

ECE4574 – Large-Scale SW Development for Engineering Systems

Lecture 16 – Thoughts on SW Architecture

Creed Jones, PhD

Course Updates

- Project Day, Wednesday, Oct 25
 - Please come to class and sit with your group. It's a chance for me to meet briefly with each of you and for you to work on your project - perhaps your sprint report.
 - Online students, join the lecture zoom session and I will meet with your teams first.
- HW3 is posted
 - Due Friday, November 10
- Project
 - Sprint 2, week 2
- Next Quiz (6) is Wednesday, November 1

Key Topics for Today

Software Architecture in Practice: Challenges and Opportunities

- Concept of Software Architecture
- Software Requirements
- Software Design
- Software Construction and Testing
- Software Maintenance

Much of today's lecture comes from "Software Architecture in Practice: Challenges and Opportunities" by Wan et al

- Wan, Zhiyuan, et al.
"Software Architecture in Practice: Challenges and Opportunities." arXiv preprint arXiv:2308.09978 (2023).
- Posted to Canvas and available through VT library.

12 [cs.SE] 22 Aug 2023

Software Architecture in Practice: Challenges and Opportunities

Zhiyuan Wan
Zhejiang University
Hangzhou, China
wanzhiyuan@zju.edu.cn

Yun Zhang*
Hangzhou City University
Hangzhou, China
yunzhang@hzcu.edu.cn

Xin Xia
Huawei
Hangzhou, China
xin.xia@acm.org

Yi Jiang
Huawei
Shanghai, China
jiangyi54@huawei.com

David Lo
Singapore Management University
Singapore, Singapore
davidlo@smu.edu.sg

ABSTRACT

Software architecture has been an active research field for nearly four decades, in which previous studies make significant progress such as creating methods and techniques and building tools to support software architecture practice. Despite past efforts, we have little understanding of how practitioners perform software architecture related activities, and what challenges they face. Through interviews with 32 practitioners from 21 organizations across three continents, we identified challenges that practitioners face in software architecture practice during software development and maintenance. We reported on common software architecture activities at software requirements, design, construction and testing, and maintenance stages, as well as corresponding challenges. Our study uncovers that most of these challenges center around management, documentation, tooling and process, and collects recommendations to address these challenges.

conducted interviews with 32 participants involved in the design, implementation and maintenance of software architectures. Our research question is *what are the software architecture related activities performed in practice, and the corresponding challenges faced by practitioners?* Interview participants come from 21 organizations of varying sizes, from small startups to large technology companies. They have diverse roles in software development and maintenance, including architect, development, testing and project management. During the interviews, we explored various aspects of software architecture practice, including the architectural styles applied, techniques and processes followed, and tools utilized. We also sought to identify where challenges arise in software architecture practice during software development and maintenance.

We observe that challenges in software architecture practice surface at different stages of software development and maintenance process: (1) Evolution and changes of software requirements at the requirements stage; (2) Design documentation requirements

How is Software Architecture done today and what are the major challenges? Let's look at a recent paper

