Abstract

Today’s fast pace of the market and the technology development forces the product creators to rethink their development approach. One of the directions is to maximize the return on investments of frequently used functions, for instance by re-use, component based design or by a platform approach. The architecting effort is a key success factor to combine re-use approaches with fast and innovative product creation.

In this presentation we will present a case, discuss the role of the architecture, and elaborate the essential architecture ingredients for a successful platform creation, and evolution, and innovative product creation.
Q: How to manage platform architectures?

Recommendations

- case
- market driven
- architected
- process
- time dimension
- platform
case:
- company overview
- time line
- technology innovations
context, product(s) and design

Q: How to manage platform architectures?

Recommendations

case

architecting

platform

market driven

process

time dimension

How to Create a Manageable Platform Architecture?

version: 1.0
September 6, 2020
HMPAcaseOutline
Phases of Medical Imaging


Advanced Development
("Common Viewing")

Development of 1st product
Easyvision RF
Basic Application plus toolboxes

Parallel Development of 2nd product
Easyvision CT/MR
Easyvision Xray R1

Family Development
Easyvision RF R2
Easyvision CT/MR
Easyvision Xray R2
Easyvision RAD
EasyReview

Transformation in re-useable components
Medical Imaging Platform

How to Create a Manageable Platform Architecture?
Gerrit Muller
Technology innovations by Common Viewing

- standard UNIX based workstation
- full SW implementation, more flexible
- object oriented design and implementation (Objective-C)
- graphical User Interface, with windows, mouse et cetera
- call back scheduling, fine-grained notification
- data base engine, fast, reliable and robust
- extensive set of toolboxes
- property based configuration
- multiple co-ordinate spaces
Idealized layers September 1991

How to Create a Manageable Platform Architecture?

Gerrit Muller

version: 1.0
September 6, 2020
HMPAswLayers1991
X-ray rooms from examination to reading around 1990

Examination Room | Control Room | Corridor or closet

Examination Room | Control Room | Reading Room
X-ray rooms with Medical Imaging applied as printserver

Examination Room

Control Room

Corridor or closet

X ray source

detector

Examination Room

Control Room

Reading Room

printer

light box

How to Create a Manageable Platform Architecture?
9 Gerrit Muller

version: 1.0
September 6, 2020
XRayRoomsPlusPrintServer
Comparison *screen copy vs optimized film*

old: screen copy

new: SW formatting

20 to 50% less film needed
How to Create a Manageable Platform Architecture?

1. Gerrit Muller

version: 1.0
September 6, 2020
HMPAswLayers1992
Example Multi Planar Reconstruction

oblique slices
curved slice
Idealized layers june 1994

How to Create a Manageable Platform Architecture?

version: 1.0
September 6, 2020
MedicalImagingLayers1994
Example CT/MR department

How to Create a Manageable Platform Architecture?
14 Gerrit Muller
version: 1.0
September 6, 2020
MedicalImagingCTMRdepartment
### How to Create a Manageable Platform Architecture?

17 Gerrit Muller

---

**Back-ends**
- Image Guided Surgery
- Review
- Rad
- CT/MR
- XRay

**Specialized applications** (Dental, bolus chase, cardio analysis, etcetera)
- MR
- CT
- RF
- Vascular
- Cardio
- PCR

**Interfacing RIS, etcetera**
- Compose
- Print
- Store
- MPR
- View
- Export
- Cluster

**Spool**
- HCU
- Store
- Image
- Gfx
- UI
- DB

**PMS-net in**
- PMS-net out

**CDSpack**
- RC driver
- HC driver
- DOR driver
- NIX

**Solaris**

**Config**
- Service
- Customization
- SW keys

**Install**
- Start up
- Remote access
- Dev. tools

**New HCU**
- Desk, cabinets, cables, etc.
- Standard Sparcstation 5 workstation

**Back-ends**
- MR
- CT
- DSI
- DCAS
- PCR
Architecting Outline

Architecting:
- What is Architecture?
- Architecture vs Description
- My View on Architecture
- ”Guiding How”
- The Art of Architecting
- More than Decomposition

Q: How to manage platform architectures?

Recommendations

case
market driven
process
time
dimension

platform
architecting

How to Create a Manageable Platform Architecture?
18 Gerrit Muller

version: 1.0
September 6, 2020
HMPAarchitectingOutline
What is Architecture?

Mark all applicable boxes

- specifications
- API's
- components (implementations)
- infrastructure
- indicators
- high level rules
- concepts
- standards
- overarching vision
- guidance monitoring
- domain codification
- other...

