Line manager</td>
</tr>
<tr>
<td>Detailed designs</td>
<td>Engineers</td>
</tr>
</tbody>
</table>
What does the System Architect do?

- think, analyze
- listen, talk, walk around
- design, brainstorm, explain
- assist project leader with work breakdown, schedule, risks
- present, meet, teach, discuss
- test, integrate
- write, consolidate, browse
- read, review
- travel to customer, supplier, conference
- provide vision and leadership
## From Detail to Overview

### Quantity per year (order-of-magnitude)

<table>
<thead>
<tr>
<th>Task</th>
<th>Quantity</th>
<th>Architect time per item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driving views</td>
<td>$10^2$</td>
<td>100 h</td>
</tr>
<tr>
<td>Shared issues</td>
<td>$10^2$</td>
<td>1 h</td>
</tr>
<tr>
<td>Touched details</td>
<td>$10^4$</td>
<td>0.5 – 10 min</td>
</tr>
<tr>
<td>Seen details</td>
<td>$10^5$ – $10^6$</td>
<td>0.1 – 1 sec</td>
</tr>
<tr>
<td>Product details</td>
<td>$10^7$ – $10^{10}$</td>
<td></td>
</tr>
<tr>
<td>Real-world facts</td>
<td>infinite</td>
<td></td>
</tr>
</tbody>
</table>

### Consolidation in deliverables
- Meetings
- Informal contacts
- Sampling scanning
Abstractions only exist for concrete facts.
Visible Output versus Invisible Work

From Manager perspective

Decreasing Visibility

Deliverables

Responsibilities

Activities

The Role and Task of the System Architect

version: 2.0
March 9, 2015
RSApyramid

The Role and Task of the System Architect

version: 2.0
March 9, 2015
RSApyramid
The Awakening of a System Architect

by Gerrit Muller    Buskerud University College

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

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

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.

March 9, 2015
status: concept
version: 1.1
Typical Growth of a System Architect

- root technical knowledge
- generalist technical knowledge
- business, application insight
- process insight
- psychosocial skills
Generalist versus Specialist

The Awakening of a System Architect

version: 1.1
March 9, 2015
MATgeneralistVsSpecialist
Generalists and Specialists are Complementary

The Awakening of a System Architect

61  Gerrit Muller
Spectrum from Specialist to System Architect

The Awakening of a System Architect

version: 1.1
March 9, 2015
MATfromSpecialistToSystemArchitect
Abstract

A system architect needs skills to apply different interaction styles, depending on the circumstances. This document discusses the following interaction styles: provocation, facilitation, leading, empathic, interviewing, whiteboard simulation, and judo tactics.
provocation
when in an impasse: provoke
effective when used sparsely

facilitation
especially recommended when new in a field:
contribute to the team, while absorbing new knowledge

leading
provide vision and direction, make choices
risk: followers stop to give the needed feedback

empathic
take the viewpoint of the stakeholder
acknowledge the stakeholder's feelings, needs, concerns

interviewing
investigate by asking questions

whiteboard simulation
invite a few engineers and walk through
the system operation step by step

judo tactics
first listen to the stakeholder and then
explain cost and alternative opportunities
Exercise Role and Task of the System Architect

Role play with 3 roles and optional observer:
• 1 operational leader (project leader)
• 1 system architect
• 1 marketing manager
• 1 observer (optional)

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

Present (max. 2 flips) the result and the process (the relation and interaction of the three roles).
Role and Task of a System Architect

**Deliverables**
- Report
- Spec
- Design

**Responsibilities**
- Balance
- Consistency
- Decomposition
- Integration

**Daily Activities**
- Think, analyze, listen, talk, walk around
- Design, brainstorm, explain
- Assist project leader with work breakdown, schedule, risks
- Present, meet, teach, discuss
- Test, integrate, write, consolidate, browse
- Read, review
- Travel to customer, supplier, conference
- Provide vision and leadership

**From detail to overview**

<table>
<thead>
<tr>
<th>Quantity per year (order-of-magnitude)</th>
<th>architect time per item</th>
</tr>
</thead>
<tbody>
<tr>
<td>consolidation in deliverables meetings</td>
<td>10</td>
</tr>
<tr>
<td>touched details</td>
<td>$10^4$</td>
</tr>
<tr>
<td>informal contacts</td>
<td></td>
</tr>
<tr>
<td>seen details</td>
<td>$10^5$ – $10^6$</td>
</tr>
<tr>
<td>product details</td>
<td>$10^7$ – $10^{10}$</td>
</tr>
<tr>
<td>real-world facts</td>
<td></td>
</tr>
</tbody>
</table>

© Gerrit Muller
Personal characteristics of a System Architect

Typical growth of a Architect

Generalist vs Specialist

Complementary Roles

Role Spectrum
Module Requirements

by Gerrit Muller       HBV-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

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

<table>
<thead>
<tr>
<th>Requirements describing the needs of the customer: Customer Needs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements describing the characteristics of the final resulting product: Product Specification</td>
</tr>
<tr>
<td>The requirements management process recursively applies definition 2 for every level of decomposition.</td>
</tr>
<tr>
<td>Requirements describing the needs of the company itself over the life cycle: Life Cycle Needs</td>
</tr>
</tbody>
</table>
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**
- How are the subsystems we will realize?
  - What are the subsystems we will realize?
    - What are the subsystems we will realize?
      - What are the subsystems we will realize?
        - What are the subsystems we will realize?
          - How will the subsystems be realized?
            - How will the subsystems be realized?
              - How will the subsystems be realized?
                - 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
Stakeholders w.r.t. Requirements

- **Customer**
  - **(purchaser, decision maker, user, operator, maintainer)**

- **Company**
  - **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)**
The “Formal” Requirements for Requirements

- 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.
The “CAFCR” model

What does Customer need in Product and Why?

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

drives, justifies, needs

enables, supports
Integrating CAFCR

What does Customer need in Product and Why?

Customer

What

Customer objectives

How

Application

Product

What

Functional

Conceptual

Realization

context

understanding

intention

objective driven

opportunity

Constraint awareness

knowledge based

Product How
CAFCR can be applied recursively

System (producer)
Customer
Business
Enables
Consumer Drives
Enables
Customer's Business
Drives
Customer Business Drives
Enables
Value Chain
larger scope has smaller influence on architecture
## Market segmentation

<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?

- CFO: Chief Financial Officer
- CIO: Chief Information Officer
- CMO: Chief Marketing Officer
- CEO: Chief Executive Officer
- CTO: Chief Technology Officer
- purchaser
- maintainer
- operator
- user
- department head
- decision maker(s)
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

Gerrit Muller

version: 0.4
March 9, 2015
BCAFCRplusLifeCycle
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

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** where requirements may have multiple drivers by means of brainstorming and discussions
- **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

Customer

What

Customer objectives

Key (Customer) Drivers

Derived

Application Drivers

Customer

How

Application

Requirements

Product

What

Functional

goal

means

may be skipped or articulated by several intermediate steps

functions

interfaces

performance figures
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)

**bottom-up**

- regulations
- "ideal" reference design
- prototyping, simulation
  (learning vehicle)
- bottom-up
  (technological opportunities)
- existing systems

Continued Product Creation Process

Feedback
Requirement Selection Process

- strategy
- roadmap
- competition
- product specification
- need characterization
- requirement phasing
- Technology, People, Process
- costs and constraints

customer needs
operational needs

Requirements Elicitation and Selection
version: 0
March 9, 2015
REQselection
Simple Qualification Method

- Important
  - Discuss
  - Don't discuss

- Urgent
  - Do
  - Discuss

- Effort
  - Don't discuss
  - Discuss

- Value
  - Do
  - Discuss
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)
• Determine the key drivers for one particular product family.
• Translate these drivers into application drivers and derive from them the requirements.
Needs and Requirements

Needs, Specification, Requirements

- Requirements describing the needs of the customer: **Customer Needs**
- Requirements describing the characteristics of the final resulting 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?
  - What are the subsystems we will realize?
  - How will the subsystems be realized?

