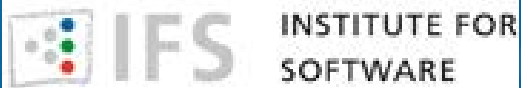


Software Architecture Metrics (SAM) Workshop ICSE 2015

METRICS FOR ARCHITECTURAL SYNTHESIS AND EVALUATION – REQUIREMENTS AND COMPILATION BY VIEWPOINT



AN INDUSTRIAL EXPERIENCE REPORT

Olaf Zimmermann

The Open Group Distinguished (Chief/Lead) IT Architect

Institute for Software, FHO HSR

Florence, May 16, 2015

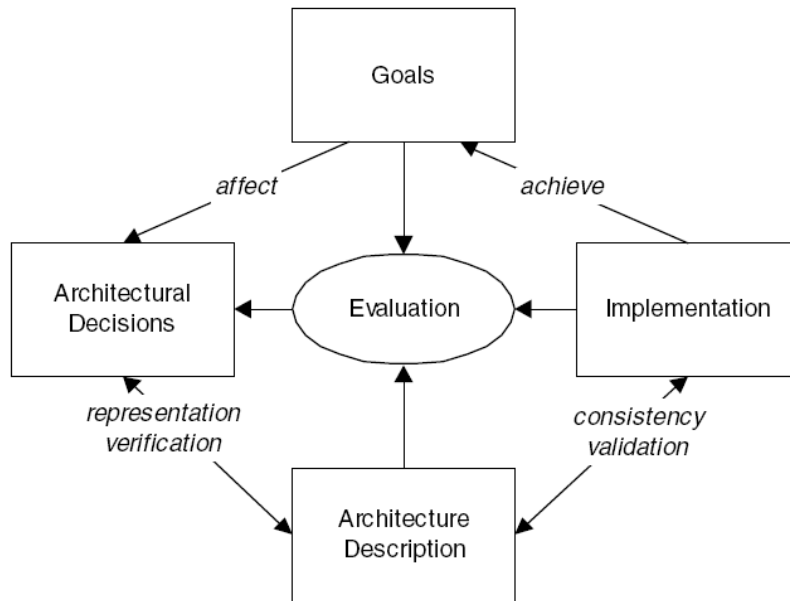


HSR

HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

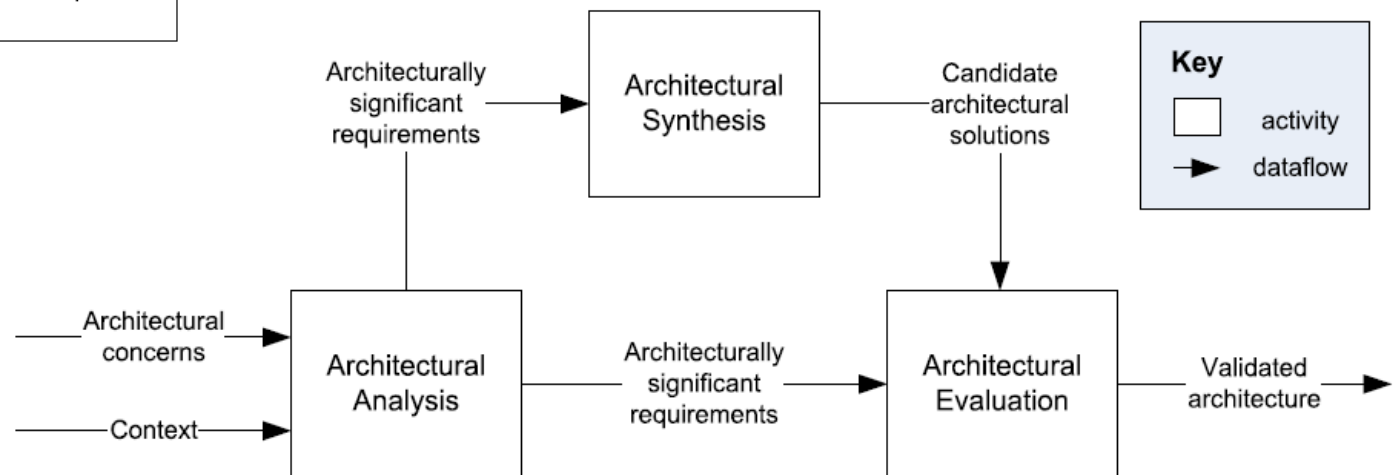
FHO Fachhochschule Ostschweiz

Background/Context: Three Types of IT Architecting Activities



Source: Hofmeister C., Kruchten P., Nord, Obbink J. H., Ran A., America P., A General Model of Software Architecture Design Derived from Five Industrial Approaches. Journal of Systems and Software 80(1), Elsevier, 2007. Pages 106-126.

How do architectural metrics fit in?



