CTT Course System Architecting SARCH for management teams

by Gerrit Muller  Buskerud University College

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

Abstract

Information about the course System Architecting for management teams.
Abstract

This article describes the condensed version the course System Architecture by the Center for Technical Training CTT. Trainer is the author of this article Gerrit Muller. At this moment this course is only accessible for Philips Employees. The course is set up to make the art of system architecting more accessible. The course will address a wide spectrum of issues in relation with system architecture, such as: processes, business, role and task of the system architect (team), generic Developments (re-use, platforms) requirements, roadmapping, skills, and psycho social factors.
<table>
<thead>
<tr>
<th>session</th>
<th>subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>day 1 morning</td>
<td>positioning the System Architecture Process</td>
</tr>
<tr>
<td></td>
<td>Product Creation Process</td>
</tr>
<tr>
<td></td>
<td>product families, generic developments</td>
</tr>
<tr>
<td>day 1 afternoon</td>
<td>role and task of the system architect</td>
</tr>
<tr>
<td></td>
<td>profile of the system architect</td>
</tr>
<tr>
<td></td>
<td>documentation, reviewing and other supportive processes</td>
</tr>
<tr>
<td>day 2 morning</td>
<td>requirements capturing, roadmapping</td>
</tr>
<tr>
<td>day 2 afternoon</td>
<td>HRM aspects; selection, appraisal, career path, etcetera</td>
</tr>
<tr>
<td></td>
<td>wrap up, expectations, how to continue, evaluation</td>
</tr>
</tbody>
</table>
Exercises

• 1 Make a map of the operational organization, from portfolio down to components, with specific products, names and roles; Discuss the relations in one of the core teams.

• 2 Role play, marketing manager + project leader + system architect + observer; prepare initial product definition (=business relevance+specification+critical design issues+plan indication)

• 3 Determine Requirements and key drivers and show the relationship

• 4 Show the roadmap as far as known now
Structure

Theory
  dull
  passive

Practical Illustration
  vivid
  passive

Interaction
  vivid
  active

Spin-off:
  cross-fertilization

Insight

Exercise

Abstraction

Course System Architecting for Management Teams
5  Gerrit Muller
Timing Template of one subject

<table>
<thead>
<tr>
<th>9:00</th>
<th>10:00</th>
<th>11:00</th>
<th>12:00</th>
<th>12:30</th>
</tr>
</thead>
<tbody>
<tr>
<td>13:30</td>
<td>14:00</td>
<td>15:00</td>
<td>16:00</td>
<td>17:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interactive Exploration</th>
<th>Broadcast</th>
<th>break</th>
<th>Broadcast</th>
<th>Interactive discussion</th>
<th>break</th>
<th>groupwork</th>
</tr>
</thead>
</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.

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.
Goals of the Gaudí Project

- Consolidate existing Systems Architecting Methods
  
  evaluate, reflect, generalize

- Make the Systems Architecting art more accessible
  
  case descriptions

- Enable the education of (future) System Architects
  
  curriculum, course material

- Research new or improved Systems Architecting Methods
  
  industry as laboratory
Modular approach
Show Early to Get Feedback

leading principle:
show documents under construction
but clearly show their status
Growth of the System Architect
Positioning Courses

The Gaudí Project

14    Gerrit Muller

version: 3.1
January 28, 2015
PEVOCposition
Positioning Books

root
technical
designer

technical
generalist

business, application insight

process insight

psycho-social
interests

Architecting
System
Performance

CAFCR
Architectural
Reasoning

Systems Architecting
Supporting Processes
Composable Architectures
Human Measure

System Modeling
and Analysis:
a Practical Approach

Gaudi
books

CAFCR
PhD thesis

version: 3.1
January 28, 2015
The Gaudí Project
15 Gerrit Muller
Productivity: number of new entries

<table>
<thead>
<tr>
<th>Year</th>
<th>SARCH</th>
<th>CAFCR</th>
<th>Evolvability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2003</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average productivity over the years is approximately 20.
Abstract

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

by Gerrit Muller       Buskerud University College

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

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

Policy and Planning Process

Product Creation Process

Customer-Oriented Process

People, Process, and Technology Management Process
Financial Characterization of Decomposition

- Customer Oriented Process
  - Sales
  - Logistics
  - Production
  - Service
  - Presales
- Product Creation Process
  - Policy and Planning Process
  - People and Technology Management Process
- Business Drivers
  - Customer Roadmap
  - Business Drivers
  - Product roadmap
  - Budget, plan
  - Product Needs and feedback
  - Order
  - Product
- Cashflow Generation
  - Support
  - Product-related processes
  - Technical Product Documentation
  - Product Needs and feedback
  - Material
- Tomorrow's Cashflow
  - Needs and Feedback
  - Technical Product Documentation
  - Product-related processes
- Assets
  - People and Technology Management Process
  - People Technology Process
  - Budgets
  - Technology, Process, and People roadmaps
- Management
  - Policy and Planning Process
  - Budget, plan
  - Product roadmap
  - People and Technology Management 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