- **How**
  - up to "atomic" components

**Requirements for Requirements**

- Specific
- Unambiguous
- Verifiable
- Quantifiable
- Measurable
- Complete
- Traceable

**Enable Human Use**

- Accessible
- Understandable
- Low threshold
CAFCR, Customer Key Driver Graph

CAFCR+ Model

Customer objectives
Application
Functional
Conceptual
Realization

Life cycle: operations, maintenance, upgrades
Development: manufacturing, installation

sales, service, logistics, production, R&D

Example Key Driver Graph

Key-drivers
- Safety
  - Reduce accident rates
  - Enforce law
  - Improve emergency response
  - Reduce delay due to accident
  - Improve average speed
  - Improve total network throughput
  - Optimize road surface
  - Speed up target groups
  - Anticipate on future traffic condition
- Effective Flow
  - Ensure traceability
  - Ensure proper alarm handling
  - Ensure system health and fault indication

Derived application drivers
- Early hazard detection
  - Warning and signaling
  - Maintain safe road condition
  - Detect and warn non-compliant 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
- Decoding
- Traffic condition dependent speed control

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

Complementary Viewpoints

Exercise Requirements Capturing
95  Gerrit Muller
version: 0
March 9, 2015
Module System Architect Toolkit

by Gerrit Muller    HBV-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

This module addresses tools and techniques available to the System Architect. It explains the basic CAFCR method and addresses story telling as method.

Distribution

This article or presentation is written as part of the Gaudi project. The Gaudi 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.

March 9, 2015
status: concept
version: 1.4
Abstract

The challenge for the architect is to cover a wide range of subjects, with many unknowns and uncertainties, while decisions are required all the time. The basic working methods, such as viewpoint hopping, modelling, handling uncertainties and WWHWW questions are described.
The seemingly random exploration path

thinking path of an architect during a few minutes up to 1 day
Scanning modes of the architect

open perceptive scanning

while structuring and judging

drunkard's walk
the world is full of interesting needs, technologies, ...

bad

straight for the goal
ignore everything that is not contributing directly to the goal

goal

good

bad
Combined open perceptive and goal oriented scanning

room for open perceptive exploration

increasing goal orientation

time
Coverage of problem and solution space

- covered or touched by architects
- covered by engineers and experts
- level of detail
- subjects
Successive quantification refinement

- Back of the envelope
- Benchmark, spreadsheet calculation
- Measure, analyze, simulate
- Cycle accurate

Order of magnitude

Guestimates

Calibrated estimates

Feasibility measure, analyze, simulate

Cycle accurate

Basic Working Methods of a System Architect
105    Gerrit Muller
Example evolution of quantification

- Measurement
- Design
- Estimate and uncertainty
- Specification
- Incomplete understanding
- Calibration
- Input
- Robustness problem
- Finished product
- Degradation performance

Basic Working Methods of a System Architect
Gerrit Muller

version: 1.5
March 9, 2015
BWMA:quantificationInTime
Quantified understanding of wafer stepper overlay

- Reticule: 15 nm
- Lens matching: 25 nm
- Single machine: 30 nm
- Stage overlay: 12 nm
- Stage grid accuracy: 4 nm
- Alignment repro: 3 nm
- Position accuracy: 7 nm
- System adjustment accuracy: 2 nm
- Off axis position accuracy: 4 nm
- Global alignment accuracy: 6 nm
- Stage Al. pos. meas. accuracy: 4 nm
- Blue align sensor repro: 3 nm
- Frame stability: 2.5 nm
- Tracking error WS: 2 nm
- Tracking error RS: 1 nm
- Tracking error: 75 nrad
- Metrology stability: 5 nm
- Process dependency sensor: 5 nm
- Matched machine: 60 nm
- Process overlay: 80 nm
Architect focus on important issues

- 80% architecting time
- 20% other issues

- 10% most important and most critical issues
- 90% all other issues

- 10% new issues
- 90% solved issues
Architect “worry” list

1. response time
from key press
until first image
on display

2. cost price
resource budgets

3. layering to separate
separation of concerns
self sustained
life-cycle separation
robust: paranoia validations

4. reliability of storage
5. database redesign
6. integration schedule
7. movement artefact
8. standby power
9. weak signal handling
10. location-based twiddle
A model is a simplified representation of part of the real world used for:

communication, documentation analysis, simulation, decision making, verification
Some examples of models

**formal analytical model**

\[ t_{\text{processing}} = t_{\text{overhead}} + n_{\text{rows}} \times t_{\text{row}} + n_{\text{row}} \times n_{\text{col}} \times t_{\text{pixel}} \]

**synchronization model**

Req

Ack

Strobe

feedback frequency: 4 kHz (0.25 msec)

**value chain model**

consumer

retailer

box-maker

semiconductor supplier

service provider

content provider

wooden model

mockup

model of coordinate system

6 degrees of freedom

6 degrees of freedom
<table>
<thead>
<tr>
<th>Types of models</th>
<th>Types of models</th>
</tr>
</thead>
<tbody>
<tr>
<td>mathematical</td>
<td>visual</td>
</tr>
<tr>
<td>linguistic</td>
<td>informal</td>
</tr>
<tr>
<td>formal</td>
<td>qualitative</td>
</tr>
<tr>
<td>quantitative</td>
<td>global</td>
</tr>
<tr>
<td>concrete</td>
<td>abstract</td>
</tr>
<tr>
<td>accurate</td>
<td>approximate</td>
</tr>
<tr>
<td>executable</td>
<td>read only</td>
</tr>
<tr>
<td>rational</td>
<td>intuitive</td>
</tr>
</tbody>
</table>
Questions

Why
What
How
Who
When
Where
Why broadens scope, How opens details
Flow from problem to solution

1. Problem understanding by exploration and simple models

2. Analysis by
   + exploring multiple propositions (specification + design proposals)
   + exploring decision criteria (by evaluation of proposition feedback)
   + assessment of propositions against criteria

3. Decision by
   + review and agree on analysis
   + communicate and document

4. Monitor, verify, validate by
   + measurements and testing
   + assessment of other decisions

vague problem statement

conflicting other decision

insufficient data

no satisfying solution

invalidated solution
### Multiple propositions

<table>
<thead>
<tr>
<th></th>
<th>20 p/m</th>
<th>high-performance sensor</th>
<th>350 ns</th>
<th>9 m/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>throughput</td>
<td></td>
<td>high-speed moves</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost</td>
<td>5 k$</td>
<td>additional pipelining</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safety</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*low cost and performance 1

<table>
<thead>
<tr>
<th></th>
<th>20 p/m</th>
<th>high-performance sensor</th>
<th>300 ns</th>
<th>10 m/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>throughput</td>
<td></td>
<td>high-speed moves</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost</td>
<td>5 k$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>safety</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*low cost and performance 2

<table>
<thead>
<tr>
<th></th>
<th>25 p/m</th>
<th>high-performance sensor</th>
<th>200 ns</th>
<th>12 m/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>throughput</td>
<td></td>
<td>high-speed moves</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost</td>
<td>7 k$</td>
<td>additional collision detector</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safety</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*high cost and performance*
## Assessment of propositions

<table>
<thead>
<tr>
<th>criterions</th>
<th>criterion weight</th>
<th>low cost and performance 1</th>
<th>low cost and performance 2</th>
<th>high cost and performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>throughput</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>cost</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>safety</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>future proof</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>effort</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>dev. time</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>risk</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>maintenance</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Recursive and concurrent application of flow

1. Problem understanding
2. Analysis
3. Decision
4. Monitor, verify, validate

Legend:
- decision flow
- analysis flow

Basic Working Methods of a System Architect
118 Gerrit Muller

version: 1.5
March 9, 2015
TORrecursion
Exploration by rapid iteration

system level  detail level

Basic Working Methods of a System Architect
119  Gerrit Muller
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

Product
What

Product
How

C
ustomer objectives

A
pplication

F
unctional

C
ceptual

R
ealization

Story How To
121        Gerrit Muller

version: 1.1
March 9, 2015
SHTfromStoryToDesign
A day in the life of Bob

bla blah bla, rabarber music bla bla composer bla bla qwewetty20 zeps.
nja nja njet njippie est quo vadis? Pjotr jaleski bla bla bla.binae fgfg gis hgrg
mjm 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 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, ...