Exemplary Enterprise Information System – IT Architect's View



Reference: IBM,
ECOWS 2007,
OOPSLA 2005

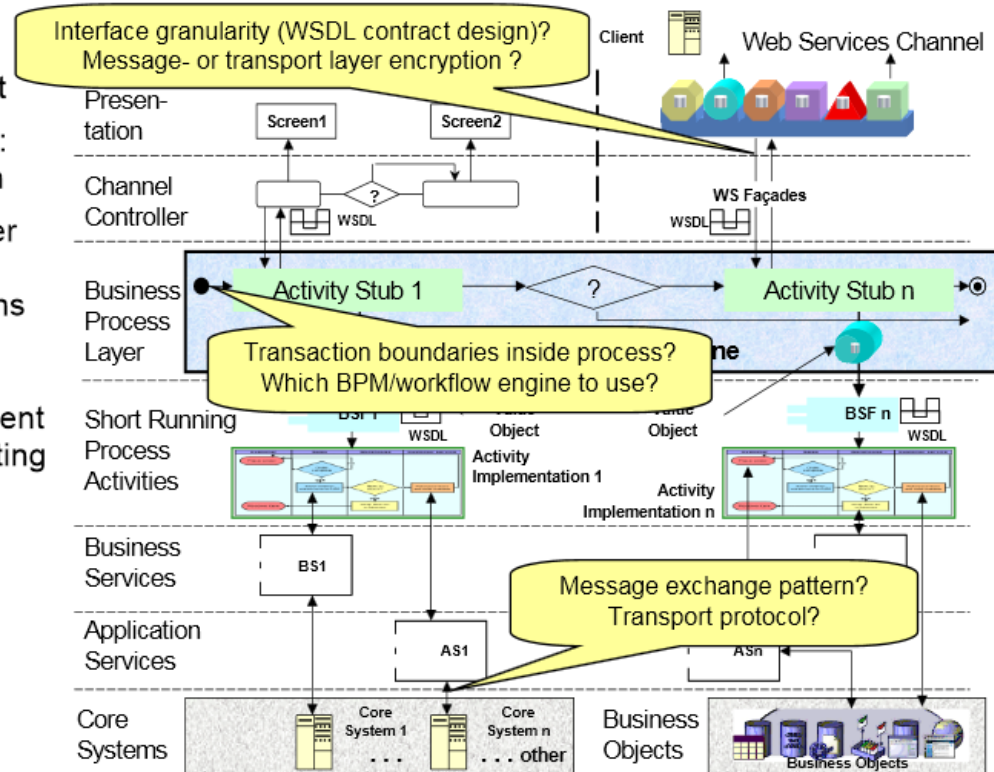
Multi-Channel Order Management SOA in the Telecommunications Industry (in production since Q1/2005) [OOPSLA 2005]

■ Functional domain

- Order entry management
- Two business processes: new customer, relocation
- Main SOA drivers: deeper automation grade, share services between domains

■ Service design

- Top-down from requirement and bottom-up from existing wholesaler systems
- Recurring architectural decisions:
 - Protocol choices
 - Transactionality
 - Security policies
 - Interface granularity



- **Utilize Architecture Metrics (AMs) during *architecture and design reviews***
 - to define scope and to assess architectural fitness and adherence to/deviation from recommended practices.
- ***Indicate complexity and technical risk***
 - e.g., to be used as input to effort estimations and project management
- ***Measure project progress***
 - on a technical level (during architectural synthesis)
- **Support architect during *transition***
 - from design-time quality attribute specifications to runtime Service Level Agreements (SLAs) and contracts (still in architectural synthesis)
- ***Benchmark architectures***
 - in domain (business) context as a variant of architectural evaluation

Critical Success Factors (CSFs) for SAM

■ ***Expressivity and elicibility***

- AMs should be able to support the five use cases effectively and efficiently.
- Little extra effort to obtain them from software architecture documents and code

■ ***Intuitivity***

- AMs should be self-explanatory: both unit and unit of measurement as in physics must be defined, value ranges should be specified.
- The AM semantics should be defined at least informally (e.g., by way of examples and counter examples).

■ ***Unambiguity***

- AMs should be well defined and use viewpoint and component/connector terminology, e.g., from IEEE 42010, patterns books, a recognized design method, or from the literature about architectural styles.

■ ***Sensitivity***

- Small changes in the architecture should not lead to radically different AM values
- AMs should not produce any surprising and misleading evaluation results.

