Grundlæggende Softwarearkitektur i Praksis

Emner

1. **Software architecture**, terminology, definitions, role in the development process, strengths and limitations.

2. **Architectural description**, Terminology, view-based descriptions, architecture description languages (ADLs)

3. **Quality attributes**, Terminology, specification, quality attribute scenarios, role of quality attributes.

4. **Architectural design**, tactics, styles, design process

5. **Product lines**, terminology, uses, principles

6. **Service oriented architecture**, reference architecture, uses, technology.

1. **Software arkitektur**, terminology, definitions, role in the development process, strengths and limitations.

**Disposition**

**1 Hvad er SA?**

**2 Definition**

**Structures**

**Properties**

**Elements**

**Relationships**

**3 Hvorfor SA?**

**4 ABC**

**5 Architectural drivers**

* 1. Hvad er IKKE SA? Box-and-line.
  2. Difference between SA and SA Description – This is not a pipe
  3. Every system has an architecture
  4. Not every system has a SAD
  5. *The software architecture of a computing system is*

*the structures† of the system, which comprise*

*software elements, the externally visible*

*properties of those elements, and the*

*relationships among them.*

Structures = views (som excel, html), separation of concerns ligesom bygningsarkitektur, eldiagrammer, vvs, floor-plans. En måde at tackle kompleksitet og at fordele opgaver. Diagrammer med præcis notation.

Properties = assumptions on elements, services, performance, security, fault handling

Externally visible = public

Relationships – Control flow, data flow, dependencies

Why SA?

* To cope with complexity – omit details – Man kan let drukne i detaljer, ikke se skoven for træer
* Balancere forskellige krav (QAS)
* Risk analysis and management
* Communication, common language and goals among stakeholders.
* Analysis -> Architecture -> Design -> implementation

Architecture Business Cycle (mynd)

Architectural drivers, determined by quality attributes

1. **Architectural description** Terminology, view-based descriptions, architecture description languages (ADLs)
2. **SAD**
   1. **SA vs SAD**
   2. **Informal vs formal**
   3. **Hvorfor SAD?**
3. **Viewpoints**

Difference between SA and SAD – This is not a pipe

Every system has an architecture

Not every system has a SAD

Description needed for communication between stakeholders

Different stakeholders – different representations. Customers not interested in architectural details.

Viewpoints (example, beer web store)

* **Module** (functionality -> implementation units)
  + Elements
    - Classes, interfaces, packages
  + Relations
    - Associations, generalizations, realization, dependencies
  + UML: Class diagrams
* **Component and connector** (functionality mapped onto runtime components + interaction)
  + Elements
    - Components
      * Functional behaviour
      * What part of the system is doing what
  + Relations
    - Connectors
      * Control and communication aspects
      * Protocols for control and data exchange
  + UML: Object diagram, interaction diagram
    - components -> active objects
    - connectors -> links and annotations, messages
    - + textual description
* Allocation (software elements – hardware units/environmental structures)
  + Elements
    - Software elements: components, objects
    - Environmental elements: Nodes
  + Relations
    - Allocated-to
    - Dependencies
    - Connections (communication paths)
  + UML: Focus on deployment. Components.

Is the description part of the architecture?

Broad range of descriptions

From informal (box-and-line) to formal (ACME, Darwin etc.). UML multiple view.

(Architectural drivers)

1. **Quality attribute scenarios**, Terminology, specification, Quality Attribute scenarios, role of quality attributes

**1 QAS vs. krav.**

**2 QAS**

**Availability**

**Modifiability**

**Performance**

**Security**

**Testability**

**Usability**

**3 Eksempler**

**4 Hvilken rolle?**

* 1. Målbare non-funktionelle krav.
  2. Alle aktører ønsker performance, sikkerhed, modificerbarhed (sammenlign med bil)
     1. specialister vil altid trække i “deres” retning.
  3. Define subjects/qualities to measure
  4. Define a scale or metric.

|  |  |
| --- | --- |
| Scenario | Customers add beer to their shopping basket ten times in one second under normal operation. |
| Relevant Quality Attributes | Performance |
| Source | Users |
| Stimulus | Add to basket |
| Artifact | System |
| Environment | Normal operation |
| Response | Items added to the basket |
| Response measure | At least 10 pr. second. |
| Questions |  |
| Issues |  |
| Scenario | The credit card validation service fails during normal operation. The failure is logged, and the user notified of the failure. The system continues to operate with no downtime. |
| Relevant Quality Attributes | Availability |
| Source | External |
| Stimulus | Payment fails unexpectedly |
| Artifact | System |
| Environment | Normal operation |
| Response | Abort the payment. Log the error. Continue in normal mode. |
| Response measure | No downtime. |
| Questions |  |
| Issues |  |

System quality attributes

* Availability
* Modifiability
* Performance
* Security
* Testability
* Usability

Business qualities

* Time to market
* Cost
* Projected lifetime
* ...

Architectural qualities

* Conceptual integrety
* Correctness and completeness
* Buildability

1. **Architectural design**, tactics, styles, design process
   1. *An architectural style is a description of component types and a pattern of their runtime control and/or data transfer*.
   2. Tactics – Ways of getting a quality, relate to QAS.
2. **Product lines**, terminology, uses, principles
   1. Core assets
      1. Requirements
      2. Design and code
      3. Architecture
      4. Documentation
      5. Project management
   2. Reuse
   3. Identificer forskelle
      1. brugergrænseflader, platforme, features, qualities
   4. Supporter forskelle
      1. Config files, defer binding time, moduler
   5. Identificer ligheder
   6. Adoption, top-down or bottom-up. Management needs to be in on the idea.
   7. Heavyweight or lightweight
   8. Evolution of product lines
      1. new versions
      2. new technology
      3. needs change
3. 6. **Service oriented architecture**, reference architecture, uses, technology.

“is a paradigm for organizing and utilizing distributed

capabilities that may be under the control of different

ownership domains”

“– provides a uniform means to offer, discover, interact

with and use capabilities to produce desired effects

consistent with measurable preconditions and

expectations”