---

**Application**

**Conceptual**

**Realization**

**Customer objectives**
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
**Personal multi media appliance**

Create a story for a personal multi media appliance. Derive a case description from the story, with functions and quantitative requirements.
Architect Way of Working

Viewpoint Hopping

Chaotic Path

Perceptive vs Judging

Varying Depth

Exercise System Architect Toolkit
128  Gerrit Muller
Some Architecting Means

Quantification and Margins

Focus on Key Issues

Phased Problem Solving

Story Telling

Exercise System Architect Toolkit
129  Gerrit Muller

version: 1.1
March 9, 2015
Module Roadmapping

by Gerrit Muller       HBV-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

This module addresses roadmapping.
Abstract

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

- **Market**
  - Application
  - Functional
  - Conceptual
  - Realization

- **Technology**
  - Products

- **People**

- **Process**

- **Time, ca 5 years**
Granularity of Roadmap Material

**Top-level roadmap**
- Single page
- Poster
- Part of many presentations

**Supporting roadmaps**
- Single page per view or per driver
- Poster
- Part of many presentations

**Supporting reports**
- Document per relevant subject
Problems that Occur without Roadmapping

- Frequent changes in product policy
- Late start up of long lead activities, such as people recruitment and process change
- Diverging activities of teams
- Missed market opportunities
Management with a Limited Horizon

2012  2013  2014

- horizon

now | feature |

Feature still unknown

- horizon

Now | feature |

Do!

- horizon

Now | feature |

Stop

- horizon

Now | feature |

Do!
Creation or Update of Roadmap in Burst Mode

Collective meeting ca 2 days

preparation by expert teams

2 weeks to digest and prepare

Roadmapping
137  Gerrit Muller

version: 2.0
March 9, 2015
ROADbursts
Typical Stakeholders of a Roadmap

- business manager
- marketing manager(s)
- people, process, and technology manager(s)
- operational manager(s)
- architect(s)
- overall enterprise responsible
- discipline or line managers
- project or program managers
Target of the First Session

Shared vision on market

First iteration of possible products as an answer to the market

Share technology status, as starting point for technology roadmap

Explore people and technology status, to identify main issues
Target of the Second Session

Obtaining a shared vision on the desired technology roadmap

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

Analyzing a few scenarios for products, technologies, people, and process
The Roadmap Update Visualized in Time

**Market:** What is needed by the customers?

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

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

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

**Process:** What processes are required to let these people realize the products and technologies?
From Roadmap to Detailed Plans

201X

Q2 Q3 Q4

roadmap n

roadmapping

Q1 Q2 Q3 Q4

201Y

roadmap n + 1

Policy and Planning Process

business plan:
budget & allocation

budget

Q1 delta

Q2 delta

Q3 delta

budget

Q1 delta

detailed planning

market events

tech hurdle

detailed planning

market events

tech hurdle

Product Creation Process
### 3-Tier Approach

<table>
<thead>
<tr>
<th>horizon</th>
<th>update</th>
<th>scope</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>roadmap</td>
<td>5 years</td>
<td>1 year</td>
<td>portfolio</td>
</tr>
<tr>
<td>budget</td>
<td>1 year</td>
<td>3 months</td>
<td>program</td>
</tr>
<tr>
<td>detailed plan</td>
<td>1 mnth-1yr</td>
<td>1 day-1 mnth</td>
<td>program or activity</td>
</tr>
</tbody>
</table>
Selection of most important or relevant issues

Key drivers as a means to structure the roadmap

Nothing is certain; ambiguity is normal

Use facts whenever possible

Don’t panic in case of impossibilities
Recognizable issues for all stakeholders

Clear positioning in time; uncertainty can be visualized

The main events (enabling or constraining) must be present

Limited amount of information to maintain the overview
# Sources of Facts

<table>
<thead>
<tr>
<th>Category</th>
<th>Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market analysis reports</td>
<td>number of customers, market size, competition, trends</td>
</tr>
<tr>
<td>Installed base</td>
<td>change requests, problem reports, historical data</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>statistical process control</td>
</tr>
<tr>
<td>Suppliers</td>
<td>roadmaps, historical data</td>
</tr>
<tr>
<td>Internal reports</td>
<td>technology studies, simulations</td>
</tr>
</tbody>
</table>
Causes for Overestimation

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

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

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

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

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

Too many bells and whistles without business or customer value
Abstract

The lifecycle of a product category in the market determines many aspects of the architecting approach. The lifecycle consists typical of 4 phases: infancy, adolescence, mature and aging.

A discontinuity in market success is seen in the transition from one phase to the next phase. The explanation given is that the phases differ in characteristics and require different approaches. The right approach for one phase is sub optimal for the next phase. A set of characteristics per phase is given and the consequences for architecting are discussed.
Ideal Bathtub Curve

Market Product Life Cycle Consequences for Architecting
version: 1.2
March 9, 2015
MPLifecycleGraphIdeal
Market Product Life Cycle Phases in Practice

- Infancy
- Adolescence
- Maturity
- Aging

Ideal "bathtub" curve

Product unable to make transition

Sales volume vs. time
Examples of Product Classes on the Curve

- Infant
  - functional MRI
  - digital TV

- Adolescence
  - DVD+RW
  - flat TV

- Maturity
  - MRI scanner
  - DVD

- Aging
  - X-ray systems
  - VCR
  - TV

Market Product Life Cycle Consequences for Architecting

version: 1.2
March 9, 2015
MPLifecycleGraphExamples
## Attributes per Phase

<table>
<thead>
<tr>
<th></th>
<th>Infancy</th>
<th>Adolescence</th>
<th>Mature</th>
<th>Ageing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Driving factor</strong></td>
<td>Business vision</td>
<td></td>
<td>Stable business model</td>
<td>Harvesting of assets</td>
</tr>
<tr>
<td><strong>Value from</strong></td>
<td>Responsiveness</td>
<td>Features</td>
<td>Refinements / service</td>
<td>Refining existing assets</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td>Discovery</td>
<td>Select strategic</td>
<td>Prioritize</td>
<td>Low effort high value only</td>
</tr>
<tr>
<td><strong>Dominant technical concerns</strong></td>
<td>Feasibility</td>
<td>Scaling</td>
<td>Legacy</td>
<td>Lack of product knowledge</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Obsolescence</td>
<td>Low effort for obsolete technologies</td>
</tr>
<tr>
<td><strong>Type of people</strong></td>
<td>Inventors &amp; pioneers</td>
<td>Few inventors &amp; pioneers &quot;designers&quot;</td>
<td>&quot;Engineers&quot;</td>
<td>&quot;Maintainers&quot;</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Chaotic</td>
<td></td>
<td>Bureaucratic</td>
<td>Budget driven</td>
</tr>
<tr>
<td><strong>Dominant pattern</strong></td>
<td>Overdimensioning</td>
<td>Conservative expansion</td>
<td>Midlife refactoring</td>
<td>UI gadgets</td>
</tr>
</tbody>
</table>
From Market, Product, Technology to People, Process

