Role of Systems Architecting in Innovation

by Gerrit Muller  
University of South-Eastern Norway-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

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.

September 6, 2020
status: preliminary draft
version: 0

logo TBD
The Embedded Systems Domain

- Chip
- Television
- Printer
- Waferstepper
- Cardio X-ray system
- MRI scanner
- GSM
Successful Innovation = Technological + Market

market innovations
segments
needs
applications
services

system

technological innovations
materials
circuits
functions
user
interface
System Architect links technology and market

Role of Systems Architecting in Innovation

- **Market innovations**
  - segments
  - applications
  - needs
  - services

- **Technological innovations**
  - materials
  - circuits
  - functions
  - user interface

- System

- Systems architect
- Engineers
- Inventors
- Marketeers
Example: Easyvision serving three URF examination rooms

URF-systems  EasyVision: Medical Imaging Workstation

typical clinical image (intestines)
Product Innovation: Easyvision applied as printserver

X ray source

detector

Examination Room

Control Room

Corridor or closet

printer

light box

Reading Room

Role of Systems Architecting in Innovation

Gerrit Muller
Market innovation: optimized film

old: screen copy

new: SW formatting

20 to 50% less film needed
Technology innovation challenges

- **Print throughput**
- **Image quality**
- **View response time**
- **Product policy:** standard HW SW "only"

- ca 1 film / minute
  - film = 4k*5k pixels
- Subsecond retrieve
  - screen = 1k*1k
- 40 MHz CPU
- 64 MByte memory
- 10 MBit/s ethernet
- 1 GByte disk

- Image processing

- Tension
Typical Growth of a System Architect

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

- **Specialist**
  - Depth of knowledge

- **Generalist**
  - Breadth of knowledge
  - Root knowledge
Generalists and Specialists are Complementary

Role of Systems Architecting in Innovation
Spectrum from Specialist to System Architect

Role of Systems Architecting in Innovation

13    Gerrit Muller

version: 0
September 6, 2020
MATfromSpecialistToSystemArchitect
More innovations in Medical Imaging

1992
- RF 1.1
  - URF basis
  - autoprint

1993
- RF 1.2
  - vascular
  - import

1994
- RF 2.1
  - cardio
  - bolus chase

1995
- RF 2.2
  - Dicom

1996
- X 3.1
  - spine

- Rad 1.1
  - PCR
  - Print
  - View, Print
  - Store, Communicate

- Rad 2.1
  - basis

- CT/MR 1.1
  - stack
  - MPR
  - dental

- CT/MR 1.2
  - MR
  - import

- CT/MR 2.1
  - volume
  - angio
Key success factor 1: innovation by all parties

- **market innovations**
  - segments
  - applications
  - needs
  - services

- **technological innovations**
  - materials
  - circuits
  - functions
  - user interface

- **system**

- **marketeers**
  - ✓

- **system architect**
  - ✓

- **engineers**
  - ✓

- **inventors**
  - ✓
The "CAFCR" model

What does Customer need in Product and Why?

Customer

What

Customer

objectives

Application

Functional

Conceptual

Realization

Product

How

drives, justifies, needs

enables, supports
Integrating CAFCR

What does Customer need in Product and Why?

Customer What
Customer How
Product What

C (Customer objectives)
Application
A
Function
F
Conceptual
C
Realization
R

context understanding
intention
objective driven

opportunities
constraint awareness
knowledge based

Version: 0
September 6, 2020
MSintegratingCAFCR
CAFCR can be applied recursively

Role of Systems Architecting in Innovation
18  Gerrit Muller

System (producer)