Customer

Value

Feedback

Requirements and Feedback

Technical Product Documentation

Product related processes

Information

Order

Product

Support

Budgets

People

Technology

Process

Product Needs and feedback

material

Customer Roadmap

Business Drivers

Product roadmap

Budget, plan

Technical Product Documentation

Product roadmap

Budget, plan

Support

People

Technology

Process

Customer

$$

Business Drivers

Technology, Process, and People roadmaps

Budgets

Product Needs and feedback

Technical Product Documentation

Product roadmap

Budget, plan

Support

People

Technology

Process
Decomposition of the Customer Oriented Process

Customer-Oriented Process
Order Acquisition
Order Realization
Service Support
Material Order Product
Order
Product Support
Information

Process Decomposition of a Business
version: 1.1
January 28, 2015
Gerrit Muller
The Product Creation Process

by Gerrit Muller    Buskerud University College

 e-mail: gaudisite@gmail.com

 www.gaudisite.nl

Abstract

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

Distribution

This article or presentation is written as part of the 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.

January 28, 2015
status: concept
version: 2.2
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

Product roadmap

Technical Product Documentation

Product related processes

Requirements and Feedback

Support

Material

Presales sales logistics production service

Information

Order

Product

Budget

Product Requirements and feedback

Technology, Process and People roadmaps

People Technology Process

People Technology

Version: 2.2
January 28, 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

<table>
<thead>
<tr>
<th>Phase</th>
<th>Feasibility</th>
<th>Definition</th>
<th>System Design</th>
<th>Engineering</th>
<th>Integration &amp; Test</th>
<th>Field Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>needs</td>
<td></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>
<td></td>
</tr>
<tr>
<td>design</td>
<td></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>
<td></td>
</tr>
<tr>
<td>engineering</td>
<td></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

<table>
<thead>
<tr>
<th>Status</th>
<th>Full under development</th>
<th>Preparing or updating work</th>
</tr>
</thead>
</table>

The Product Creation Process
version: 2.2
January 28, 2015
The Product Creation Process
version: 2.2
January 28, 2015
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
The Product Creation Process

Version: 2.2
January 28, 2015

PCP Characteristics

Characteristics of a Phase Model

- Large impact decisions
- Phase transitions
- Check points
- Concurrency
- Iteration

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

- Needs
- Specification
- Design
- Verification
- Engineering

- Order long-lead items
- Order high-cost items
- Product announcement

- Large impact decisions
- Concurrency
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 cycle
Decomposition of the 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

The Product Creation Process

version: 2.2
January 28, 2015
PCPOperationalOrganization

The diagram illustrates the operational organization of the Product Creation Process (PCP) with roles and responsibilities defined at various levels:

- **Entire portfolio**: Portfolio operational manager
- **Product family**: Family operational manager, Family marketing manager, Family architect
- **Single product**: (Single product) project leader, Product architect, Product manager
- **Subsystem**: Subsystem project leader, Subsystem architect
- **Module**: Developers
Prime Responsibilities of the Operational Leader

- Specification
- Quality
- Resources
- Time

The Product Creation Process

version: 2.2
January 28, 2015
PCPoperationalTriangle
The Rules of the Operational Game

- define project
- update project
- assess risks
- determine feasibility
- accept or reject
- execute project within normal quality rules

business management

project leader
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

39  Gerrit Muller

version: 2.3
January 28, 2015
SAPprocessSimplified
System Architecting Relation between PPP and PCP

Context: Product Portfolio, Time

Policy and Planning Process → Product Creation Process

Vision, Policy, Intention

Practical Knowledge

Feedback from Reality
The System Architecture 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
- Functionality
- Qualities
- Synergy
- Specific solution

The System Architecture Process

Gerrit Muller

version: 2.3
January 28, 2015
SAPkeyIssues
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

intentionally left blank
PCP involves **all** disciplines, much more than D&E.

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

**Phased Process**

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

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

**Incremental Development**

- test and evaluate
- requirements specification
- build
- design

2% of budget (EVO)
2 weeks (XP)
up to 2 months per cycle
PCP Decomposition and Operational Management

PCP decomposition

**Product Creation Process**

- **Operational Management**
  - Specification
  - Budget
  - Time

- **Design Control**
  - Technical

- **Marketing**
  - Profitability
  - Sellability

Architecture at all levels; From portfolio to subsystem

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

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

- **Single Product**
  - (Single Product) Project Leader
  - Product Architect
  - Product Manager