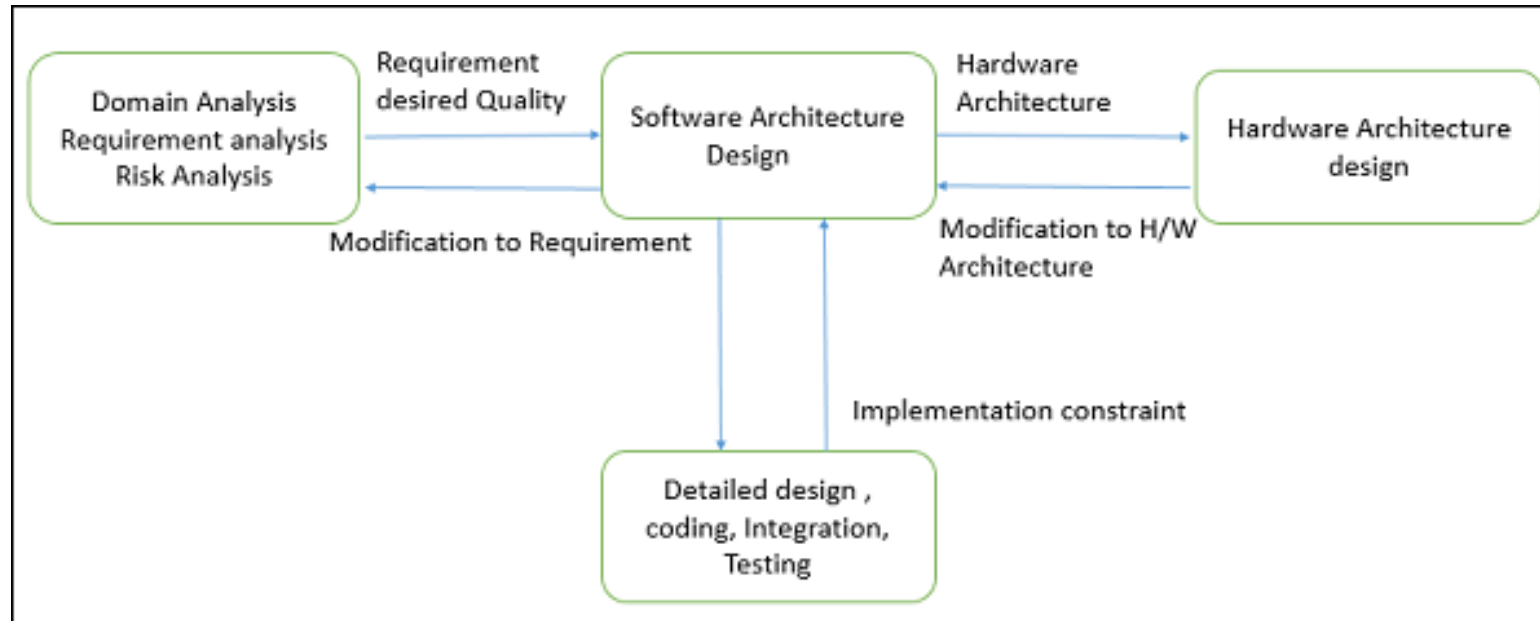
- The authors of this paper interviewed 32 “participants involved in the design, implementation and maintenance of software architectures”.
- They asked questions about what the architects do and what challenged them (problems).

Table 1: Participant and company demographics.

- The data is current (2021-2022)
- The methodology of survey and analysis conforms to good practice

Type	Break-down
Participant Role (32)	Architect (17), Development (13), Project Management (11), Testing (2)
Participant Seniority (32)	10 years of experience or more (14), 5-10 years (17), under 5 years (1)
Company Type (21)	Big tech (7), Non IT (6), Mid-size tech (5), Startup (3)
Company Location (21)	Asia (14), North America (5), Europe (2)