<table>
<thead>
<tr>
<th>Customer objectives</th>
<th>Market</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conceptual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Realization</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>Orion</th>
<th>Gemini</th>
<th>Scorpion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research</td>
<td>1</td>
<td>2</td>
<td>54</td>
</tr>
<tr>
<td>Maintenance</td>
<td>2</td>
<td>10</td>
<td>22</td>
</tr>
<tr>
<td>Total</td>
<td>83</td>
<td>133</td>
<td>136</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orion</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>30</td>
<td>50</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>Gemini</td>
<td>2</td>
<td>10</td>
<td>20</td>
<td>40</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70</td>
<td>50</td>
<td>20</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Scorpion</td>
<td>54</td>
<td>80</td>
<td>60</td>
<td>30</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70</td>
<td>50</td>
<td>20</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>maintenance</td>
<td>22</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>42</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>50</td>
<td>55</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>83</td>
<td>133</td>
<td>136</td>
<td>137</td>
<td>125</td>
<td>124</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>estimate by program manager</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orion</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>30</td>
<td>50</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>Gemini</td>
<td>2</td>
<td>10</td>
<td>20</td>
<td>40</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70</td>
<td>50</td>
<td>20</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Scorpion</td>
<td>54</td>
<td>80</td>
<td>60</td>
<td>30</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70</td>
<td>50</td>
<td>20</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>maintenance</td>
<td>22</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>42</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>50</td>
<td>55</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>83</td>
<td>133</td>
<td>136</td>
<td>137</td>
<td>125</td>
<td>124</td>
</tr>
</tbody>
</table>

feedback

homework

after iteration

version: 1.2
March 9, 2015
RSPfromMPToPP
Exercise Roadmapping
154  Gerrit Muller

Summary of strategy process

- Mission
  - Business specific, but
  - Open and generic

- Vision
  - Sharpen

- Focus
  - Input

- Inputs:
  - Market
  - Products
  - Technology
  - People
  - Process

- Forecasted facts
- Educated scenarios
- Estimates

- Roadmap
  - Input for
  - Committal plan
  - Reality facts
  - Empowerment

- Reality facts

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

The Roadmap Integrates Five Views

- Market
- Products
- Technology
- People
- Process

Multiple Levels

- Top-level roadmap
  - Single page
  - Poster
  - part of many presentations
- Supporting roadmaps
  - Single page per view or per driver
  - Poster
  - part of many presentations
- Supporting reports
  - Document per relevant subject

Creation in Teams

- Market: What is needed by the customers?
- Products: How to package technologies into products to fulfill market needs?
- Technology: What technological trends are relevant? What technologies are needed?
- People: What kind of and how many people are required to realize the products and technologies?
- Process: What processes are required to let these people realize the products and technologies?
**Time Horizons**

![Time Horizons diagram]

**Life Cycle Transitions**

![Life Cycle Transitions diagram]

**People and Process**

![People and Process diagram]
Module Product Families and Generic Developments

by Gerrit Muller       HBV-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

This module addresses product families and generic developments.
Abstract

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

In practice many difficulties pop up when product developments become coupled, due to the partial developments which are shared. This article discusses the advantages and disadvantages of a family approach based on shared developments and provides some methods to increase the chance on success.
Typical Examples of Generic Developments

- Platform
- Common components
- Standard design
- Framework
- Family architecture
- Generic aspects, functions, or features
- Reuse
- Products (in project environment)
Claimed Advantages of Generic Developments

- Reduced time to market
- Reduced cost per function
- Improved quality
- Improved reliability
- Improved predictability
- Easier diversity management
- Increases uniformity
- Employees only have to understand one base system
- Larger purchasing power
- Means to consolidate knowledge
- Increase added value
- Enables parallel developments of multiple products
- “Free” feature propagation

Building on shared components
Build every function only once
Maturing realization
Modularity
Economy of scale
Less learning
Not reinventing existing functionality
Product-to-product or project-to-project
<table>
<thead>
<tr>
<th>bad</th>
<th>good</th>
</tr>
</thead>
<tbody>
<tr>
<td>longer time to market</td>
<td>reduced time to market</td>
</tr>
<tr>
<td>high investments</td>
<td>reduced investment</td>
</tr>
<tr>
<td>lots of maintenance</td>
<td>reduced (shared) maintenance cost</td>
</tr>
<tr>
<td>poor quality</td>
<td>improved quality</td>
</tr>
<tr>
<td>poor reliability</td>
<td>improved reliability</td>
</tr>
<tr>
<td>diversity is opposed</td>
<td>easier diversity management</td>
</tr>
<tr>
<td>lot of know how required</td>
<td>understanding of one base system</td>
</tr>
<tr>
<td>predictable too late</td>
<td>improved predictability</td>
</tr>
<tr>
<td>dependability</td>
<td>larger purchasing power</td>
</tr>
<tr>
<td>knowledge dilution</td>
<td>means to consolidate knowledge</td>
</tr>
<tr>
<td>lack of market focus</td>
<td>increase added value</td>
</tr>
<tr>
<td>interference</td>
<td>enables parallel developments</td>
</tr>
<tr>
<td>but integration required</td>
<td>free feature propagation</td>
</tr>
</tbody>
</table>
Successful examples of reuse

<table>
<thead>
<tr>
<th>homogeneous domain</th>
<th>cath lab</th>
<th>MRI</th>
<th>television</th>
<th>waferstepper</th>
</tr>
</thead>
<tbody>
<tr>
<td>hardware dominated</td>
<td>car</td>
<td>airplane</td>
<td>shaver</td>
<td>television</td>
</tr>
<tr>
<td>limited scope</td>
<td>audio codec</td>
<td>compression library</td>
<td>streaming library</td>
<td></td>
</tr>
</tbody>
</table>
Limits of successful reuse

struggle with integration/convergence with other domains

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

poor/slow response on paradigm shifts

TV: LCD screens
cath lab: image based acquisition control

software maintenance, configurations, integration, release

MRI: integration and test
wafersteppers: number of configurations

how to innovate?
Drivers for Generic Developments

Customer value
- application adaptability
- availability variations
- new features originating from different products
- timely availability
- reliability
- quality increase
- maturity
- predictability
- availability integrated base product

Internal benefits
- asset creation
- increase economy of scale

Extrovert driver
Introvert driver

Product Families and Generic Aspects
version: 2.3
March 9, 2015
Gerrit Muller
Granularity of generic developments shown in 2 dimensions

Delegated integration

Shared integration

system
platform
subsystem
module
component

actual integration level

intended integration level

system
component
subsystem
platform
module

EV
R/F

CV
MIP
EVM

Generator

flat detector

CCD
Modified Process Decomposition

Customer-Oriented Process
Product Creation Process
Policy and Planning Process
Shared Assets Creation Process
People, Process, and Technology Management Process

Customer Roadmap → Business Drivers
Budget, plan → Product roadmap
Material → Needs & Feedback
Product-related processes → People Technology Process
Product needs and feedback → Technical Product Doc.
Needs & Feedback → Policy and Planning Process

People, Process, and Technology Management Process

Product Families and Generic Aspects
167 Gerrit Muller
version: 2.3 March 9, 2015 GDprocessDecompositionFamily
Financial Viewpoint on Process Decomposition

Management

Cashflow generation

Tomorrow's Cashflow

Strategic Asset Generation

Assets

Customer

Sales

Logistics

Production

Service

Presales

Information

Order

Support

Product Needs and feedback

Material

Needs and feedback

Technical Product Documentation

Product related processes

People

Technology

Process

Technology, Process and People roadmaps

Budgets

Product generics roadmap

Product roadmap

Budget, plan

Product Needs and feedback

Material

Technical Product Documentation

Product related processes

People Technology Process

People Technology

Assets

Product Families and Generic Aspects

version: 2.3
March 9, 2015

Gerrit Muller
Propagation Delay Platform Feature to Market

Product Families and Generic Aspects

version: 2.3
March 9, 2015
GDpropagationDelay
Sources of Failure in Generic Developments

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

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

Product Families and Generic Aspects

173 Gerrit Muller

version: 2.3 March 9, 2015 GDmodels
What are the top 3 benefits for your product family or generic development?
What are the top 3 disadvantages?
Harvesting Synergy

Contradicting Experiences

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

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

Drivers

Customer value
- application adaptability
- availability of accumulated feature set
- design for configurability

Internal benefits
- asset creation
- increase economy of scale

Extrovert driver
- timeliness
- availability variations
- new features originating from different products
- shared architectural framework

Introvert driver
- reliability
- quality increase
- predictability
- maturity
- availability integrated base product

Shared Asset Creation Process

Longer Chains

Exercise Product Families and Generic Developments

version: 2.3
March 9, 2015
Some Architecting Means

Organizational Complexity

Delay to Market

Pitfalls

Successful and Failing Models

Exercise Product Families and Generic Developments

version: 2.3
March 9, 2015

176  Gerrit Muller
Module Supporting Processes

by Gerrit Muller      HBV-NISE

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.