- **Subsystem**
  - Subsystem Project Leader
  - Subsystem Architect

- **Module**
  - Developers

Operational Commitment

- **Specification**
- **Resources**
- **Time**

Core: Operational + Technical + Commercial

Exercise Product Creation Process

version: 2.3
January 28, 2015
System Architecture Process

In Business Context

Key Issues

5 Views

Exercise Product Creation Process
Gerrit Muller

version: 2.3
January 28, 2015
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

<table>
<thead>
<tr>
<th>Advantage</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduced time to market</td>
<td>building on shared components</td>
</tr>
<tr>
<td>Reduced cost per function</td>
<td>build every function only once</td>
</tr>
<tr>
<td>Improved quality</td>
<td></td>
</tr>
<tr>
<td>Improved reliability</td>
<td>maturing realization</td>
</tr>
<tr>
<td>Improved predictability</td>
<td></td>
</tr>
<tr>
<td>Easier diversity management</td>
<td>modularity</td>
</tr>
<tr>
<td>Increases uniformity</td>
<td></td>
</tr>
<tr>
<td>Employees only have to understand one base system</td>
<td>less learning</td>
</tr>
<tr>
<td>Larger purchasing power</td>
<td>economy of scale</td>
</tr>
<tr>
<td>Means to consolidate knowledge</td>
<td></td>
</tr>
<tr>
<td>Increase added value</td>
<td>not reinventing existing functionality</td>
</tr>
<tr>
<td>Enables parallel developments of multiple products</td>
<td></td>
</tr>
<tr>
<td>“Free” feature propagation</td>
<td>product-to-product or project-to-project</td>
</tr>
</tbody>
</table>

**Product Families and Generic Aspects**

- **Gerrit Muller**

*version: 2.3  January 28, 2015  GDclaims*
<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><strong>homogeneous domain</strong></th>
<th><strong>cath lab</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>MRI</strong></td>
</tr>
<tr>
<td></td>
<td><strong>television</strong></td>
</tr>
<tr>
<td></td>
<td><strong>waferstepper</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>hardware dominated</strong></th>
<th><strong>car</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>airplane</strong></td>
</tr>
<tr>
<td></td>
<td><strong>shaver</strong></td>
</tr>
<tr>
<td></td>
<td><strong>television</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>limited scope</strong></th>
<th><strong>audio codec</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>compression library</strong></td>
</tr>
<tr>
<td></td>
<td><strong>streaming library</strong></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
- predictability
- availability integrated base product
- asset creation
- increase economy of scale
- maturity

Internal benefits

Introvert driver

Extrovert driver

Internal benefits

Version: 2.3
January 28, 2015
GDdrivers
Granularity of generic developments shown in 2 dimensions

Delegated integration

Shared integration

system
platform
subsystem
module
component

generate
MIP
EVM
Generator
flat
detector
CCD

actual integration level
intended integration level

component
module
subsystem
platform
system
Modified Process Decomposition
Value and Feedback Flow

Policy and Planning Process

Customer-Oriented Process

Product Creation Process

Shared Assets Creation Process

People, Process, and Technology Management Process
Sources of Failure in Generic Developments

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

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

- **Lead Customer**
  - direct feedback
  - too specific?

- **Carrier Product**
  - product feedback
  - product specific?

- **Platform**
  - feedback problem
  - too generic

- **Technology Push**
  - no feedback

- **Product Creation Process**
  - customer oriented process (sales, service, production)

- **People and Technology Management Process**

- **Creating Generic Components**

- **Policy and Planning**

- **Suppling Business**

- **Customer**
Exercise Generic Developments

What are the top 3 benefits for your product family or generic development?
What are the top 3 disadvantages?
Contradicting Experiences

**good**
- reduced time to market
- reduced investment
- reduced (shared) maintenance cost
- improved quality
- improved reliability
- easier diversity management
- means to consolidate knowledge
- increased 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

- application adaptability
- availability of accumulated feature set
- design for configurability
- availability variations
- new features originating from different products
- shared architectural framework
- timely availability
- reliability
- quality increase
- predictability
- maturity
- asset creation
- increase economy of scale
- availability integrated
- base product

Shared Asset Creation Process

- Policy and Planning Process
- Customer-Oriented Process
- Product Creation Process
- Shared Assets Creation Process

Longer Chains

- Policy and Planning Process
- Customer-Oriented Process
- Product Creation Process
- Shared Assets Creation Process
Some Architecting Means

Organizational Complexity

Delay to Market

Pitfalls

Successful and Failing Models

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

Successful and Failing Models

Exercise Product Families and Generic Developments

version: 2.3
January 28, 2015

Gerrit Muller
Abstract

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

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

The Role and Task of the System Architect

version: 2.0
January 28, 2015
RSAdeliverables
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

module     subsystem     system

Requirement    Spec    Design    Realization

modules

Function    Quality

KISS

Elegance    Simple    Integrity    Fitting

satisfied stakeholders

context

system
<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>

Examples of Secondary Responsibilities
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