How to Create a Manageable Platform Architecture?

version: 1.0
September 6, 2020
HMPAwhatIsArchitecture
Architecture

Subset of which architect is aware

Flattened into

Architecture description

Actually written by architect(s)
My View on Architecture

Understanding Why
Describing What
Guiding How

Do the right things
Do the things right
"Guiding How” by providing rules for:

1. **Functional Decomposition**

2. **Construction Decomposition**

3. **Allocation**

4. **Infrastructure**

5. **Choice of integrating concepts**
The Art of Architecting

Stakeholders

Expectations

Architecting

Facts

Architecture(s) → problems, legacy

Architecture uncertainties unknowns

Facts, Expectations and Intuition may be false
Integration requires a critical mindset that is alert for unknowns

Architect(s)

Intuition, assumptions, beliefs, bias

analyze
assess
balance
trade-off
decide

vision
overview
insight
understanding

How to Create a Manageable Platform Architecture?

version: 1.0
September 6, 2020
LWAarchitecting
Architecting is much more than Decomposition

Decomposition is "easy"

Integration is difficult
Platform Outline

Platform:
- Why Platforms?
- What is a Platform?
- Platform Source Deliverables
- Example of Platform Efficiency
- Embedding Costs of Purchased SW

Q: How to manage platform architectures?

Recommendations

Case

Market driven

Architecting

Process

Time dimension

How to Create a Manageable Platform Architecture?

25  Gerrit Muller

version: 1.0
September 6, 2020
HMPAplatformOutline
Why Platforms?

Customer value
- application adaptability
- availability variations
- new features originating from different products
- timely availability
- reliability
- asset creation
  - increase economy of scale

Internal benefits
- availability of accumulated feature set
- design for configurability
- shared architectural framework
- quality increase
- predictability
- availability integrated base product
- maturity

Extrovert driver
Introvert driver

How to Create a Manageable Platform Architecture?

version: 1.0
September 6, 2020
Gerrit Muller
What is a Platform?

huge product integration effort
very flexible
low coupling
configuration management???

no product integration effort
not flexible
high coupling
configuration management

Legend:
- **Product**: Implementation concepts
- **Platform**: Applications + integration glue + infrastructure

How to Create a Manageable Platform Architecture?

27 Gerrit Muller
How to Create a Manageable Platform Architecture?

Gerrit Muller

version: 1.0
September 6, 2020
HMPAplatformDeliverablesSimplified
And now in More Detail...

<table>
<thead>
<tr>
<th>development process</th>
<th>code</th>
<th>specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>test code &amp; data</td>
<td>requirements</td>
</tr>
<tr>
<td></td>
<td>source code</td>
<td>interfaces</td>
</tr>
<tr>
<td></td>
<td>target OS</td>
<td>design</td>
</tr>
<tr>
<td></td>
<td>purchased SW</td>
<td>reports</td>
</tr>
<tr>
<td></td>
<td>generation recipes</td>
<td>manuals</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>configuration management</th>
<th>development environment</th>
<th>documentation tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>compiler, linker, ...</td>
<td>word processing</td>
</tr>
<tr>
<td>problem reports</td>
<td>dev. cluster OS</td>
<td>drawing</td>
</tr>
<tr>
<td>change requests</td>
<td>meta data (review, metrics)</td>
<td>spreadsheets</td>
</tr>
<tr>
<td>documentation</td>
<td>customization</td>
<td>publishing</td>
</tr>
<tr>
<td>documentation</td>
<td>dev process support</td>
<td>management</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>source code</td>
</tr>
<tr>
<td>target OS</td>
</tr>
<tr>
<td>purchased SW</td>
</tr>
<tr>
<td>generation recipes</td>
</tr>
<tr>
<td>compiler, linker, ...</td>
</tr>
<tr>
<td>dev. cluster OS</td>
</tr>
<tr>
<td>meta data (review, metrics)</td>
</tr>
<tr>
<td>customization</td>
</tr>
<tr>
<td>dev process support</td>
</tr>
<tr>
<td>word processing</td>
</tr>
<tr>
<td>drawing</td>
</tr>
<tr>
<td>spreadsheets</td>
</tr>
<tr>
<td>publishing</td>
</tr>
<tr>
<td>management</td>
</tr>
</tbody>
</table>

---

How to Create a Manageable Platform Architecture?

29    Gerrit Muller

version: 1.0
September 6, 2020
HMPAplatformDeliverables
## Example of Platform Efficiency

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>applications</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>number of inputs (a.o. modalities)</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>applications</td>
<td></td>
<td>27</td>
<td>35</td>
<td>37</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>total</td>
<td></td>
<td>52</td>
<td>62</td>
<td>72</td>
<td>79</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Purchased SW Requires Embedding

How to Create a Manageable Platform Architecture?

