How to Create a Manageable Platform Architecture?

by Gerrit Muller   University of South-Eastern Norway-NISE
   e-mail: gaudisite@gmail.com
   www.gaudisite.nl

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.

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.
Q: How to manage platform architectures?

Recommendations

- market driven
- process
- time dimension

- platform
- architecting
- case
case:
- company overview
- time line
- technology innovations
- 1991, 1992, 1994, and 1996 context, product(s) and design

Q: How to manage platform architectures?

Recommendations

architecting

process

market driven

time dimension

platform

case
Philips Medical Systems, schematic organization

Philips Medical Systems

- **US**
- **Non X-ray modalities**
  - **MR**
  - **CT**
  - **Medical Imaging**

- **Conventional X-ray**
  - **Cardio Vascular**
  - **URF**
  - **Surgery**

- **Common X-ray Components**
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

Standard Sun workstation
SunOS, SunView
Basic Application

Legend
- user interface
- toolbox
- operating system
- hardware

How to Create a Manageable Platform Architecture?
Gerrit Muller
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

printer

Reading Room

light box
Comparison *screen copy* vs *optimized film*

**old: screen copy**

**new: SW formatting**

20 to 50% less film needed
Medical Imaging R/F

<table>
<thead>
<tr>
<th>Print</th>
<th>Store</th>
<th>View</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spool</td>
<td>HCU</td>
<td>Store Image Gfx UI DB</td>
<td>PMS-net in PMS-net out</td>
</tr>
<tr>
<td>RC driver</td>
<td>HC driver</td>
<td>DOR driver</td>
<td>NIX SunOS</td>
</tr>
</tbody>
</table>

Standard IPX workstation

- **User interface**
- **Application functions**
- **Toolbox**
- **Operating system**
- **Hardware**
- **SW infrastructure**
- **Connected system**

How to Create a Manageable Platform Architecture?

11 Gerrit Muller
Example Multi Planar Reconstruction

How to Create a Manageable Platform Architecture?
12 Gerrit Muller

version: 1.0
September 9, 2018
MedicalImagingMPRexample
How to Create a Manageable Platform Architecture?

How to Create a Manageable Platform Architecture?

13 Gerrit Muller

version: 1.0
September 9, 2018
MedicalImagingLayers1994
Example CT/MR department

MR Examination room

Control room

"MPR" room

CT Examination room

Control room

Reading Room
Vision: Medical Imaging in Healthcare

Operating theatre  trauma room

Radiology department

IT infrastructure in basement

How to Create a Manageable Platform Architecture?

Gerrit Muller
How to Create a Manageable Platform Architecture?

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

19 Gerrit Muller

version: 1.0
September 9, 2018
Architecture vs Description

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

How to Create a Manageable Platform Architecture?

version: 1.0
September 9, 2018
LWAArchitectureHow
The Art of Architecting

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

Facts, problems, legacy

Architecting

- analyze
- assess
- balance
- trade-off
- decide
- vision
- overview
- insight
- understanding

Architecture uncertainties unknowns

Stakeholders

Expectations

Architect(s)

Intuition, assumptions, beliefs, bias
Architecting is much more than Decomposition

Decomposition is "easy"

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

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

Internal benefits
- asset creation
- increase economy of scale
- design for configurability
- shared architectural framework
- predictability
- availability integrated base product
- reliability
- quality increase
- maturity

Extrovert driver

Introvert driver

How to Create a Manageable Platform Architecture?

Gerrit Muller

version: 1.0
September 9, 2018
GDdrivers
What is a Platform?

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

no product integration effort
not flexible
high coupling
configuration management

How to Create a Manageable Platform Architecture?

27 Gerrit Muller
How to Create a Manageable Platform Architecture?

Gerrit Muller

version: 1.0
September 9, 2018
HMPAplatformDeliverablesSimplified

Platform Source Deliverables

development process
code
specifications
configuration management
development environment
documentation tools
infrastructure
### Development Process
- Code specifications
- Development environment
- Documentation
- Tools

### Code
- Test code & data
- Source code
- Target OS
- Purchased SW
- Generation recipes

### Configuration Management
- Meta data (review, metrics)
- Customization
- Dev. process support

### Development Environment
- Compiler, linker, ...
- Dev. cluster OS
- Change requests documentation

### Infrastructure
- Dev. process support
- Word processing
- Drawing
- Spreadsheets
- Publishing
- Management
### 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>
</tr>
</thead>
<tbody>
<tr>
<td>platform</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>37</td>
</tr>
<tr>
<td>total</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>
</tr>
</thead>
<tbody>
<tr>
<td>people per application</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?**

version: 1.0
September 9, 2018
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
Example of Embedding Problems

Architectural mismatch:
wrappers, translators, conflicting controls

Additional code
and complexity,
no added value

Poor performance;
aditional resource usage

Problems
Architecture
Reuse
non problem

How to Create a Manageable Platform Architecture?

version: 1.0
September 9, 2018

ARmergeProblems
Time Dimension Outline

Q: How to manage platform architectures?

- Recommendations
- Case
- Architecting
- Platform
- Time dimension

- Who is First: Platform or Product?
- Platform Stability
- The First Time Right?
- Evolution of Easyvision Platform
- Lifecycle Differences
Who is First: Platform or Product?

