Software Architecture in Practice

Welcome to SAiP

Henrik Bærbak Christensen

- Associate Professor
  - Computer science department / Aarhus University
  - Interests: Software architecture, software engineering, teaching
  - Leader of AU part-time education in SW engineering

- Industrial experience
  - Architect and developer for a product suite of meteorological systems for Danish airports.
  - Collaborations with Danish companies: Danfoss, SSE, KMD, B&O, Terma, Rambøll, Jyske Bank, Uber, and many others…
  - Imhotep: Courses and consulting… www.imhotep.dk

Imhotep
Vidensformidling inden for
Softwareudvikling

Credits...

- Several people have contributed to this course

  - Klaus Marius Hansen, DIKU/Microsoft
  - Aino Corry, Metadeveloper
  - Former students ☺

  - And my son, Magnus!

- The audience:
  - Master students: 1st of 3 courses
    - 1: Basic Software Architecture
    - 2: Adv. Software Architecture
    - 3: Project in Software Architecture

- Consequence:
  - All courses are a progression…
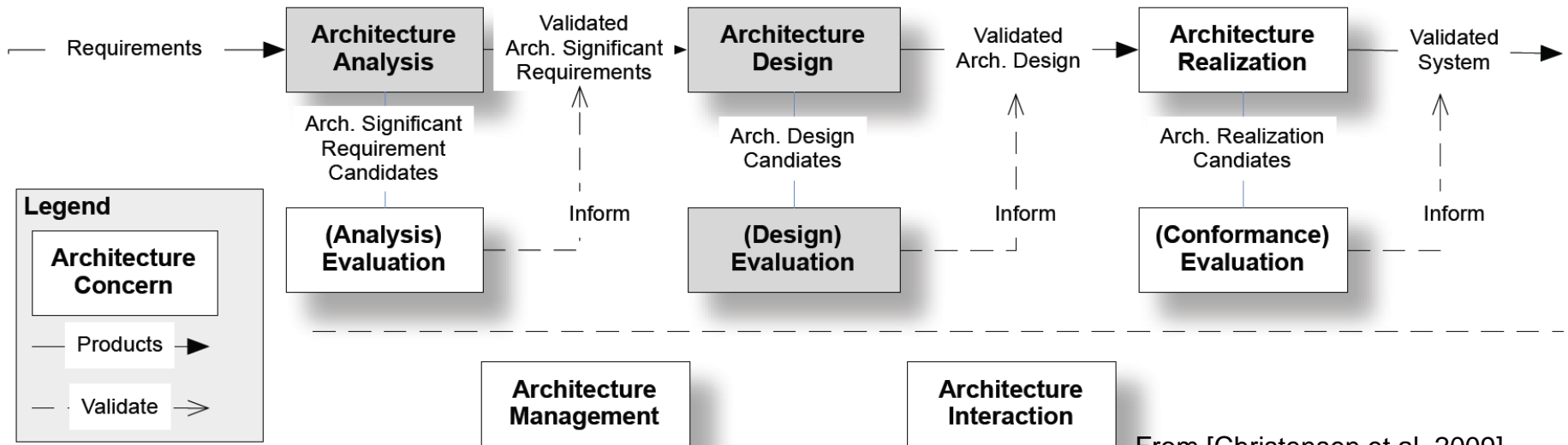
Henrik Bærbak Christensen

The courses

- Fundamental
  - Documentation, design, evaluation
  - More on *core constructive* tasks

- Advanced
  - Perspectives, performance, NoSQL, energy awareness
  - More on *perspectives* and *new approaches*

- Project
  - Have a go at your own (in groups ☺)

A framework for the topics

- Doing software architecture...

From [Christensen et al. 2009]

Goal of the present course

- Terminology
  - to support communication

- Techniques
  - to support practical work
    - Will be illustrated by *examples* of concrete tools
  - in the *mental tool* sense
    - patterns, conceptual frameworks, technology independent designs

- Perspective
  - Some topics only exemplified or outlined...