31 Gerrit Muller

version: 1.0
September 6, 2020
HMPAembedding
Embedding Costs of Purchased SW

- Installation
- Configuration
- Customization
- Start up, shutdown
- Specifications
- Interface to application SW
- Exception handling
- Resource allocation and monitoring provision
- Resource tuning, see above
- Safety design
- Security design

- functional
- system design
- sw design
- add semantics level
- use of appropriate low level mechanisms
- match to high level mechanisms:
  - notification, scheduling
  - job requests, subscriptions

- System monitor
- Error propagation
- Logging
- CPU
- Memory
- Disk

How to Create a Manageable Platform Architecture?

version: 1.0
September 6, 2020
HMPAembeddingCost
Example of Embedding Problems

Architectural mismatch:
wrappers, translators, conflicting controls

additional code
and complexity,
no added value

Poor performance;
additional resource usage

Problems Architecture Reuse non problem

Version: 1.0
September 6, 2020

How to Create a Manageable Platform Architecture?
Time Dimension:
- Who is First: Platform or Product?
- Platform Stability
- The First Time Right?
- Evolution of Easyvision Platform
- Lifecycle Differences

Q: How to manage platform architectures?

Recommendations

- case
- architecting
- platform
- time dimension
- market driven
- process
Who is First: Platform or Product?

Platform as consolidation baseline

R1  R2  R3

Platform

How to Create a Manageable Platform Architecture?

version: 1.0
September 6, 2020
HMPreleaseModel

Gerrit Muller
Myth: Platforms are Stable

How stable is a platform or an architecture?

Dynamic Market

Architecture
Components
Platform

Fast changing Technology
The First Time Right?

First time right?  maybe  unlikely  miracle  impossible

person years  1  10  100  1000
stepsize: 3 months
elapsed time: 25 months
Feedback (2)

How to Create a Manageable Platform Architecture?

version: 1.0
September 6, 2020
LWAfeedbackMedium
Small feedback cycles result in Faster Time to Market

3rd generation components are mature, active maintenance needed.
Growth and change continues, some "old" components become obsolete

Last changed in:
1991
1992
1994
Lifecycle Differences

- problem response
- commodity hardware and software
- minor SW release
- new generation of magnets, gradients, detectors
- major SW release
- workstation useful life
- MR scanner useful life

- clinical prototype
- procedural change
- legislation change

- 3 months
- 1 year
- 10 years
Reference Model for Healthcare Automation

<table>
<thead>
<tr>
<th>Information Handling</th>
<th>Imaging and Treatment</th>
<th>Image Handling</th>
<th>Archiving</th>
</tr>
</thead>
<tbody>
<tr>
<td>entirely distributed</td>
<td>localised</td>
<td>distributed</td>
<td>service business</td>
</tr>
<tr>
<td>wide variation</td>
<td>patient focus</td>
<td>limited</td>
<td>not health care specific</td>
</tr>
<tr>
<td>due to &quot;socio-geographics&quot;:</td>
<td>safety critical</td>
<td>variation</td>
<td>extreme robust</td>
</tr>
<tr>
<td>psycho-social,</td>
<td>limited variation</td>
<td>due to &quot;nature&quot;:</td>
<td>fire, earthquake,</td>
</tr>
<tr>
<td>political, cultural factors</td>
<td>due to &quot;nature&quot;:</td>
<td>human anatomy</td>
<td>flood proof</td>
</tr>
<tr>
<td>human anatomy</td>
<td>human anatomy</td>
<td>pathologies</td>
<td>life time</td>
</tr>
<tr>
<td>pathologies</td>
<td>pathologies</td>
<td>imaging physics</td>
<td>100 yrs (human life)</td>
</tr>
<tr>
<td>imaging physics</td>
<td>imaging physics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>base technology</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not health care specific</td>
<td>short life-cycles</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rapid innovation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How to Create a Manageable Platform Architecture?

Gerrit Muller

version: 1.0
September 6, 2020
MiCAFreferenceModel
Q: How to manage platform architectures?

Process View:
- Simplified Process Decomposition
- Financial View on Process Decomposition
- Value and Feedback Flow
- Propagation Delay
- Sources of Failure
- Models for Generic Development

Recommendations

Case

Architecting

Platform

Time Dimension

Process

Market Driven

How to Create a Manageable Platform Architecture?