Value Chain
larger scope has smaller
influence on architecture

Customer Business

Consumer

Customer's Customer Business

Enables

Drives

Consumer Drives
Enables

Customer Business

Drives

Enables

larger scope has smaller influence on architecture
### CAFCR applied on Security

<table>
<thead>
<tr>
<th><strong>C</strong>ustomer objectives</th>
<th><strong>A</strong>pplication</th>
<th><strong>F</strong>unctional</th>
<th><strong>C</strong>onceptual</th>
<th><strong>R</strong>ealization</th>
</tr>
</thead>
<tbody>
<tr>
<td>sensitive information</td>
<td>selection</td>
<td>functions for</td>
<td>cryptography</td>
<td>specific</td>
</tr>
<tr>
<td>trusted</td>
<td>classification</td>
<td>administration</td>
<td>firewall</td>
<td>algorithms</td>
</tr>
<tr>
<td></td>
<td>people</td>
<td>authentication</td>
<td>security zones</td>
<td>interfaces</td>
</tr>
<tr>
<td></td>
<td>information</td>
<td>badges</td>
<td>authentication</td>
<td>libraries</td>
</tr>
<tr>
<td></td>
<td>authentication</td>
<td>passwords</td>
<td>registry</td>
<td>servers</td>
</tr>
<tr>
<td></td>
<td>guards</td>
<td>locks / walls</td>
<td>logging</td>
<td>storage</td>
</tr>
<tr>
<td></td>
<td>administrators</td>
<td>guards</td>
<td></td>
<td>protocols</td>
</tr>
</tbody>
</table>

**desired characteristics, specifications & mechanisms**

<table>
<thead>
<tr>
<th>not trusted</th>
<th>social contacts</th>
<th>missing functionality</th>
<th>holes between concepts</th>
<th>bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>open passwords</td>
<td>functionalty</td>
<td></td>
<td>buffer overflow</td>
</tr>
<tr>
<td></td>
<td>blackmail</td>
<td>wrong quantification</td>
<td></td>
<td>non encrypted</td>
</tr>
<tr>
<td></td>
<td>burglary</td>
<td></td>
<td></td>
<td>storage</td>
</tr>
<tr>
<td></td>
<td>fraud</td>
<td></td>
<td></td>
<td>poor exception</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>handling</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>unworkable procedures</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

**threats**

- versions
- bugs
- buffer overflow
- non encrypted storage
- poor exception
- handling
Deliverables of the System Architect
Responsibilities of the System Architect

- Balance
- Consistency
- Decomposition
- Integration
- Overview
- Requirement
- Spec
- Design
- Realization
- module
- subsystem
- system
- Quality
- Function
- modules
- KISS
- Elegance
- Simple
- Integrity
- Fitting
- satisfied stakeholders
- system
- context
- Role of Systems Architecting in Innovation

version: 0
September 6, 2020
RSA 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
From Detail to Overview

<table>
<thead>
<tr>
<th>consolidation in deliverables</th>
<th>driving views</th>
<th>Quantity per year (order-of-magnitude)</th>
<th>architect time per item</th>
</tr>
</thead>
<tbody>
<tr>
<td>meetings</td>
<td>shared issues</td>
<td>10</td>
<td>100 h</td>
</tr>
<tr>
<td>informal contacts</td>
<td>touched details</td>
<td>10²</td>
<td>1 h</td>
</tr>
<tr>
<td>sampling scanning</td>
<td>seen details</td>
<td>10⁴</td>
<td>0.5 – 10 min</td>
</tr>
<tr>
<td></td>
<td>product details</td>
<td>10⁵ – 10⁶</td>
<td>0.1 – 1 sec</td>
</tr>
<tr>
<td></td>
<td>real-world facts</td>
<td>infinite</td>
<td></td>
</tr>
</tbody>
</table>
Key Success Factor 2: highly iterative

![Diagram showing relationships between customer objectives, application, functional, conceptual, and realization aspects]

cost revisited in context of clinical needs and realization constraints; note: original threads are significantly simplified
Key Success Factor 3: Architect as Integrator

meddling architect

team full of heroes
Innovation Challenges in Embedded Systems

discover latent needs
enable emergence
where is the business
globalization
hype waves
Moore's law

creativity
market dynamics

security
interoperability

power consumption
reliability

privacy, DRM
versus usability
emerging behavior, future vs legacy
heterogeneous vendors
complexity
heterogeneity

weight, cost, performance

#engineers involved