Ground Truth

- SAiP is
  - Academic   Terminology, process, methods
  - Constructive   Tools, application, validation

- Ground Truth
  - We need to tie *knowledge and skills* to *machinery*
  - Our tool stack is *open source*
    - *Java, Gradle, JaCoCo, IntelliJ, Docker, JMeter, MongoDB, …*

  - *You will need to get dirty fingers…*

My Conviction

- Software architecture is about designing systems so they fulfill our requirements to *quality attributes*
  - Like modifiability, availability, performance, testability, …
    - We will get back to these soon, but…
- **Many of these manifest themselves in the way we write our code!**
  - *Software architecture is not about drawing seven boxes and add a few lines between them!*
  - *It is about ensuring that our code, down to statement level, adheres to the architectural design that we specify!*

Example

- A case study: *PizzaLand*
  - Rudimentary pizza ordering system, designed for a short course
    - Has a SQL server as storage engine

- Original code was the *copy-from-the-web* code
  - Functionally correct! *Open a connection pr request…*

```java
public String order(String customerName, Ingredient ingredient1, Ingredient ingredient2) {
  String lastId = null;
  try {
    openConnection();
    statement = connect.createStatement();
    statement.executeUpdate(
            s: "INSERT INTO " + ORDER_TABLE_NAME + " values "
                + "(default, \"" + customerName + "\", "
                + "\"" + ingredient1 + "\", "
                + "\"" + ingredient2 + "\", "
                + "FALSE);" );
    // Next, get that auto created ID as the order ID
    resultSet = statement.executeQuery( s: "SELECT LAST_INSERT_ID();");
    // advance resultSet - to the first item which is the last id
    resultSet.next();
    lastId = resultSet.getString( i: 1);
  } catch (Exception e) {
```

```java
private void openConnection() throws SQLException {
  // Setup the connection with the Maria DB
  String connectionString =
          "jdbc:mariadb://" + hostname + ":" + port + "/" + database + "?"
                  + "user=" + username + "&password=" + password;
  connect = DriverManager.getConnection(connectionString);
}
```

Example

- This coded was exposed to a load test
  - 450 threads issuing order/review/finish requests
- Result – no errors ☺
  - ~ 145 transactions/second, Median response time: 1650ms
- Then I rewrote it using the C3P0 connection pool
  - (And I am not kidding, it *is* called C3P0 ☺)
- Result

```java
public Connection getConnection() throws SQLException {
  return connectionPool.getConnection();
}
```

  - ~ 260 transactions/second, Median response time: 7ms
- *That is a factor 235 faster, in median response time!*
  - *Energy: 29% less energy pr request (48mJ versus 67 mJ)*

Course Website

- All information can be found on BrightSpace

- Or at the backdoor to slides…

- [Demo]

Contacts

- If you have questions on...
  - Use the web forum
    - Set it to notify you by mail
  - ... Or contact me by mail (welcome letter)

- AU's GDPR policy ☹
  - … is a closed system, so check Brightspace/AU mail often…

- Education (EVU):
  - Me again...

Organization

The major 'components'

- Seminars
  - Three in each course
  - Seven hours on selected Fridays
    - Lecturing, group exercises, discussions, lunch
  - Cover ~2-3 weeks of material
- Weekplans (**heartbeat!**)
  - Learning goals, reading, (slides), exercises
- Mandatory exercises = *Primary learning vehicle*
  - In groups of **two to three** persons
  - Must be passed to attend exam
- Exam
  - Graded on 7 scale

Henrik Bærbak Christensen 15

Note about Scheduling

- Week 7 ("vinterferie") does not exist at University…
  - But of course does for many of you…
  - Seminar 2 was moved late November to avoid U7!