Gerrit Muller

version: 1.0
September 6, 2020
HMPAprocessOutline
Simplified Process Decomposition

- **customer**
- **supplying business**
  - **strategy process**
  - **customer oriented (sales, service, production) process**
  - **product creation process**
  - **people, process and technology management process**

---

How to Create a Manageable Platform Architecture?

version: 1.0
September 6, 2020
RSPprocessDecomposition
Modified Simplified Process Decomposition

Philips business

policy and planning

PCP

customer oriented process (sales, service, production)

create generic components

people and technology management process

customer

value
How to Create a Manageable Platform Architecture?

Philips business

customer

management
cashflow generation
value

create
tomorrow's cashflow

strategic asset

generation

people & assets

technology management process

Policy and Planning
Customer
Philips Business Value
PCP
Customer Oriented Process (sales, service, production)

People and Technology Management Process
Create Generic Components

How to Create a Manageable Platform Architecture?  47  Gerrit Muller

version: 1.0
September 6, 2020
SWRprocessDecompositionFamilyByValue
Value and Feedback Flow

Philips business

policy and planning

customer oriented process
(sales, service, production)

PCP

create generic components

people and technology management process

customer

feedback

value
## Sources of Failure in Platform Developments

<table>
<thead>
<tr>
<th>Technical</th>
<th>Process/People/Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Too generic</td>
<td>• Forced cooperation</td>
</tr>
<tr>
<td>• Innovation stops (stable</td>
<td>• Time platform feature to market</td>
</tr>
<tr>
<td>interfaces)</td>
<td>• Unrealistic expectations</td>
</tr>
<tr>
<td>• Vulnerability</td>
<td>• Distance platform developer to customer</td>
</tr>
<tr>
<td></td>
<td>• No marketing ownership</td>
</tr>
<tr>
<td></td>
<td>• Bureaucratic process (no flexibility)</td>
</tr>
<tr>
<td></td>
<td>• New employees, knowledge dilution</td>
</tr>
<tr>
<td></td>
<td>• Underestimation of platform support</td>
</tr>
<tr>
<td></td>
<td>• Overstretching of product scope</td>
</tr>
<tr>
<td></td>
<td>• Nonmanagement, organizational scope increase</td>
</tr>
<tr>
<td></td>
<td>• Underestimation of integration</td>
</tr>
<tr>
<td></td>
<td>• Component/platform determines business policy</td>
</tr>
<tr>
<td></td>
<td>• Subcritical investment</td>
</tr>
</tbody>
</table>
How to Create a Manageable Platform Architecture?

51 Gerrit Muller

version: 1.0
September 6, 2020
GDmodels
Market Driven Outline

Q: How to manage platform architectures?

Recommendations

Market Driven:
- The “CAFCR” model
- Example Platform Scoping
- Customer Key Drivers
The “CAFCR” model

What does Customer need in Product and Why?

Customer What

Customer How

Product What

Product How

C - Customer objectives

A - Application

F - Functional

C - Conceptual

R - Realization

drives, justifies, needs

enables, supports

How to Create a Manageable Platform Architecture?

version: 1.0

September 6, 2020

CAFCR annotated
Five viewpoints for an architecture

Customer objectives

Application

Functional

Conceptual

Realization

What does Customer need in Product and Why?

Customer What

App

F

C

R

Product How

context understanding

intention driven

objective

opportunity awareness

constraint based

know how based

How to Create a Manageable Platform Architecture?

version: 1.0

September 6, 2020

AMOintegratingCAFCR
Example Platform Scoping
Customer Key Drivers Motorway Management

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
Finally All Design Decisions are Related to Market

**Customer objectives**
- useable
- effective
- efficient

**Application**
- diagnostic
- quality
- time
- efficient

**Functional**
- IQ spec
- throughput
- purchase price

**Conceptual**
- render engine

**Realization**
- S
- P
- M

**Operational constraints**
- profit margin
- standard workstation
- Philips operational view
- (manufacturing, service, sales)

**Cost revisited in context of clinical needs and realization constraints;** note: original threads are significantly simplified.
Recommendations

Q: How to manage platform architectures?

- **to create** successful products
- **educate** artful architects
- **accept** heterogeneous solutions
- **implement** agile lifecycle decoupling
- **stimulate** evolution, and continuous refactoring

**Recommendations**

- **identify** key drivers
- **maintain** focused scope
- **ensure** market and business feedback
- **process**
- **architecting**
- **platform**
- **case**
- **market driven**
- **time dimension**

How to Create a Manageable Platform Architecture?

version: 1.0

September 6, 2020

HMPA recommendations