March 9, 2015
status: draft
version: 1.4
Granularity of Documentation

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

Abstract

The design of documentation is discussed, with emphasis on the requirements, the need for decomposition, the measures needed to maintain overview and criteria for granularity.
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
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
- **context** interacts with others
- **author** writes specification
- **specification** describes implementation
- **implementation** realizes consumer
- **consumer** uses
- **producer** realizes specification

Legend:
- **relation**
- **artifact**
- **stakeholder**
Decomposition of Large Documents

compound document

document structure

overview

document

document

document

document

document

Documentation Tree by Recursive Decomposition

Granularity of Documentation
Gerrit Muller

version: 1.2
March 9, 2015
DGdocumentRecursion
Payload: the Ratio between Content and Overhead

Granularity of Documentation

version: 1.2
March 9, 2015
DGpayload

Granularity of Documentation

188 Gerrit Muller
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
multiple related views

quantifications

one topic
per A3

capture "hot" topics

digestable
(size limitation)

practical
close to stakeholder experience

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

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

draft

- final review = final check contents
- consultation & review

concept

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

final review

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

authorized

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

authorization = check process

change request
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

Net change = \sum \text{all Forces} \induces \text{Reaction} \counteract \text{Support}
Template as Support for Process

- Principle → Drives → Process → Elaborated in Procedure → Supported by Formalism → Tool
- Abstract ← Specific and Executable
 Types of Templates

Recommended template type

layout only

meta information

prescribing contents

Header
Body
Footer

Title, Date
Body
Page, Author

Title, Date
Title, Date
Title, Date

Title
Author

Abstract

1 Introduction
2 Scope

Page, Author

3 Design

Page, Author

17 Interfaces

Page, Author

version: 1.6
March 9, 2015
THTTypes
<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>
<tr>
<td>• Use templates for</td>
<td>meta-information.</td>
<td></td>
</tr>
<tr>
<td>• Use checklists for</td>
<td>structure and contents.</td>
<td></td>
</tr>
</tbody>
</table>
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
Example Guidelines Meta Information

Mandatory per page:
- Author
- Title
- Status
- Version
- Date of last update
- Unique Identification
- Business Unit
- Page number
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

- Policy
  - Design
  - Requirements
  - Integrate
  - Test

- Requirements and specification
- Design
- Integrate
- Test

- Strategy
  - Feasibility
  - Definition
  - System design
  - Engineering
  - Integration & test
  - Field monitoring
  - 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
Alternatives to Integrate a Subsystem Early in the Project

- Existing subsystems (prototype)
- New subsystems
to-be-integrated subsystem

Physical environment:
- Physical reality
- Simulated environments

Virtual environment:
- Simulated subsystems
- Stubs

Spectrum:
- Simple
- Complex

System Integration How-To
215  Gerrit Muller
version: 0.2
March 9, 2015
SINTenvironments
## Stepwise Integration Approach

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></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

- correlate stage source
- correlate stage destination
- calibrate x,y measurement
- measure x,y source
- measure x,y destination
- control x,y destination
- position x,y source
- position x,y destination
- measure alignment signal
- adjust light source
- adjust lens
- align source destination
- focus
- measure
- process
- expose
- qualify
### Roles and Responsibilities During the Integration Process

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</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

supplier

Customer-Oriented Process

Product Creation Process

company

sales logistics production

Customer-Oriented Process

goods flow

Product Creation Process

Technical Product Documentation (TPD)

Purchasing Process

customer

Operation Process

life cycle

requirements

tender

order

product

orders

goods

specs

TPD

219 Gerrit Muller

System Integration How-To

version: 0.2
March 9, 2015
SINTprocessDecomposition
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.
## Requirements Entire Documentation

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

## Requirements per Document

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

## Decompose Large Documents

- Document structure
- Overview
  - Compound document
  - Document
  - Document
  - Document

## Recursive Decomposition

- Document structure
- Overview
  - Compound document
  - Atomic document
  - Document structure
  - Overview
Maximize Payload

A3s

Light Weight Review

Summary Module Supporting Processes

version: 0.2
March 9, 2015
Systems Integration

Integration Starts at Feasibility

Bottom-up

Alternatives for Early Integration

Propagation of Configuration Issues

Summary Module Supporting Processes

version: 0.2
March 9, 2015
Abstract

This module addresses the role of software in complex systems
The Role of Software in Systems

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

Abstract

Software is a dominating factor in the development of complex systems. It plays a crucial role in the performance of the final product at the one hand, while it contributes significant to the development cost and elapsed time of development. This paper will discuss the role of software in the broader system context. An improved understanding of the role of software enables the system architect, and the other stakeholders of the product creation process, to integrate the software development better. In this way hardware-software tradeoffs can be made, balancing performance, costs and risks.
Relative Contribution of SW

The Role of Software in Systems

2000

1970

100%

relative effort

physics/chemistry, etc.

mechanics

electronics

SW

version: 1.3
March 9, 2015
RSWrelativeEffort

Embedded Systems Innovation
Gerrit Muller
Mismatch between Role and Discipline

**role of software**
integration technology
captures *application* functionality
defines lot of *system* behavior
determines how much of potential *system* performance is achieved
acts as director

**focus of software discipline**
software technologies, such as:
  - programming languages
  - data bases
  - operating systems
  - component technologies
  - engineering practices

mismatch!
Control Hierarchy along Technology axis

- human user
- application SW
- control SW
- digital electronics
- analog or power electronics
- mechanical device
- optical device
- sensor

Legend:
- local automation or safety
The Role of Software in Systems

231     Gerrit Muller

version: 1.3
March 9, 2015
SWdisciplineCharacterization

Characterization of disciplines

- Mechanics
- Analogue / power Electronics
- Digital Electronics
- Software

Characteristics:
- concrete ↔ abstract
- tangible ↔ intangible
- mature ↔ immature
- production lead-time ↔ flexible
- material cost
## Quality Attributes annotated with SW relation

<table>
<thead>
<tr>
<th>Usable</th>
<th>Interoperable</th>
<th>Serviceable</th>
<th>Ecological</th>
</tr>
</thead>
<tbody>
<tr>
<td>usability</td>
<td>connectivity</td>
<td>serviceability</td>
<td>ecological footprint</td>
</tr>
<tr>
<td>attractiveness</td>
<td>3rd party extendible</td>
<td>configurability</td>
<td>contamination</td>
</tr>
<tr>
<td>responsiveness</td>
<td>liability</td>
<td>installability</td>
<td>noise</td>
</tr>
<tr>
<td>image quality</td>
<td>testability</td>
<td>disposability</td>
<td>disposability</td>
</tr>
<tr>
<td>wearability</td>
<td>liability</td>
<td></td>
<td>noise</td>
</tr>
<tr>
<td>storability</td>
<td>testability</td>
<td></td>
<td>disposability</td>
</tr>
<tr>
<td>transportability</td>
<td>standards</td>
<td></td>
<td>disposability</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dependable</th>
<th>Future Proof</th>
<th>Logistics Friendly</th>
</tr>
</thead>
<tbody>
<tr>
<td>safety</td>
<td>evolvability</td>
<td>manufacturability</td>
</tr>
<tr>
<td>security</td>
<td>portability</td>
<td>logistics flexibility</td>
</tr>
<tr>
<td>reliability</td>
<td>upgradability</td>
<td>lead-time</td>
</tr>
<tr>
<td>robustness</td>
<td>extendibility</td>
<td></td>
</tr>
<tr>
<td>integrity</td>
<td>maintainability</td>
<td></td>
</tr>
<tr>
<td>availability</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Effective</th>
<th>Consistent</th>
<th>3rd Party Extendible</th>
</tr>
</thead>
<tbody>
<tr>
<td>throughput or productivity</td>
<td>reproducibility</td>
<td></td>
</tr>
<tr>
<td>predictability</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Efficient</th>
<th>Logistics Friendly</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource utilization</td>
<td>manufacturability</td>
</tr>
<tr>
<td>cost of ownership</td>
<td>logistics flexibility</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logistics Friendly</th>
<th>Logistics Friendly</th>
</tr>
</thead>
<tbody>
<tr>
<td>lead-time</td>
<td></td>
</tr>
</tbody>
</table>