The Role and Task of the System Architect

Version: 2.0
January 28, 2015
RSAactivities
## From Detail to Overview

<table>
<thead>
<tr>
<th></th>
<th>Quantity per year (order-of-magnitude)</th>
<th>architect time per item</th>
</tr>
</thead>
<tbody>
<tr>
<td>driving views</td>
<td>10</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

**The Role and Task of the System Architect**

74 Gerrit Muller

version: 2.0
January 28, 2015
RSAdetailHierarchy
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

Gerrit Muller

version: 2.0
January 28, 2015

RSApyramid
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.
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
January 28, 2015
MATgeneralistVsSpecialist
Generalists and Specialists are Complementary

The diagram illustrates the complementary expertise between generalists and specialists. Generalists possess a broad range of knowledge, while specialists have deep knowledge in specific areas. The diagram suggests that a combination of these two types of expertise is necessary for comprehensive problem-solving and innovation.
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.
Architecting Interaction Styles

**provocation**
- when in an impasse: provoke
- effective when used sparsely

**facilitation**
- especially recommended when new in a field:
  - contribute to the team, while absorbing new knowledge
- provide vision and direction, make choices
- risk: followers stop to give the needed feedback

**leading**
- take the viewpoint of the stakeholder
- acknowledge the stakeholder's feelings, needs, concerns

**empathic**
- investiagte by asking questions

**interviewing**
- invite a few engineers and walk through the system operation step by step

**whiteboard simulation**
- first listen to the stakeholder and then
- explain cost and alternative opportunities

**judo tactics**
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
- Report
- Spec
- Design

**Responsibilities**
- Balance
- Consistency
- Decomposition
- Integration
- Overview
- KISS
- Elegance
- Simple
- Integrity
- Fitting

**Daily Activities**
- Ideas: think, analyze
- Planning: brainstorm, explain
- Project: design, assist project leader, work breakdown, schedule, risks
- Present: present, meet, teach, discuss
- Support: travel to customer, supplier, conference
- Provide vision and leadership
- Test, integrate
- Write, consolidate, browse
- Read, review

**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</td>
<td>10</td>
</tr>
<tr>
<td>meetings</td>
<td>10^2</td>
</tr>
<tr>
<td>informal contacts</td>
<td>10^3</td>
</tr>
<tr>
<td>scanning</td>
<td>10^4</td>
</tr>
<tr>
<td>product details</td>
<td>10^5 – 10^6</td>
</tr>
<tr>
<td>real-world facts</td>
<td>infinite</td>
</tr>
</tbody>
</table>
Personal characteristics of a System Architect

Typical growth of a Architect

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

Generalist vs Specialist

- specialist
- generalist

Complementary Roles

Role Spectrum

- aspect architect
- systems architect

Exercise Role and Task System Architect
86     Gerrit Muller

version: 0.2
January 28, 2015
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.
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
High cohesion (within the unit)
Low coupling (outside of the unit)
Accessibility for the readers
Low threshold for the reader
Low threshold for the author
Manageable steps to create, review, and change
Clear responsibilities
Clear position and relation with the context
Well-defined status of the information
Timely availability
Accessibility Requirements

Ease of reading, “juiciness”

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

Understandability

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

Reachability in a limited number of steps
Responsibility Requirements

- single author
- limited amount of reviewers
Scalability Requirements

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

- **Project leader** is responsible for time, budget, result.
- **Architect or editor** is responsible for technical.
- **Author** writes **specification** and **interprets** it.
- **Consumer** uses **implementation** and **interacts** with **context**.
- **Producer** realizes **artifact** and **interacts** with **context**.

Legend:
- **Relation**
- **Artifact**
- **Stakeholder**

Granularity of Documentation
95  Gerrit Muller

version: 1.2
January 28, 2015
DGdocumentationRoles
Decomposition of Large Documents

compound document

document structure

overview

document
document

document

document

document

Payload: the Ratio between Content and Overhead

Granularity of Documentation

version: 1.2
January 28, 2015
DGpayload
Abstract

LEAN product development is in the process and means area pragmatic. Low tech tools, such as paper, pen and magnets, with very direct interaction are used. For communication the use of single A3-size documents is promoted, because this is a manageable amount of information.
Characteristics of LEAN

A holistic, systems approach to product development including people, processes, and technology.

Multi-disciplinary from the early start, with a drive to be fact based.

Customer understanding as the starting point.

Continuous improvement and learning as cultural value.

