

Copyright (unless noted otherwise): Olaf Zimmermann, 2017. All rights reserved.

## Concept/Topic: Software Architecture Fundamentals - Motivation, Definitions, Role

### Context and Motivation

Every system has an architecture. Many systems are too large, too complex or too important to let this architecture simply emerge and evolve, e.g. through frequent stakeholder interactions and continuous refactoring (although this opinion is sometimes stated in certain developer communities, more on that later). See introductory article by P. Eeles<sup>1</sup> for additional context information and motivation, as well as Chapter 1 in G. Starke's book "Effektive Softwarearchitekturen" (ESWA), 7. Auflage, Hanser-Verlag (in German).

### Definition(s)

Three definitions of Software Architecture (SWA) were given in the lecture:

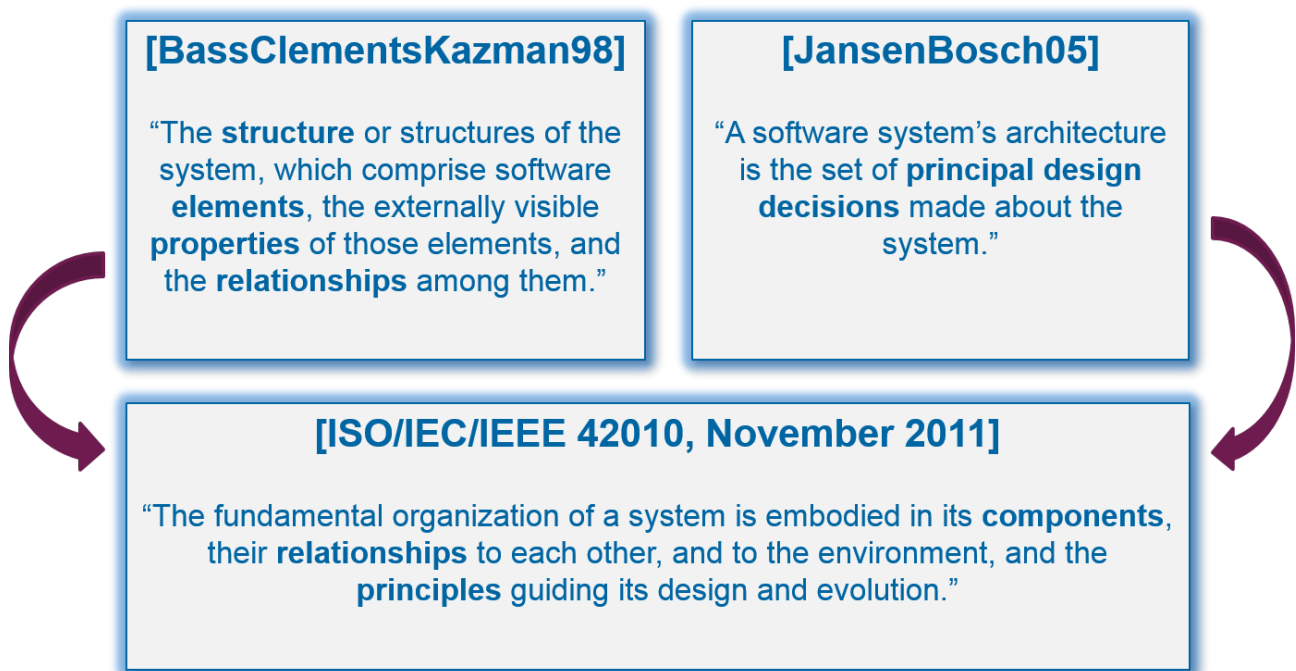


Figure 1: Three definitions of Software Architecture (SWA)

### Examples

Two of the exemplary architectures from the lecture are described in two conference papers presented at OOPSLA 2004<sup>2</sup> and OOPSLA 2005<sup>3</sup>.

The figure "Sample architecture overview diagram" shows the layered architecture overview diagram of one of them (with overlaid decisions made).