- First course is a bit 'uneven'

- Exam
  - 28 March

Template for seminars

- Presentations
  - outlining some topics

- Work in groups
  - analysis, reading, application, discussion

- Wrap-up
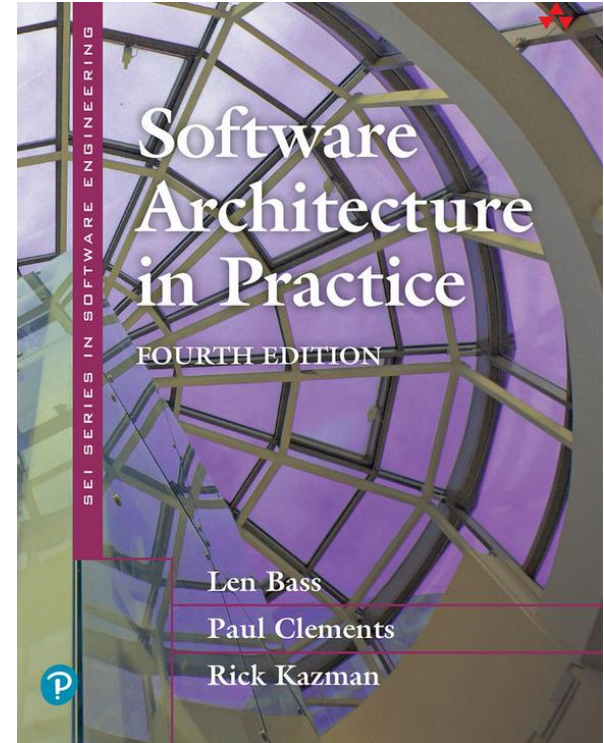 - common understanding...

I will not cover all…

- I am terribly chatty ☹
  - Maybe my stories are funny, or sleepy?

- I will try to make working seminars
  - You will work more and listen less

  - I will not cover all
    - Not all slide sets are going to be presented
    - Reading material will be left – for reading!

Literature

AARHUS UNIVERSITET

- Software Architecture in Practice

- Academic angle
  - Terminology and Concepts
  - Classification and Framework
- Practical angle
  - Concrete techniques
  - Case studies

- Many well-proved techniques

Software Architecture in Practice
FOURTH EDITION

Len Bass
Paul Clements
Rick Kazman

Research papers

- Other material are often (research) papers or…

    - I will make them available through the 'Literature (Download)' link in BrighSpace…

    - If I forget ☺, drop a note on the Forum …

Studying at University

- Maybe studying at university is a new thing for you
  - Many of you have not been "in school" for years
    - (But remember, this is not a "school" – it is a "study")
      - In "school" the teacher knows the answer, not true here ☺

- A few hints
  - Learn to read *fast* ("at skimme") to spot the pieces of info you need to know/remember (or remember where is!)
    - Use my slides as main resource as to 'what is important'…
  - *It is a contact sport!*
    - You do not learn heart surgery only by reading a book!
      - *Practice, practice, practice*

Practical Note

- In course 2, I will introduce a mandatory assignment about *energy awareness…*

- Which means – We need to measure power consumption
  - … of a running system under realistic load

- **Which requires that you have access to**
  - At least one *additional* computer/laptop pr group
    - We cannot generate the load on that machine (why not?)
  - … which has a power meter installed
    - **Powerstat installed on a non-VM Ubuntu 22.04 LTS OS**

"If you think good architecture is expensive, try bad architecture."

Brian Foote and Joseph Yoder

The Group Aspect

Working on the Mandatory

Working in Groups

- Always somewhat of a challenge
  - May work excellent
    - Improve morale, support each other, supplementary skills, discussions lead to improved understanding
  - May work less well
    - Differences in goals, ambition levels, working patterns, skill sets, commitment
    - *Only bad things happen quickly*

- *Form groups, be open on problems, stay in touch with me, be prepared to change, rotate today*

Let us start the process…

- Please present *yourself*

    – Expected outcome of SAiP

    – Group aspects
        - Geography – where are you located

        - Working habits

        - Exposure to tool stack: Java and friends

Questions?