Small distance between engineers and real systems, including manufacturing, sales and service and the system of interest.
Example of A3 Architecture Overview

A3 architecture overview of the Metal Printer (all numbers have been removed for competitive sensitivity)

- **Author:** Gerrit Muller
- **Version:** 0.1
- **Date Last Update:** August 3, 2010
- **Status:** Preliminary Draft

### Key Performance Parameters

**Customer key-drivers and Key Performance Parameters**

1. **Close doors**
2. **Align**
3. **Move to proximity**
4. **Process**
5. **Move substrate unloading position**
6. **Open doors**

**Formula print cycle time**

\[ t_{\text{print}} = t_{\text{prepare}} + t_{\text{align}} + t_{\text{chamber}}(\text{thickness}) + t_{\text{finalize}} \]

- **t_{\text{prepare}}** = t_{\text{close doors}} + t_{\text{move to proximity}}
- **t_{\text{align}}**
- **t_{\text{chamber}}**
- **t_{\text{finalize}}** = t_{\text{move to unload}} + t_{\text{open doors}}

**Notes:**
- Original diagram was annotated with actual performance figures for confidentiality reasons these numbers have been removed.
multiple related views
quantifications

one topic per A3
capture "hot" topics
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*.
micro specification control board

- very dynamic, many changes
- light weight review process

production

used by customers

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

draft
- final review = final check contents
- authorization = check process

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

authorized
- change request

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

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

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

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

The introduction of a new process (way of working) is quite often implemented by supplying ready-to-go tools and templates. This implementation mainly serves the purpose of a smooth introduction of the new process. Unfortunately the benefits of templates are often cancelled by unforeseen side-effects, such as unintended application, inflexibility, and so on. This intermezzo gives hints to avoid the Template Trap, so that templates can be used more effectively to support introduction of new processes.
Rationale for Templates

- Low threshold to apply a (new) process (1)
- Low effort to apply a (new) process (2)
- No need to know low level implementation details (3)
- Means to consolidate and reuse experiences (4)
Bogus Arguments for Templates

- Obtain a uniform look (5)
- Force the application of a (new) process (6)
- Control the way a new process is applied (7)
Forces of Change: Action = - Reaction

\[ \sum \text{all Forces} \longrightarrow \text{induces} \rightarrow \text{New Process} \]

\[ \text{Reaction} \quad \text{counteract} \quad \text{Support} \]

\[ \text{Net change} = \text{all Forces} \]
Template as Support for Process

principle \(\xrightarrow{\text{drives}}\) process \(\xrightarrow{\text{elaborated}}\) procedure \(\xrightarrow{\text{supported}}\) tool

abstract \(\xrightarrow{\text{is}}\) specific and executable

formalism

SAPabstractionHierarchy

version: 1.6
January 28, 2015
Types of Templates

- **Header**
- **Body**
- **Footer**

**recommended template type**

- **Title**
- **Author**
- **Body**
- **Page, Author**

layout only

- **Title, Date**
- **Body**
- **Page, Author**

meta information

- **Title, Author**
- **Body**
- **Page, Author**

prescribing contents

- **Title, Date**
- **1 Introduction**
- **2 Scope**
- **Page, Author**

- **Title, Date**
- **3 Design**
- **Page, Author**

- **Title, Date**
- **17 Interfaces**
- **Page, Author**

---

**Template How To**

Gerrit Muller

version: 1.6
January 28, 2015
THTypes
## Recommendation

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

- Use templates for meta-information.
- Use checklists for structure and contents.
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
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
- Requirements and specification
- Design
- Integrate & Test
- Product operational life cycle

1. Strategy
2. Feasibility
3. Definition
4. System design
5. Engineering
6. Integration & test
7. Field monitoring
8. 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
Integration Takes Place in a Bottom-up Fashion

- component
- subsystem
- system function
- product
- context

integrate
alpha test
Transition from Previous System to New System

1. **Existing Base System**
   - **New Base Systems**
     - **2 Partial Systems for SW Testing**
       - **SW Dev System**
         - **Test HW Subsystem**
         - **Test SW for New HW Subsystem**
         - **Integrate HW System**
       - **Existing Base System**
         - **Test and Refine Application**
         - **Integrate and Refine Application**
       - **Adopt Existing Base SW**
         - **Integrate System**

2. **New HW Subsystem**
   - **SW Dev System**
     - **Test HW Subsystem**
     - **Integrate Subsystem**
   - **Existing Base System**
     - **Test SW for New HW Subsystem**
     - **Integrate System**
   - **Adopt Existing Base SW**
     - **Integrate System**

3. **New Application**
   - **Test and Refine Application**
   - **Integrate and Refine Application**
   - **Existing Base System**
   - **New Base Systems**

**Time**

---

System Integration How-To
124  Gerrit Muller

version: 0.2  
January 28, 2015  
CVIntegrationPlan
Alternatives to Integrate a Subsystem Early in the Project

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

Physical environment spectrum:
- Physical reality
- Simulated physical
- Complex virtual
- Simple virtual