Concept of Software Architecture



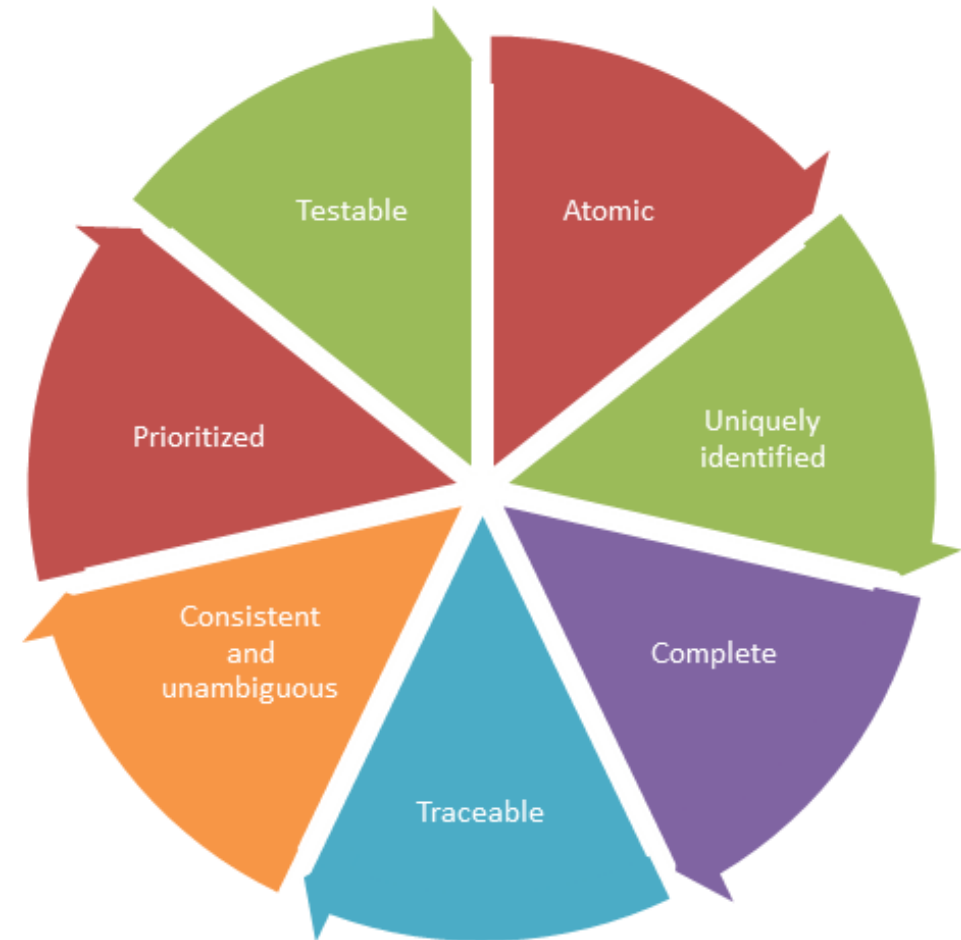
- “Software architecture refers to a collection of design decisions that affect the structure, behavior and overall quality of a software system.”
- It's both technical and *socio-technical*
 - How are design decisions made
 - How are they communicated
 - How are they implemented

Design Decisions are the essence of defining an architecture (and all of a design)

- What is the logical breakdown of the work?
- Which known architectural pattern should I use (if any)?
- What components should I reuse / buy / open-source / build?
- What development methodologies and practices should I use, and where?
- What personnel should I use, in what structure?
- What requirements may need to be negotiated if possible?