<sup>1</sup><https://www.ibm.com/developerworks/rational/library/feb06/eeles/index.html>

<sup>2</sup><http://soadecisions.org/soad.htm#oopsla04>

<sup>3</sup><http://soadecisions.org/soad.htm#oopsla05>

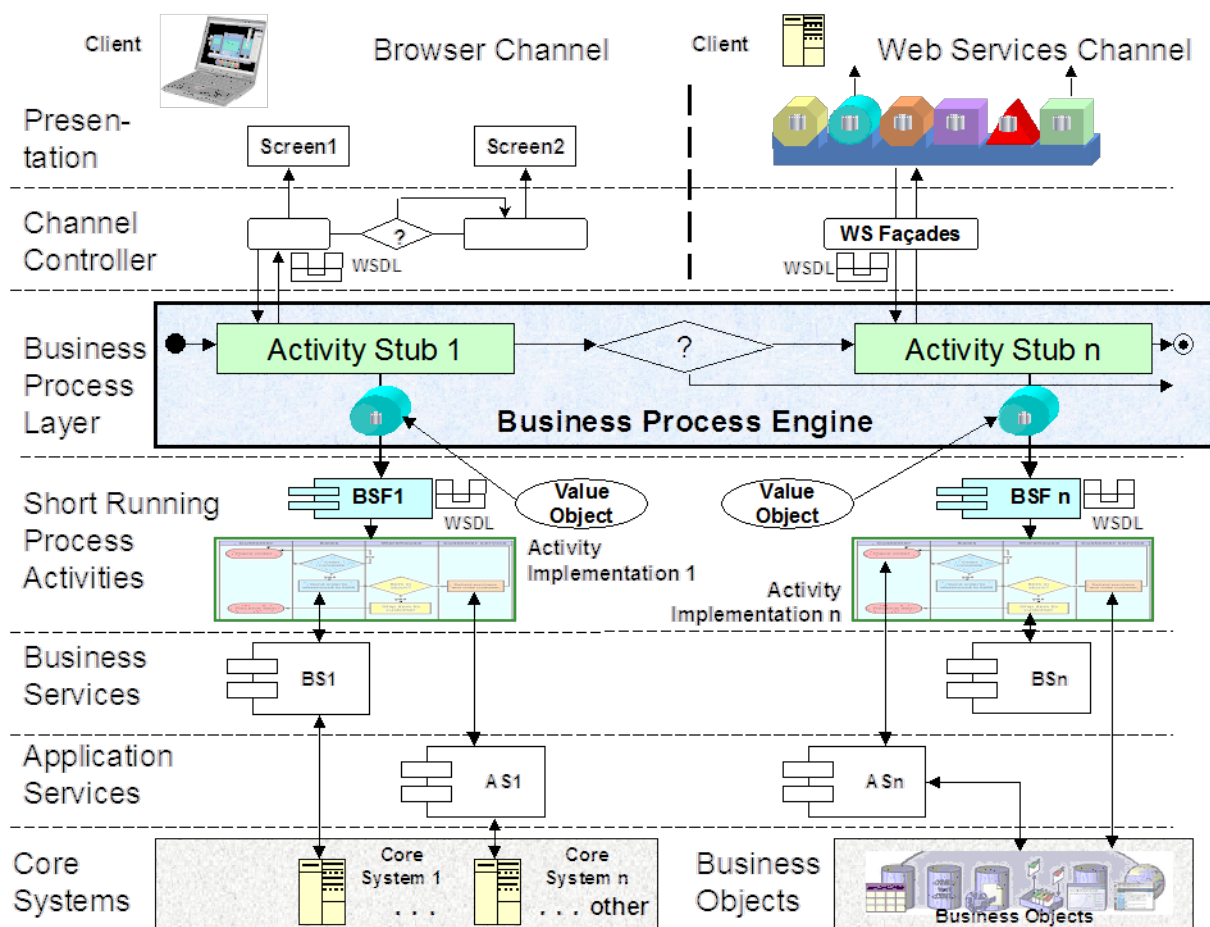


Figure 2: Sample architecture overview diagram

## The Role of the SWA a.k.a. What do Architects (Really) Do?

Responsibilities of a software architect in custom application development can be summarized as:

- Synthesizes technical solution from requirements (supported by methods)
- Technically leads project and estimates development efforts
- Coaches developers and other technical staff

Key concepts when doing so are quality attributes, architectural patterns, and architectural decisions. Quality attributes are architecturally significant requirements that drive selection of architectural patterns, which is captured in the form of architectural decisions. Additional decisions deal with technology selection.

See introductory article by P. Eeles<sup>4</sup> for more detailed coverage, as well as Chapter 1 in the ESWA book by G. Starke (in German).

P. Kruchten, one of the creators of the Unified Process (UP), presents his view in a JSS article called What do software architects really do?<sup>5</sup>; it matches the lecturer's personal experience well. Other book authors and conference presenters such as S. Brown, M. Fowler and S. Tilkov point out that architects should be close to, and involved with, technology and code, see this online article<sup>6</sup> this IEEE Software paper<sup>7</sup> and this presentation<sup>8</sup>. Architects in particular must be "generalizing specialists"<sup>9</sup> to use a term introduced by S. Ambler as part of his *agile modeling* initiative.