How to Create a Manageable Platform Architecture?

version: 1.0
September 9, 2018
HMPAreleaseModel

platform baseline

platform as consolidation baseline
Myth: Platforms are Stable

Dynamic Market

How stable is a platform or an architecture?

Architecture

Components

Platform

Fast changing Technology
The First Time Right?

First time right?  maybe  unlikely  miracle  impossible

person years  1  10  100  1000
Feedback

stepsize: 3 months
elapsed time: 25 months
Feedback (2)

stepsize: elapsed time

3 months
25 months
2 months
12 months

Start Start Target

Target

Start

Start

Start
Small feedback cycles result in Faster Time to Market

- 1991: Growth
- 1992: Growth and change continues, some "old" components become obsolete
- 1994: "3rd generation components are mature, active maintenance needed"
- 1996: Growth and change continues, some "old" components become obsolete

Last changed in:
- 1991
- 1992
- 1994

How to Create a Manageable Platform Architecture?

version: 1.0
September 9, 2018
LWAplatformEvolution
Lifecycle Differences

- **Problem response**: 3 months
- **Clinical prototype**: 1 year
- **Procedural change**: 1 year
- **Legislation change**: 10 years
- **Workstation useful life**: 10 years
- **MR scanner useful life**: 10 years

- **Commodity hardware and software**:
  - Minor SW release
  - Major SW release

- **New generation of magnets gradients detectors**
Reference Model for Healthcare Automation

**Information Handling**
- Entirely distributed
  - Wide variation due to "socio-geographics":
    - Psycho-social,
    - Political, cultural factors

**Image Handling**
- Distributed
  - Limited variation due to "nature":
    - Human anatomy
    - Pathologies
    - Imaging physics

**Base Technology**
- Not health care specific
  - Short life-cycles
  - Rapid innovation

**Archiving**
- Service business
  - Not health care specific
  - Extreme robust
    - Fire, earthquake, flood proof
  - Life time
    - 100 yrs (human life)

**Imaging and Treatment**
- Localised
  - Patient focus
  - Safety critical
  - Limited variation due to "nature":
    - Human anatomy
    - Pathologies
    - Imaging physics

How to Create a Manageable Platform Architecture?
Gerrit Muller

version: 1.0
September 9, 2018
MICAFReferenceModel
Process View:
- Simplified Process Decomposition
- Financial View on Process Decomposition
- Value and Feedback Flow
- Propagation Delay
- Sources of Failure
- Models for Generic Development

Q: How to manage platform architectures?
Recommendations

- case
- market driven
- process
- platform
- time dimension

How to Create a Manageable Platform Architecture?
Gerrit Muller
Simplified Process Decomposition

How to Create a Manageable Platform Architecture?

Gerrit Muller

version: 1.0
September 9, 2018
RSPprocessDecomposition
Modified Simplified Process Decomposition

Philips business

policy and planning

customer oriented process
(sales, service, production)

PCP

create generic components

people and technology management process

customer
Financial View on Process Decomposition

Philips business

management
planning

customer

value

cashflow generation

create
strategic asset
generation

people
assets

create
technology management process

How to Create a Manageable Platform Architecture?

version: 1.0
September 9, 2018
SWRprocessDecompositionFamilyByValue
Philips business

policy and planning

PCP

create generic components

people and technology management process

customer oriented process
(sales, service, production)

customer

feedback

value

How to Create a Manageable Platform Architecture?
Gerrit Muller

version: 1.0
September 9, 2018
Propagation Delay Platform Feature to Market

How to Create a Manageable Platform Architecture?

Gerrit Muller
# 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</td>
<td>• Time platform feature to market</td>
</tr>
<tr>
<td>(stable 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?

50  Gerrit Muller

version: 1.0
September 9, 2018
GDpitfalls
Models for Platform 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)
- Create generic components

**Supplying Business**
- Policy and planning
- People and technology management process

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

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 9, 2018
HMPAmarketOutline
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
Five viewpoints for an architecture

- **Customer What**: What does Customer need in Product and Why?
- **Customer How**: Based
- **Product What**: What does Customer need in Product and Why?
- **Product How**: How
- **Application**: What
- **Functional**: How
- **Conceptual**: What
- **Realization**: How

**Context understanding**: Opportunities
**Intention**: Constraint awareness
**Objective driven**: Know how
**Realization**: Integrating CAFCR
Example Platform Scoping

intelligent buildings

motorway management

railway stations

airport terminals

shared core technology

Closed Circuit TV

audio broadcasting

access control

networking
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
diagnostic quality

Application
image quality

Functional
IQ spec

Conceptual
render engine

Realization
processing library
pixel depth
memory limit

cost revisited in context of clinical needs and realization constraints; note: original threads are significantly simplified

How to Create a Manageable Platform Architecture?

57 Gerrit Muller

version: 1.0
September 9, 2018
MITORthread50
Q: How to manage platform architectures?

Recommendations

- Identify key drivers
- Accept heterogeneous solutions
- Stimulate evolution, and continuous refactoring
- Implement agile lifecycle decoupling
- Maintain focused scope
- Ensure market and business feedback
- Educate artful architects
- To create successful products

How to Create a Manageable Platform Architecture?

version: 1.0
September 9, 2018
HMPA_recommendations