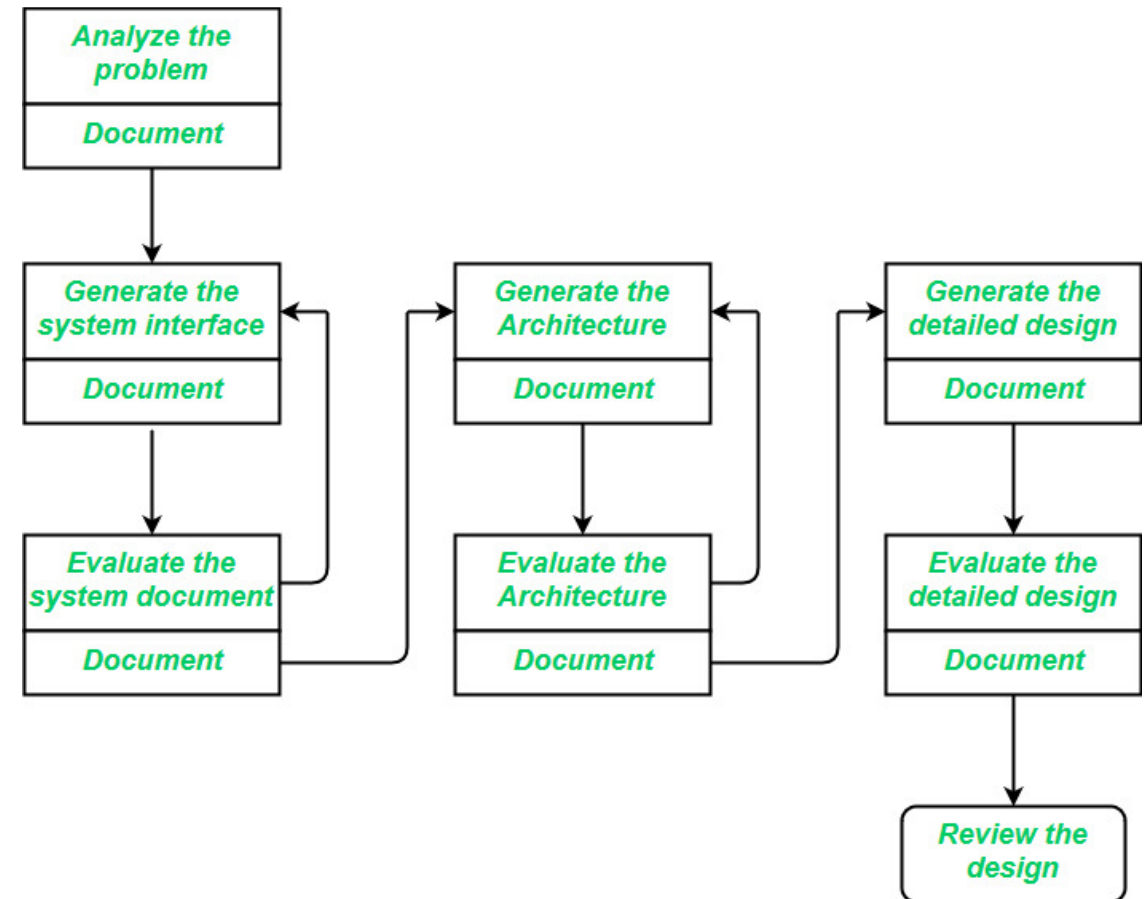
Software Requirements are the driving force behind architectural decisions

- Some requirements are met by the overall system
 - Can't point to a specific place
- Changes in requirements can have major impact
 - Difficult or impossible to anticipate
- Think and speak in terms of tradeoffs