### Legend
- **weak SW relation**
- **strong SW relation**
Design Aspects related to SW

C: Customer objectives
A: Application
F: Functional
C: Conceptual
R: Realization

- design philosophy per quality attribute
  - performance, safety, security, ...
- granularity, scoping, containment, cohesion, coupling
- interfaces, allocation, budgets
- information model (entities, relations, operations)
- identification, naming
  - HAL_message_acknowledge_status versus ACK
- static characteristics, dynamic behavior
- system-level infrastructure
- software development process, environment, repository, and tools
- life cycle, configuration management, upgrades, obsolescence
- feedback tools, for instance monitoring, statistics, and analysis
- persistence
- licensing, SW-keys
- setup sequence, initialization, start-up, shutdown
- technology choices
- make, outsource, buy, or interoperability decisions

e.g., distributed or centralized control

The Role of Software in Systems
233  Gerrit Muller

version: 1.3
March 9, 2015
RSWdesignAspects
SW Mechanisms

Customer objectives  Application  Functional  Conceptual  Realization

error handling, exception handling, logging
processes, tasks, threads
configuration management; packages, components, files, objects, modules, interfaces
automated testing: special methods, harness, suites
signaling, messaging, callback scheduling, notification, active data, watchdogs, timeouts
locking, semaphores, transactions, checkpoints, deadlock detection, rollback
identification, naming, data model, registry, configuration database, inheritance, scoping
resource management, allocation, fragmentation prevention, garbage collection
persistence, caching, versioning, prefetching, lazy evaluation
licensing, SW-keys
bootstrap, discovery, negotiation, introspection
call graphs, message tracing, object tracing, etc.
distribution, allocation, transparency; component, client/server, multitier model
Describe the SW in a complex product, from different viewpoints for instance:

- Give an indication of the size/complexity
- Outline the SW architecture
- Identify the top 3 critical characteristics
- Identify potential improvements
- Process
- Development environment
Role of Software

- Integration technology captures application functionality and defines the system behavior.
- It determines how much of the potential system performance is achieved.
- Acts as a director.

Focus of software discipline:

- Software technologies, such as:
  - Programming languages
  - Data bases
  - Operating systems
  - Component technologies
  - Engineering practices

Control Hierarchy

- Human user
- Application SW
- Control SW
- Digital electronics
- Analog or power electronics
- Mechanical device
- Optical device
- Sensor

Discipline Characteristics

- Mechanics
- Analogue / power Electronics
- Digital Electronics
- Software

- Concrete
- Tangible
- Mature
- Production lead-time
- Material cost

- Abstract
- Intangible
- Immature
- Flexible?
Abstract

This module addresses the presentation of architectural issues to higher management teams.
Simplistic Financial Computations for System Architects.

by Gerrit Muller     HBV-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

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

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.

March 9, 2015
status: preliminary draft
version: 1.3
Product Margin = Sales Price - Cost

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

Cost per product, excluding fixed costs purchase price of components may cover development cost of supplier
Profit as function of sales volume

- **Sales Volume in units**
- **Income**
- **Expenses**
- **Fixed Costs**
- **Variable Costs**
- **Break Even Point**
- **Expected Sales Volume**

**Simplistic Financial Computations for System Architects.**

Gerrit Muller

version: 1.3
March 9, 2015
SFCprofitAndSalesVolume
Investments, more than R&D

financing

marketing, sales

training sales&service

NRE: outsourcing, royalties

research and development

business dependent: pharmaecutics 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

\[ \sum \text{income}_{\text{service}} \]

\[ \sum \text{sales price}_{\text{option}} \times \text{volume}_{\text{option}} \]

\[ \text{sales price}_{\text{product}} \times \text{volume}_{\text{product}} \]

- other recurring income
- services
- options, accessories
- products

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

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

version: 1.3
March 9, 2015
SFHockeyStick
What if ...?

- Early more expensive product + follow-on
- Delay of 3 months
- Original model

Simplistic Financial Computations for System Architects.

version: 1.3
March 9, 2015
SfHockeyStickWhatIf

Gerrit Muller
Stacking Multiple Developments

Simplistic Financial Computations for System Architects.

version: 1.3
March 9, 2015
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.

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.

March 9, 2015
status: concept
version: 0.1
Architectural issues related to managerial viewpoints

How to present architecture issues to higher management

version: 0.1
March 9, 2015
AMI introduction
Characteristics of managers in higher management teams

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

highly variable characteristics
? technology knowledge from extensive to shallow
? style from power play to inspirational leadership
How to prepare

Always prepare with small team!

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

How to present architecture issues to higher management
version: 0.1
March 9, 2015
AMIpreparation
Recommended content

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

supported by facts and figures
Mentioned info, shown info and backup info

**Market drivers**
- Cost
- TTM
- WOW
- DRM
  - Integration
  - Multiple suppliers
  - Nifty features
  - Fashionable design
  - Hollywood pact
  - Standards
  - MPEG4
  - MP3
  - ePen
  - GPS sensor
  - GSM
  - UTMS
  - BT
  - 802.11b

**Options**
- A
- B

**Typical performance**
- Transfer/s
- Load

**Bill of material**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infra</td>
<td>7</td>
</tr>
<tr>
<td>Sensor</td>
<td>6</td>
</tr>
<tr>
<td>Display</td>
<td>20</td>
</tr>
<tr>
<td>Power</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>36</td>
</tr>
</tbody>
</table>

**Schedule**

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

**ftes**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infra</td>
<td>2</td>
</tr>
<tr>
<td>Control</td>
<td>6</td>
</tr>
<tr>
<td>Display</td>
<td>6</td>
</tr>
<tr>
<td>Power</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>17</td>
</tr>
</tbody>
</table>

**Power budget**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infra</td>
<td>7</td>
</tr>
<tr>
<td>Sensor</td>
<td>6</td>
</tr>
<tr>
<td>Display</td>
<td>20</td>
</tr>
<tr>
<td>Power</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>36</td>
</tr>
</tbody>
</table>

**profit-investment**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Price</td>
<td>10</td>
</tr>
<tr>
<td>Cost</td>
<td>3</td>
</tr>
<tr>
<td>Sales</td>
<td>1M</td>
</tr>
<tr>
<td>Costs</td>
<td>10M</td>
</tr>
<tr>
<td>Investment</td>
<td>2M</td>
</tr>
<tr>
<td>Profit</td>
<td>5M</td>
</tr>
</tbody>
</table>

**Operating principle**

**worst-case performance**

**Power details**

**recommendation**
- Recommendation: select A
- Follow up: allocate Jan, Piet, Klaas per 1/11
- Go/no-go 1/1/03

**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

prepare in team of 4

13:30
14:00
15:00

present and discuss

1 1 2 2

3 3 4 4

16:00
17:00

feedback
Simplistic Financial Computations

Product Margin = Sales Price - Cost

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

Cost per product, excluding fixed costs.

Purchase price of components may cover development cost of supplier.