Architectural Metrics (AMs) Identified in ICSE SAM Paper

Viewpoint	Architectural Metric (AM)	Type (Unit of Measurement)
<i>Scenario Viewpoint</i>	Number and weight of use cases Number of secondary actors (and cadence of external interface connections) Specificity and measurability of NFR/quality attribute specifications	Counter (1...1000) Counter and score Binary score
<i>Logical Viewpoint</i>	Number of external interfaces and number of interface invocations Number of components and connector per component	Counter Counter
<i>Development Viewpoint</i>	(out of scope of this report)	n/a
<i>Process Viewpoint</i>	Process Counter Process Coordination Means Interprocess Communication (IPC) and Remote Call Counter Application State and User Session State Workload Profile	Counter Index/Score Counters Size (Bytes) Aggregated (Complex)
<i>Physical Viewpoint</i>	Tier Counter Clustering Index	Counter Index/Score
<i>Architectural Decision Viewpoint</i>	Number of architecture design problems solved Number of options considered per problem	Counter Counter
<i>Information Viewpoint</i>	Data model size and structure (e.g., number of entities and entity relationships) Transaction management profile, e.g. number of system transactions and their size/duration	Index/Score Aggregated (Complex)
<i>Patterns Metrics</i> (here: POSA, PoEAA, EIP books)	E.g. number of layers, number of controllers in MVC pattern E.g. length and complexity of EIP integration flows	Counter Index/Score
<i>Domain- and Style- Specific Metrics</i> (JEE, SOA, MOM, RDB)	E.g. number of servlets, number of message channels E.g. number of SQL tables, queries, foreign key relationships	Counter Counter

Workshop Questions

■ What is your message?

- *Metrics matter – if collected and interpreted adequately (“operating range”)*

■ What issues are you facing?

- *So many architectural design issues and options (and related metrics), so little time (to compile and interpret the metrics)*

■ What question are you proposing for further discussion?

■ *Context*

- *Do we need context metrics?*
- *Are the use cases and CSFs context-specific?*
- *Do the metrics and their use change by application domain and software genre?*

■ *Roles and viewpoints*

- *Do different architects need different metrics (e.g., enterprise architects)?*
- *Which information to capture per metric?*

■ *Relationship to other specializations in SE*

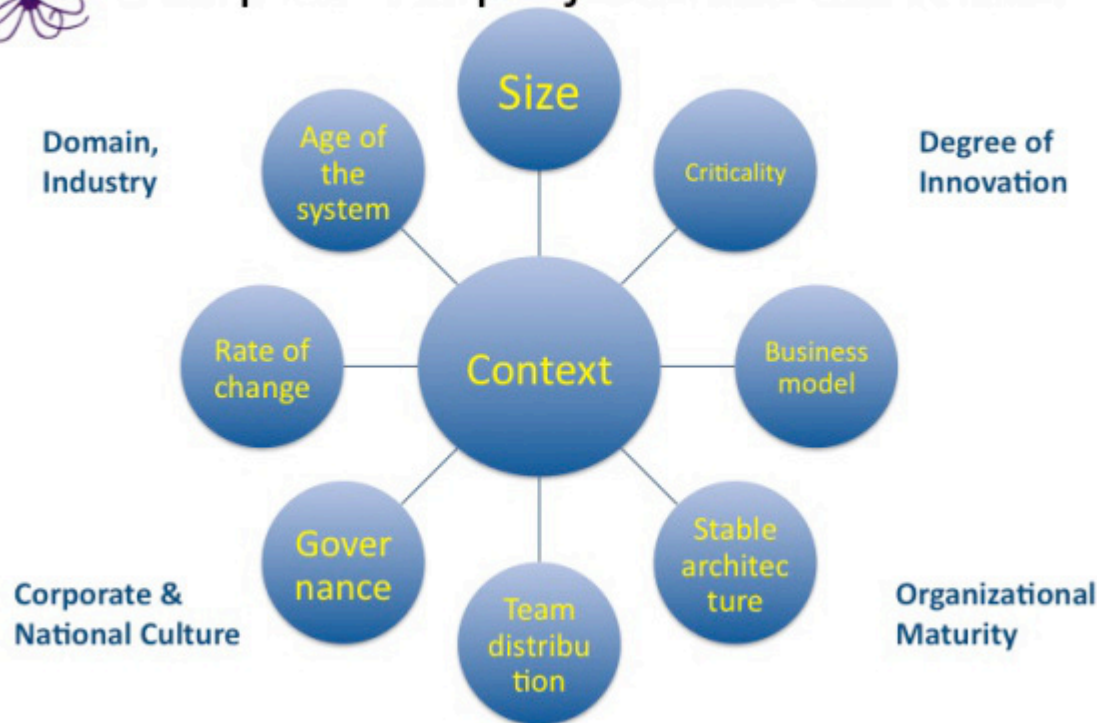
- *Why and how must architecture metrics be different from code metrics?*

The Context Octopus: Eight Dimensions

- <http://philippe.kruchten.com/2009/07/22/the-context-of-software-development/>



Octopus: “All projects are different!”



Feb. 2011

Copyright © 2005-11 by Philippe Kruchten

4

■ Discussion

- [How the paper made the discussant think differently about his or her own work]
- [Some advice for the authors in continuing their work]
- [Discussion questions based on the presented paper]