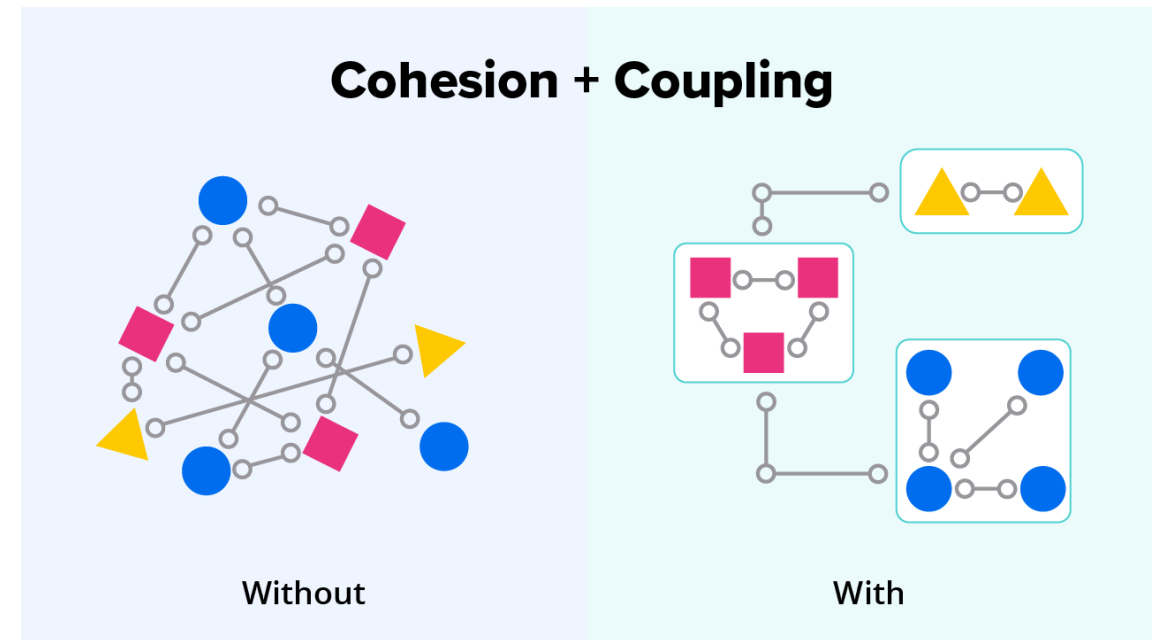
Software Design – Documentation

- It's difficult to completely document a software architecture
 - Lots of tools exist
 - Everyone finds them incomplete
 - No one uses formal methods for testing the models
- “Architecture documentation becomes obsolete as software evolves”
 - Rarely fixed to keep current
- Good specification of the interfaces is most important



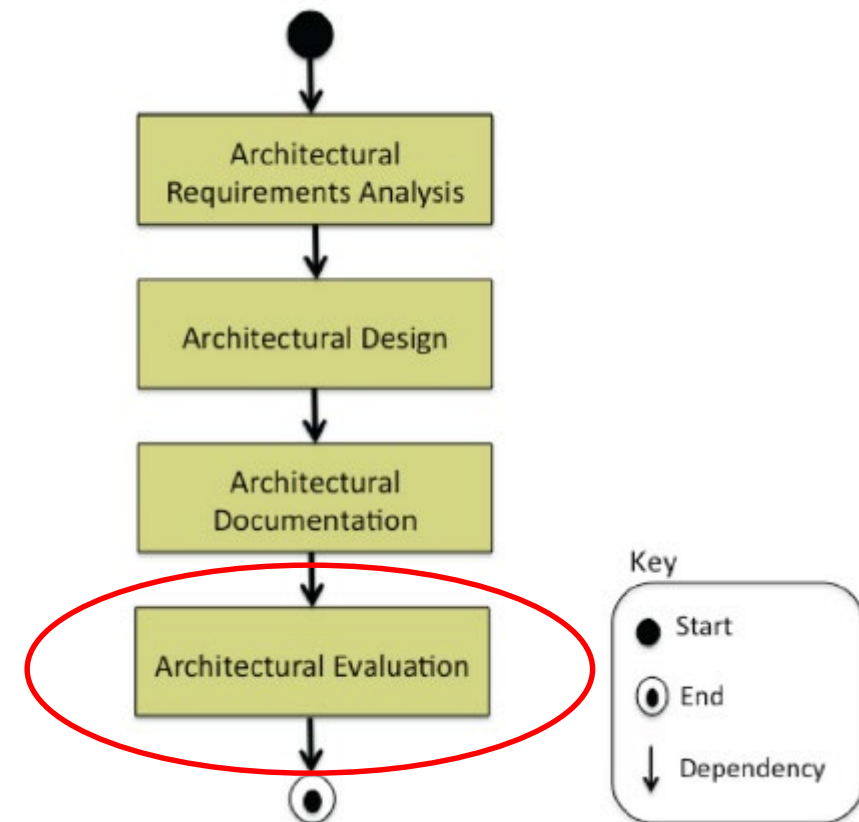
Software Design – Do Architects really apply the common principles (high cohesion, low coupling, etc.)?