Virtual environment:
- Simulated subsystems
- Stubs

To-be-integrated subsystem

(modified) existing subsystems

(prototype) new subsystems

System Integration How-To
125 Gerrit Muller
### Stepwise Integration Approach

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

1. Correlate stage source
2. Measure x,y source
3. Measure x,y destination
4. Calibrate x,y measurement
5. Position x,y source
6. Position x,y destination
7. Measure alignment signal
8. Adjust light source
9. Adjust lens
10. Align source destination
11. Measure
12. Process
13. Expose
14. Focus
15. Measure x,y
16. Position x,y
17. Destination
18. Qualify
Roles and Responsibilities During the Integration Process

- **project leader**
  - organization
  - resources
  - schedule
  - budget

- **systems architect/engineer/integrator**
  - system requirements
  - design inputs
  - test specification
  - schedule rationale
  - troubleshooting
  - participate in test

- **system tester**
  - test
  - troubleshooting
  - report

- **logistics and administrative support**
  - configuration
  - orders
  - administration

- **engineers**
  - design
  - component test
  - troubleshooting
  - participate in test

- **machine owner**
  - maintain test model
  - support test
Configuration Management Entities

- **supplier**
  - Customer Oriented Process
  - Test models
  - Test models

- **company**
  - Content of pipeline
  - Specification
  - Test models
  - TPD

- **customer**
  - Product
  - Life cycle
  - Orders

Legend:
- Data
- Physical entity

System Integration How-To
130 Gerrit Muller

version: 0.2
January 28, 2015
SINTconfigurationManagement
Typical Order of Integration Problems

1. The (sub)system does not build.
2. The (sub)system does not function.
3. Interface errors.
4. The (sub)system is too slow.
5. Problems with the main performance parameter, such as image quality.
6. The (sub)system is not reliable.
Make a design for the documentation structure of the case, take into account a.o.:
- target audience per documentation module
- lifecycle
- author
- size (budget)

Present (max 1 flip) the proposed documentation structure and the rationale.
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

Recursive Decomposition

Summary Module Supporting Processes
Gerrit Muller
version: 0.2
January 28, 2015
Maximize Payload

A3s

<table>
<thead>
<tr>
<th>title</th>
<th>history</th>
<th>diagrams</th>
<th>1. aap</th>
<th>2. noot</th>
<th>3. mies</th>
<th>lists</th>
<th>and ca 50% text</th>
</tr>
</thead>
</table>

Light Weight Review

draft
concept
final review
authorization

consultation & review

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

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

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

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

Light Weight Review

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

consultation

change request

intentionally left blank
Systems Integration

Integration Starts at Feasibility

Alternatives for Early Integration

Propagation of Configuration Issues
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”

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

- What are the subsystems we will realize?

How

- How will the subsystems be realized?

up to "atomic" components
System as a Black Box

System seen as black box

inputs
functions
quantified characteristics
restrictions, prerequisites
boundaries, exceptions
standards, regulations

interfaces

outputs
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 How

Product What

Product How

drives, justifies, needs

enables, supports

Customer objectives

Application

Functional

Conceptual

Realization
Integrating CAFCR

What does Customer need in Product and Why?

Customer What
- Customer objectives

Customer How
- Application

Product What
- Functional
- Conceptual

Product How
- Realization

Context understanding
- Intention
- Objective driven

Opportunities
- Constraint awareness
- Knowledge based

Short introduction to basic “CAFCR” model
146  Gerrit Muller
CAFCR can be applied recursively

Short introduction to basic “CAFCR” model
Gerrit Muller

version: 0.4
January 28, 2015
CAFCRrecursion
<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
- department head
- user
- maintainer
- operator

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

Short introduction to basic “CAFCR” model

version: 0.4
January 28, 2015
BCAFCRwhosTheCustomer
CAFCR+ model; Life Cycle View

Customer objectives

Application

Functional

Conceptual

Realization

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
150  Gerrit Muller

version: 0.4
January 28, 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
- Enforce speed compliance
- Enforce red light compliance
- Enforce weight compliance

Requirements

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

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

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

Customer

How

Application

Derived Application Drivers

Product

What

Functional

What

How

Functional goal

means

may be skipped or articulated by several intermediate steps

functions

interfaces

performance figures

"REQfromDriverToRequirement"
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

Needs

Feedback

Continued Product Creation Process
Requirement Selection Process

- customer needs
- operational needs
- strategy
- roadmap
- competition
- product specification
- need characterization
- requirement phasing
- Technology, People, Process
- costs and constraints
Simple Qualification Method

- Discuss
- Do
- Don't discuss
- Don't discuss

- Important
- Urgent
- Effort
- Value
Examples of Quantifiable Aspects

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

Use relative scale, e.g. 1..5 1=low value, 5 -high value
Ask several knowledgeable people to score
Discussion provides insight  (don't fall in spreadsheet trap)
Exercise Requirements Capturing

- Determine the key drivers for one particular product family.
- Translate these drivers into application drivers and derive from them the requirements.
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?
- **How**
  - What are the subsystems we will realize?
  - How will the subsystems be realized?