Profit as function of sales volume

Hockey stick and scenarios

intentionally left blank
Presentation to Management

Managerial Viewpoints

Prepare Content, Understand Audience

Always prepare with small team!

content 70% of effort
- gather facts
- perform analysis
- identify goal and message
- make presentation
- polish presentation form

understand audience 30% of effort
- gather audience background
- perform analysis
- identify audience interests
- simulate audience, exercise presentation

Show underlying info

Form, do and do not

poor form can easily distract from purpose and content

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

back-up material

backup material

presenter’s appearance
- well dressed
- self confident but open
- readable
- use demos and show artifacts

Summary Module Management Presentation

version: 0.1

March 9, 2015

Gerrit Muller
Abstract

The module Human Side addresses the psycho-social aspects of systems architecting.
Abstract

Systems architects interact quite often with many humans, and create products that must satisfy human needs. Insight in human aspects is crucial. However, human aspects span a very broad field, the human sciences, that differs quite significantly from the technical background of most architects.
Overview of Human Aspects

- **heterogeneous cultures**
  - individual
  - bilateral
  - group

- **homogeneous culture**
  - group

- **cultural diversity**
  - networked groups
  - networked society

- **number of involved humans**
  - psychology
  - psychiatry
  - group dynamics
  - cultural anthropology
  - physiology
  - pedagogy
  - didactics
  - political science
  - ergonomics
  - medicine
  - sociology
  - criminology
Context and Stakeholders of Product Creation

Competitors

Customer
Business

user

user

customer

Complementors

service

production

sales

Product Creation

internal company world

people

technology

Suppliers

Suppliers

The Human Side of Systems Architecting
264 Gerrit Muller

version: 1.0
March 9, 2015
StakeholderWorldView
A working group, consisting of

- Dieter Hammer (Technical University Eindhoven),
- Jaap van Rees (Van Rees adviesbureau),
- Jeroen van Hoven (Erasmus University Rotterdam),
- Kees van Overveld (Philips Research/TUE),
- Daan Rijsenbrij (Cap Gemini),
- Nathalie Masseus (Cap Gemini),
- and Gerrit Muller (Philips Research)

wants to increase the awareness in the ICT-architecture community of the human aspects.
Abstract

We discuss in this paper a set of skills and techniques to cooperate effectively between two individuals. We show the wonders of communication and then we address techniques such as investigation and acknowledgement, constructive feedback, conflict management, appraisal, good practices in a conversation, searching for ideas.
Active listening: the art of the receiver to decode the message

- **encoding**: based on emotional state, relation with the other, the objective, the situation, age, status, education, cultural background
- **verbal message**: from: "Listening and communicating" by Lia Charité, www.liacharite.nl
- **nonverbal message**: own interpretation of idea
- **decoding**: based on emotional state, relation with the other, the objective, the situation, age, status, education, cultural background

idea to be expressed

from: "Listening and communicating" by Lia Charité, www.liacharite.nl
Intense interaction needed for mutual understanding

to calibrate:
repeat many times with different examples, illustrations, and explanations

Human Side: Interpersonal Skills
268 Gerrit Muller

version: 0.1
March 9, 2015
CVCcodingCalibration
Mutual understanding as function of time

level of mutual understanding

intense interaction

no interaction

intense interaction

time
The material for interpersonal skills is based on a set of techniques from a course "Interpersonal Management Skills" by Hay Management Consultants in 1998.
investigate:
What has been said and why?

When a decision will be taken or an action will be started on the basis of exchanged information, opinions or suggestions or when the first reaction is to reject, ignore or contradict what you just heard.

acknowledge:
Paraphrase what has been said and why? i.e. use your own words
Constructive Feedback

How
+ Indicate the strong points to be kept
+ Indicate the points to be improved
+ Search for solutions which build upon the strong points and improve the weak points

When
You want to facilitate someone to improve his/her performance
Conflict Management

How?
define the positions:
* indicate what is important for you and why
* investigate and acknowledge what is important for the other and why

When
in case of conflict

option A
option B

When
in case of conflict

How?
define the positions:
* indicate what is important for you and why
* investigate and acknowledge what is important for the other and why

If you are willing and able to consider alternatives:
Search for alternative solutions

If you are not willing and able to consider alternatives, or no acceptable solution for both parties can be found:

IF

Finish the conversation:
* acknowledge the right to have a different opinion
* indicate your decision and why
When
Someone’s performance is important for you
* exceeding the expectations
* meets expectations continuously
* meets expectations, which exceed the normal performance level of this person

Appraise only when authentic!

How
+ Mention the performance very specific.
+ Mention the personal qualities which lead to this performance.
+ Describe which advantages arise for you, the department or the organization.
When you open a conversation
formulate the purpose

When you finish the conversation
summarize the agreements and the action plan
Searching for Ideas

When asking for a suggestion
When supplying a suggestion
When you use or build upon ideas of others
When you need new or more creative ideas
give a reaction
ask for a reaction
mention the source of the ideas
remove limitations temporarily or add limitations
Abstract

The creation of products requires many different people to cooperate. The work is often organized in teams. The team members have complimentary skills and knowledge. In many management courses the need to design teams is emphasized. Unfortunately, often these recommendations are ignored. We re-iterate in this paper the rationale for teams and the recommendations for designing the team itself.
Teams consist of complementary people

but in the team two can hear, two can see, and two can speak
Organization size and teams

1
2
4
8
16
32
64

128

256

512

room

floor

building

campus
Very simplistic team model

**Legend**
- productive work
- communication

1-person team: eff = 100%
2-person team: eff = 75%
3-person team: eff = 50%
4-person team: eff = 25%
Hierarchical simplistic team model

legend
- productive work
- communication

2-person team
eff = 75%

3-person team
eff = 66%

4-person team
eff = 62.5%

9-person team
eff ≈ 56%
Many personality and role models are available.

**Myers-Briggs Type Indicators**

- **Extraversion** (E) vs. **Introversion** (I)
- **Sensing** (S) vs. **iNtuition** (N)
- **Thinking** (T) vs. **Feeling** (F)
- **Judging** (J) vs. **Perceiving** (P)

**Belbin’s team roles**

- **plant**
  - creative
  - resource investigator
  - enthusiastic
  - communicator

- **team worker**
  - co-operative,
  - averts friction

- **implementer**
  - disciplined,
  - conservative,
  - do-er

- **shaper**
  - driver,
  - dynamic

- **completer finisher**
  - conscientious,
  - painstaking

- **co-ordinator**
  - mature,
  - chairman

- **monitor evaluator**
  - sober,
  - analytical

- **specialist**
  - single-minded,
  - rare skills

**Six thinking hats by Edward de Bono**

- **neutral**
  - facts

- **feeling**
  - instinctive

- **negative**
  - flaws

- **positive**
  - benefits

- **creative**
  - ideas

- **process**
  - meta
Process of creating and using a team

well-defined charter

What, When, Where, How, Whom

team owner
determines charter
to be respected by receivers

output

with sufficient room for the team to determine the way-of-working

team
“War Room” is very effective
Concurrency and Fragmentation lower efficiency

How many (semi-)concurrent tasks can a person handle?
Working in burst-mode (concentrating on one task for one day, week or month) can increase efficiency.