The Guest Editorial of the 2016 IEEE Special Issue "Role of the Software Architect"<sup>10</sup> discusses how the role is currently changing due to recent advances in methods and technology. The authors of the book "Continuous Architecture" shared their thoughts on the role and how it changes here<sup>11</sup>.

## Architecting Phases

Three phases were introduced in the lecture today; they will serve as one of the organizing principles for the remaining lessons.

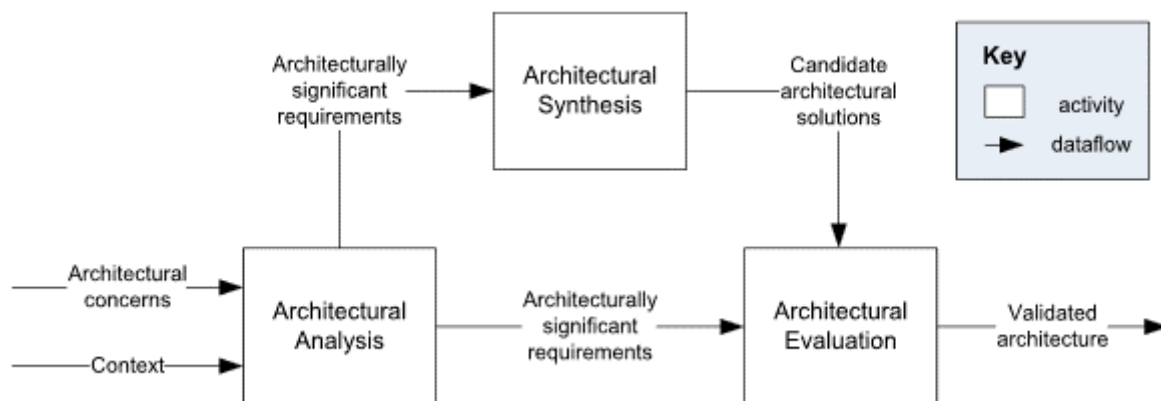


Figure 3: The three general architecting phases: Analysis, Synthesis, Evaluation (a.k.a. Requirements Elicitation, Design, Review)

See links in **Context and Motivation** section for application examples.

<sup>4</sup><https://www.ibm.com/developerworks/rational/library/feb06/eeles/index.html>

<sup>5</sup>[https://pkruchten.files.wordpress.com/2010/05/kruchten\\_2008\\_journal-of-systems-and-software.pdf](https://pkruchten.files.wordpress.com/2010/05/kruchten_2008_journal-of-systems-and-software.pdf)

<sup>6</sup><https://www.infoq.com/articles/brown-are-you-a-software-architect>

<sup>7</sup><http://martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf>

<sup>8</sup><https://speakerdeck.com/stilkov/why-architects-fail>

<sup>9</sup><http://www.agilemodeling.com/essays/generalizingSpecialists.htm>

<sup>10</sup><http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7725214>

<sup>11</sup><https://pgppgp.wordpress.com/2016/09/25/role-of-the-architect/>

## Follow-On Topics and Concepts

- Architectural Significance<sup>12</sup>
- Viewpoints<sup>13</sup>
- Agile Architecting<sup>14</sup>
- Methods<sup>15</sup>
- SMART NFRs<sup>16</sup>

## More General Information on Software Architecture

The foundations of the discipline date back to late 1990s (industry and academia):

- Bass/Clements/Kazman “Software Architecture in Practice”
- Shaw/Garlan “Software Architecture – Perspectives on an Emerging Discipline”
- Buschmann et al. “Patterns of Software Architecture” (POSA)

See IFS website Architectural Knowledge Hubs<sup>17</sup> for a curated list of pointers; for instance, P. Kruchten’s selection of his own articles on architecture<sup>18</sup>, all of which are worth reading. iSAQB maintains an e-book collecting SWA terms (with translations from English to German and vice versa), <https://leanpub.com/isaqbglossary>.

---

<sup>12</sup>./ZIO-ArchitecturalSignificanceFactSheet.pdf

<sup>13</sup>./ZIO-ViewpointFactSheet.pdf

<sup>14</sup>../lesson2/ZIO-AgileArchitectingFactSheet.pdf

<sup>15</sup>../lesson2/ZIO-MethodFactSheet.pdf

<sup>16</sup>../lesson2/ZIO-SmartNFR-FactSheet.pdf

<sup>17</sup><https://www.ifs.hsr.ch/index.php?id=13193&L=4>

<sup>18</sup><https://philippe.kruchten.com/architecture/>