Requirements for Requirements

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

Enable Human Use

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

**CAFCR+ Model**

- **C**ustomer objectives
- **A**pplication
- **F**unctional
- **C**onceptual
- **R**ealization

**Life cycle**
- operations
- maintenance
- upgrades

**development**
- manufacturing
- installation

**Sales, service, logistics, production, R&D**

**Example Key Driver Graph**

<table>
<thead>
<tr>
<th>Key drviers</th>
<th>Derived application drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety</td>
<td>Reduce accident rates</td>
</tr>
<tr>
<td></td>
<td>Enforce law</td>
</tr>
<tr>
<td></td>
<td>Improve emergency response</td>
</tr>
<tr>
<td>Effective Flow</td>
<td>Reduce delay due to accident</td>
</tr>
<tr>
<td>Smooth Operation</td>
<td>Optimize road surface</td>
</tr>
<tr>
<td></td>
<td>Speed up target groups</td>
</tr>
<tr>
<td>Environment</td>
<td>Anticipate on future traffic condition</td>
</tr>
</tbody>
</table>

**Requirements**
- Early hazard detection with warning and signaling
- Maintain safe road condition
- Detect and warn noncompliant vehicles
- Enforce speed compliance
- Enforce red light compliance
- Enforce weight compliance

**Note:** The graph is only partially elaborated for application drivers and requirements.

**Complementary Viewpoints**

- **top-down**
  - key-drivers (customer, business)
  - operational drivers (logistics, production, etc.)
  - roadmap (positioning and trends in time)
  - competition (positioning in the market)
  - regulations
  - “ideal” reference design
  - prototyping, simulation (learning vehicle)

- **bottom-up**
  - existing systems
  - bottom-up

**Exercise Requirements Capturing**

Gerrit Muller

version: 0
January 28, 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

- **Customer objectives**
- **Application**
- **Functional**
- **Conceptual**
- **Realization**

### Market

- **Products**

### Technology

### People

### Process

- **Marketing**
- **Architect**
- **Technology, process manager**

Time, ca 5 years
Granularity of Roadmap Material

Top-level roadmap
- Single page
- Poster
- part of many presentations

Supporting roadmaps
- Single page
- 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

Feature still unknown

Do!

Stop

Do!
Management with a Broader Time Perspective

Legend:
- Number of people allocated
- Time

2012

- Now
- Feature

2013

- Now
- Feature

2104

- Preparation by 0.5 person
- Work with 1.5 persons

- Continue with 0.5 person
- Work with 1.5 persons

Roadmapping
170 Gerrit Muller

version: 2.0
January 28, 2015
ROADanalogManagement
Creation or Update of Roadmap in Burst Mode

Collective meeting ca 2 days

2 weeks to digest and prepare

Preparation by expert teams

Roadmapping

Roadmapping

Gerrit Muller

version: 2.0
January 28, 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
budget
Q1 delta

201Y
Q1 Q2 Q3 Q4
roadmap n + 1

Policy and Planning Process

business plan: budget & allocation

detailed planning

market events

Product Creation Process

budget
Q2 delta
Q3 delta

budget
Q1 delta

Tech hurdle
## 3-Tier Approach

<table>
<thead>
<tr>
<th></th>
<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>
<td>vision</td>
</tr>
<tr>
<td>budget</td>
<td>1 year</td>
<td>3 months</td>
<td>program</td>
<td>commitment</td>
</tr>
<tr>
<td>detailed plan</td>
<td>1 mnth-1yr</td>
<td>1 day-1 mth</td>
<td>program or activity</td>
<td>control means</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

- Market analysis reports
  - number of customers, market size, competition, trends
- Installed base
  - change requests, problem reports, historical data
- Manufacturing (statistical process control)
  - statistical process control
- Suppliers (roadmaps, historical data)
  - roadmaps, historical data
- Internal reports (technology studies, simulations)
  - technology studies, simulations
Causes for Overestimation

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

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

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

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

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

Too many bells and whistles without business or customer value
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
January 28, 2015
MPLifecycleGraphideal
Market Product Life Cycle Phases in Practice

- Infancy
- Adolescence
- Maturity
- Aging

Sales volume vs. time graph showing ideal "bathtub" curve and observed curve deviation, with products unable to make transition to Maturity phase.
Examples of Product Classes on the Curve

- MRI scanner
- X-ray systems
- VCR
- DVD
- DVD+RW
- DVD
- digital TV
- flat TV
- TV

Market Product Life Cycle Consequences for Architecting

version: 1.2
January 28, 2015
MPLifecycleGraphExamples
<table>
<thead>
<tr>
<th>Attributes per Phase</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>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>
Summary of strategy process