- six tasks in parallel: all results are late
- six tasks sequential: first result in 1/6 of time!
One person will be member of multiple teams

It is quite normal to participate in many teams simultaneously. However, a team can only function if the members are sufficiently available!
Critical Success Factors for teams

- well defined charter
- clear owner of the result
- respect for the output of the team
- freedom of way-of-working
- housing and location
- availability of team members
- complementary roles
- diversity, pluriformity
Abstract

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

version: 1.0
March 9, 2015

FPsystemArchitect

System Architect

communication
teamwork
documentation
multitasking
flexible, open
authority by expertise
specialist
generalist
conceptual
pragmatic
constructive critical
fast absorption of knowledge
creativity
manual skills
process insight
political insight
improvement
completeness
schedule
monitor progress
initial cost
decision making
customer value
sales features
commercial insight
coaching
selection
appraisal
motivation

decision making
monitor progress
schedule
initial cost
Operational Leader

Function Profiles; The Sheep with Seven Legs
292    Gerrit Muller
Commercial Manager

Function Profiles; The Sheep with Seven Legs
version: 1.0
March 9, 2015
FPcommercialManager
The numbers behind the bars

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>systems architect</td>
<td>9 8 9 9 9 9 3</td>
<td>9 9 7 9 9 8</td>
<td>3 7 7 5 3 5</td>
<td>2 5 8 8 4 4</td>
<td>6 5 3 7</td>
<td>test engineer</td>
<td>5 6 4 4 5 8 4</td>
<td>7 4 9 6 9 4</td>
<td>9 6 4 4 3 4</td>
<td>3 3 3 6 3 2</td>
<td>2 2 2 2 4</td>
<td>developer</td>
<td>6 8 7 4 6 6 9</td>
<td>3 6 8 6 5 9</td>
<td>7 6 5 4 7 7</td>
<td>6 8 4 5 3 2</td>
<td>2 2 3 2 2</td>
<td>operational leader</td>
<td>8 8 4 9 5 4 2</td>
<td>6 4 9 5 5 5</td>
<td>3 9 9 5 9 9</td>
<td>9 9 9 7 5 3</td>
<td>5 6 6 8</td>
<td>line manager</td>
<td>6 6 4 4 5 5 7</td>
<td>4 6 5 5 4 4</td>
<td>2 9 6 9 4 4</td>
<td>4 4 4 5 3 3</td>
<td>9 9 9 9</td>
</tr>
</tbody>
</table>
Make a (critical and honest) profile of yourself and of the operational or the line manager, who thinks he is managing you. 
Select 2 characteristics which you find difficult to assess or where you expect that other people will have a totally different perception. Discuss these 2 characteristics in the group. 
Present (max 1 flip) the highlights.
### Characteristics

|--------------------------|---------------|----------|---------------|--------------|------------------------|------------|------------|------------|-----------|--------------|---------|--------------------------------|------------|---------------|----------------|------------------|-------------|-------------|----------------|-------------|------------------|----------------|-------------|-----------------|-----------|-----------|-----------|------------|

---

**Exercise Psycho-Social Side**

297    Gerrit Muller  

version: 1.0  
March 9, 2015  
FPtableExercise
The Wonder of Bilateral Communication

- Idea to be expressed
- Encoding based on:
  - Emotional state
  - Relation with the other
  - The objective
  - The situation
  - Age, status
  - Education
  - Cultural background

- Decoding based on:
  - Emotional state
  - Relation with the other
  - The objective
  - The situation
  - Age, status
  - Education
  - Cultural background

Team work
- Deaf: cannot hear
- Blind: cannot see
- Mute: cannot speak

but in the team two can hear, two can see, and two can speak

Impact of Size

- 1 (room)
- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512 (campus)

Summary Module Human Side

version: 1.0
March 9, 2015
Teams

Role variations

Design the team!

Design team environment

Focus, avoid fragmentation

Summary Module Human Side
299    Gerrit Muller

version: 1.0
March 9, 2015
Function Profiles

Sheep with 7 Legs?

intentionally left blank

intentionally left blank

intentionally left blank

intentionally left blank
Wrap Up; module 10 SARCH

by Gerrit Muller       Buskerud University College

  e-mail: gaudisite@gmail.com

  www.gaudisite.nl

Abstract

This module addresses the Wrap Up of the course System Architecture
Abstract

Reflection facilitates the learning process. We discuss a simple reflection model and provide some means for reflection.
Colophon

Merete Faanes from Buskerud University College created the educational flow *Reflective Practice*. Reflective Practice is a thread throughout the entire master Systems Engineering to stimulate students to relate *Education and Practice*.

These workshops are the result of the cooperation of Merete Faanes and Gerrit Muller.
When to Reflect

- Reflection Before Action
  - anticipation
  - preparation

- Reflection In Action
  - concurrent

- Reflection On Action
  - retrospective

---

Reflection applied on Systems Architecting

version: 0
March 9, 2015
RASAtime
Scope: What to Reflect on

- operational or life cycle context
  - system of interest
    - component or function of interest

- organization
  - project
    - team
      - individual

- principle
  - process or method
    - procedure or technique
      - tool or notation

**technical**  **psychosocial**  **means**
Reflection Cycle

experiencing → reflecting → generalizing → applying →
observing → analyzing → interpreting → explaining →
testing → conceptualizing → observing

source: Kolb's learning cycle
http://www.infed.org/biblio/b-explrn.htm
Example of Reflection Questions

What stakeholders are involved?
What are their needs and concerns?
What is our goal?
How did we get in the current situation?
What is going well, what is going bad?
What approach can we take?
What do we expect to happen?
et cetera
Recommended Reflection Report Content

subject or goal

description of your experiences

analysis

lessons learned

actions as follow-up

avoid broad generic statements

illustrate with specific examples
Make a personal improvement “roadmap” (a many year vision) and a personal improvement plan (feasible and visible first steps).

- Identify needed improvements, which can be influenced by yourself.

- Determine what you need to do to trigger the improvement and whom needs to be involved.

- Try to link your improvements to the rest of the business, for instance to planned products, conferences, platform releases or whatever recognizable anchor is available.
SES A Homework Assignment

- after ~3 weeks:
  a powerpoint presentation with figures, diagrams and tables of the SESA views

- after ~6 weeks:
  - a concept report with updated figures, diagrams and tables.
  - Add some explanatory text in the report.
  - Maximum size of the report 20 pages; less is better

- after ~9 weeks:
  a complete report where the feedback on the concept report has been processed

- after 10 weeks:
  personal reflection, plan and roadmap.
Viewpoints

- process and organization; how does the product/system creation process work?
  Diagram of the **de facto operational organization** (e.g. like the Monday morning SESA exercise). Note: no nice looking official diagrams, rather the actual situation with names. This actual situation might differ from the theory. Reflect on these differences, and the consequences.

- role and task of the system architect
- requirements management; especially a **customer key driver graph** for your system
- system architect toolkit; give examples typical tools, techniques and methods as applied on your system, and provide a **story** for your system.

- roadmapping; make a coarse **roadmap of market, product and technology** for your part of the company (in a broader context than the system only); pay special attention to the "outside" world, e.g. relevant trends.

- generic developments/product families; show and reflect on how your company tries to address similarity between systems, projects or products

- supporting processes, especially documentation

- **presentation to management**, especially high level financial figures for your system. Submit this as a separate presentation. You may use the presentation of the course itself, with updates based on the board meeting. Provide reflection on the presentation: How was the presentation in retrospect? How did the BoM respond?

- role of software in your system (so not the tools that are used in your organization)

- psycho social side
Recommendations and Guidelines

- Make and communicate visualizations (diagrams, figures, models, graphs) first.

- Use this assignment as opportunity to talk with other people in your organization.

- Reflect in the tekst on the viewpoint and its actual status; what works well, what can be improved?

- Note the maximum size of 20 pages; smaller reports get better grades :-)

Exercise Wrap Up
312      Gerrit Muller

version: 0
March 9, 2015
MSWUhomeworkRecommendations
• in the personal plan and roadmap make sure that you relate these to your company; what does the company need and what do you want/what are your capabilities.

• the personal plan is short term oriented: what do you plan to do in the next days/weeks. Think about practical steps that allow you to learn and to earn credit.

• the personal roadmap is long term: where do you want to be in 3 to 5 years? How does this fit in your company? What steps are required?

• personal reflection max 1 A4, personal plan max 1 A4, personal roadmap max 1 A4.
Submission instructions

use for all deliverables the following conventions:

filename: SESA <your name> <subject> .<version>.<extension>

e.g. RP John Student preassignment My Role.2.doc

where subject = {report | plan | ...}

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

subject: SESA <subject>

"standard" file types preferred, e.g. pdf, jpg, doc, xls, ppt

submission deadline complete assignment: 10 weeks after end of course

Note: intermediate submissions are mandatory

Extension is only possible after permission from the teacher. Consequence of an extended deadline is that the grade may be registered in the formal HBV administration a year later.