- The boundaries between components is often unclear
 - Difficult to assess coupling
- Interdisciplinary knowledge is required
 - Skill in several areas of SW
 - Problem domain knowledge
 - Existing code
- Changing requirements can undo a good, principled design



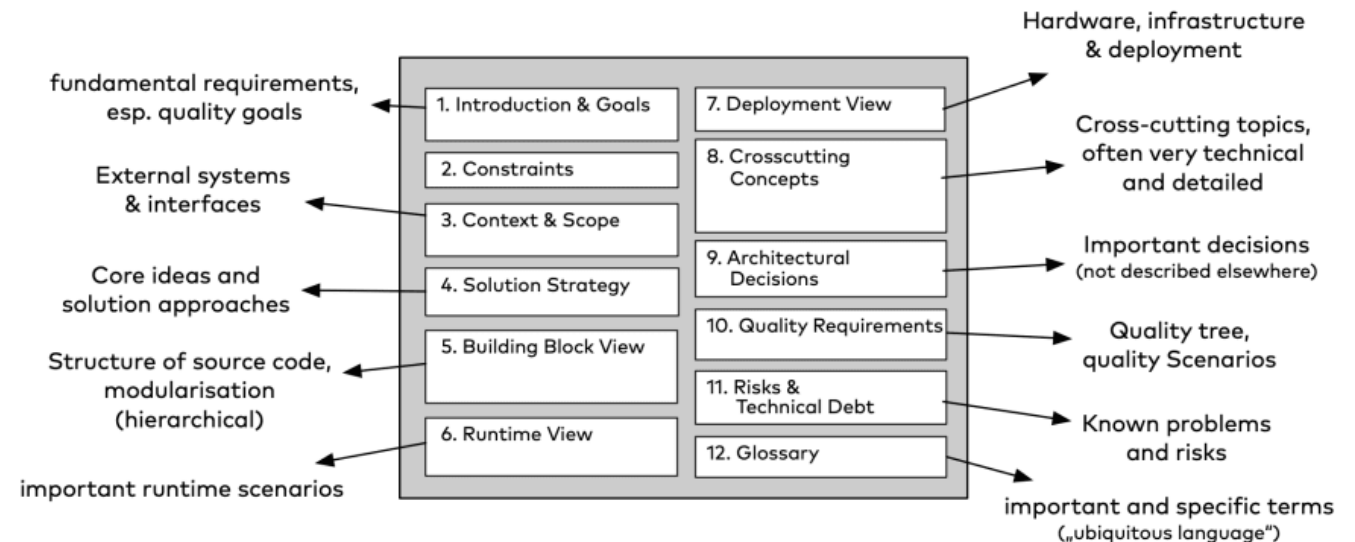
Software Design – Can we analyze a design for quality?

- Yes, but it's a significant effort
 - “requires a standard process, active involvement of external experts, and tool support”
- It's common to spend more time examining certain components or sections
 - Critical areas
 - New components
 - Complex sections
- Effective and universal measures don't exist



Software Construction and Testing – Are we actually following the architecture, or working around it?

- It's rare to check whether the detailed design and implementation is adhering to the architecture
 - Contorting the middleware, for example
- Documentation of the architecture becomes obsolete and isn't always maintained



Technical debt is a useful concept for short-term decisions during development

- “technical compromises that yield short-term benefits but hurt the long-term success of software systems”
 - temporary linkages to avoid protocol implementations
 - “just hard-code this for now, we can come back and fix it later”
 - lack of documentation (comments)
- Also, bad coding practice
 - “without a deep understanding of a programming language or technology, developers tend to misuse the features of the language or technology when performing specific programming tasks... unintentionally incur technical debt.”

-

Software Maintenance – as a product evolves, does the architecture change, and if so, how?

- Tools don't exist to make this efficient
- A common way is to use a set of testing frameworks and apply them periodically
- If product requirements evolve, the architecture can:
 - Stay the same
 - Change gradually
 - Be completely reworked



The authors identified four areas of improvement for software architecture

1. Management
need to allocate sufficient resources
2. Documentation
have better tools for documentation, and use (create and refer to) them
3. Tooling
architecture conformance checking, traceability
4. Process
document the reasons for design decisions

Key Topics for Today

Software Architecture in Practice: Challenges and Opportunities

- Concept of Software Architecture
- Software Requirements
- Software Design
- Software Construction and Testing
- Software Maintenance