Mission

Vision

Business specific, but open and generic

Empowerment

Input focus

Process

People

Market

Products

Technology

Forecasted facts

Educated scenarios

Roadmap

Input for next roadmap

Input for committal plan

Reality facts

Context overview

Exercise Roadmapping

version: 1.2
January 28, 2015
RSP summary
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

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

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

<table>
<thead>
<tr>
<th>Market</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collective meeting ca 2 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collective meeting ca 2 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collective meeting ca 2 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>People</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collective meeting ca 2 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collective meeting ca 2 days</td>
</tr>
</tbody>
</table>

Order of Creation

- **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: 2 weeks to digest and prepare
**Time Horizons**

```
201X  201Y
Q2    Q3    Q4    Q1    Q2    Q3    Q4    Q1

roadmap n

roadmapping

budget

Q1 delta

Q2
delta

Q3
delta

Q4
delta

Policy

and Planning

Process

business plan:

budget & allocation

detailed planning

market events

tech hurdle

Product Creation

Process
```

**Life Cycle Transitions**

```
Infancy  Adolescence  Maturity  Aging

sales volume

ideal "bathtub" curve

observed curve

product unable to make transition

time

Infancy  Adolescence  Maturity  Aging

intentionally left blank
```

**People and Process**

```
Customer objectives

Application

Market

Functional

Products

Conceived

Technology

Realization

feedback

after iteration

People

Process

estimation  

system 2002 2003 2004 2005 2006 2007

software 2002 2003 2004 2005 2006 2007

Orion  1  3  6  1  2

Gemini  2  4  8  2  4

Iridium  5  10  20  5  10

research  4  8  16  4  8

maintenance  2  4  8  2  4

total  16  32  64  16  32

estimate by program manager

estimate by people manager

version: 1.2
January 28, 2015
```
Abstract

The module Human Resource Management addresses the HRM aspects of systems architects, such as the profile of an architect, selection, education, appraisal and motivation.
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.
<table>
<thead>
<tr>
<th>Communication</th>
<th>Teamwork</th>
<th>Multitasking</th>
<th>Flexible, Open</th>
<th>Authority by Expertise</th>
<th>Specialist</th>
<th>Generalist</th>
<th>Conceptual</th>
<th>Pragmatic</th>
<th>Constructive Critical</th>
<th>Creative</th>
<th>Fast Absorption of Knowledge</th>
<th>Process Insight</th>
<th>Political Insight</th>
<th>Improvement</th>
<th>Completeness</th>
<th>Schedule</th>
<th>Monitor Progress</th>
<th>Decision Making</th>
<th>Initial Cost</th>
<th>Commercial Insight</th>
<th>Coaching</th>
<th>Selection</th>
<th>Appraisal</th>
<th>Motivation</th>
</tr>
</thead>
</table>

**System Architect**

Function Profiles; The Sheep with Seven Legs

194    Gerrit Muller

version: 1.0
January 28, 2015
FPSystemArchitect
Operational Leader

Function Profiles; The Sheep with Seven Legs
version: 1.0
January 28, 2015
Gerrit Muller
## The numbers behind the bars

<table>
<thead>
<tr>
<th>Function Profiles; The Sheep with Seven Legs</th>
<th>version: 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Gerrit Muller</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>systems architect</th>
<th>test engineer</th>
<th>developer</th>
<th>operational leader</th>
<th>line manager</th>
<th>commercial manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>
How to appraise or assess an architect?

by Gerrit Muller  Buskerud University College

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

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

Distribution

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

January 28, 2015
status: planned
version: 0.1
Problem statement

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

abstract (vague) responsibilities
lot of overlap of responsibilities

How to assess an architect?
Tangible deliverables based upon many invisible activities

Deliverables
- paperwork only

Responsibilities
- abstract and qualitative

Activities
- necessary but invisible

many very detailed

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

Stakeholders
- expectations, needs, concerns, constraints

Architecting
- team is enabled

PCP team
- architect, project leader, engineers, product manager

preceeding architecture

result satisfies

problem know how

solution know how

legenda
- business context
- technology context
- human context
Yardsticks for architect assessment

formalized expectations

*function* appraisal system,
f.i. from Hay Management Consultants
impact scope of control freedom of thinking

job description

deliverables timing

career development plan

skills know how

actual architect performance

*architecture* fitness

sales turnover business success market continuity

internal *stakeholder* satisfaction

contribution deliverables timing skills know how

How to appraise or assess an architect?

205 Gerrit Muller
How to appraise or assess an architect?

1. Ask for ranking
2. Ask for justification (why ...?)
3. Clarify criterions
4. Iterate ranking and justification

Jim Green (family architect)
Yo Nerd (SW engineer)
D. Blackhat (product architect)
Ju Nior (product architect)
John Brown (product architect)
Joe Go (project leader)
Se Nior (chief designer)
The Boss (business manager)

very high

value for the company

potential

low