
Designing Software Architecture

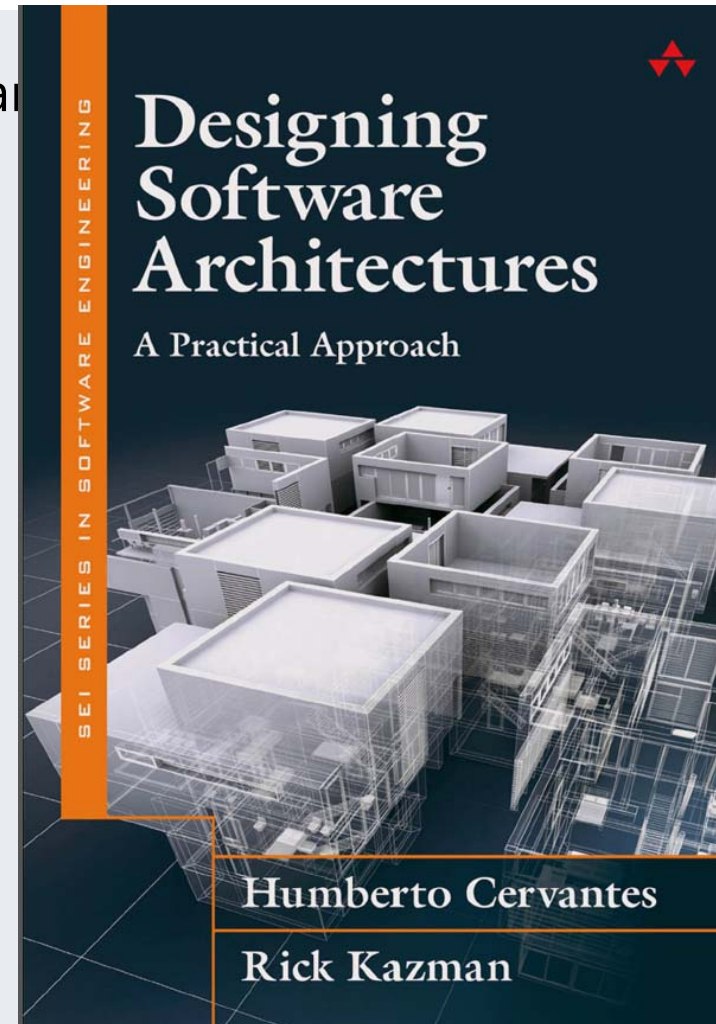
-Introduction

Eunseok Lee, Prof.

College of Software
Sungkyunkwan University

About the book

- Humberto Cervantes & Rick Kazman
- SEI Series in software Engineering
- Carnegie Mellon University
- Addison-Wesley 2016



Objectives

- Provide an introduction to the topic of software architecture;
- Discuss what architecture is and why it is fundamental to take it into account when developing software systems;
- Discuss the different activities that are associated with the development of software architecture;
- Discuss the role of the architect;
- Introduce the Attribute-Driven Design (ADD) method.

Topics covered

- Software architecture
- Importance of software architecture
- Software architecture design processes
- Software architect
- ADD(Attribute-Driven-Design)

1. Motivation

- Goal is to teach you how to design software architecture in a systematic, predictable, repeatable, and cost-effective way.
- In most fields, “design” involves the same sorts of challenges and considerations:
 - meeting stakeholder needs,
 - adhering to budgets and schedules,
 - dealing with constraints, and so forth

2. Software Architecture

- "The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both."
<Software Architecture in Practice, 3rd edition, 2012>
- "The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution."
<IEEE Std. 1471, 2000>
- "The architecture is Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution."
<ISO/IEC/IEEE 42010, 2011>

2. Software Architecture

2.1 The Importance of Software Architecture

- Inhibit or enable a system's driving quality attributes.
- The decisions made in an architecture allow you to reason about and manage change as the system evolves.
- The analysis of an architecture enables early prediction of a system's qualities.
- A documented architecture enhances communication among stakeholders.
- The architecture influences the structure of an organization, and vice versa.
- An architecture can provide the basis for evolutionary, or even throwaway, prototyping.

2. Software Architecture

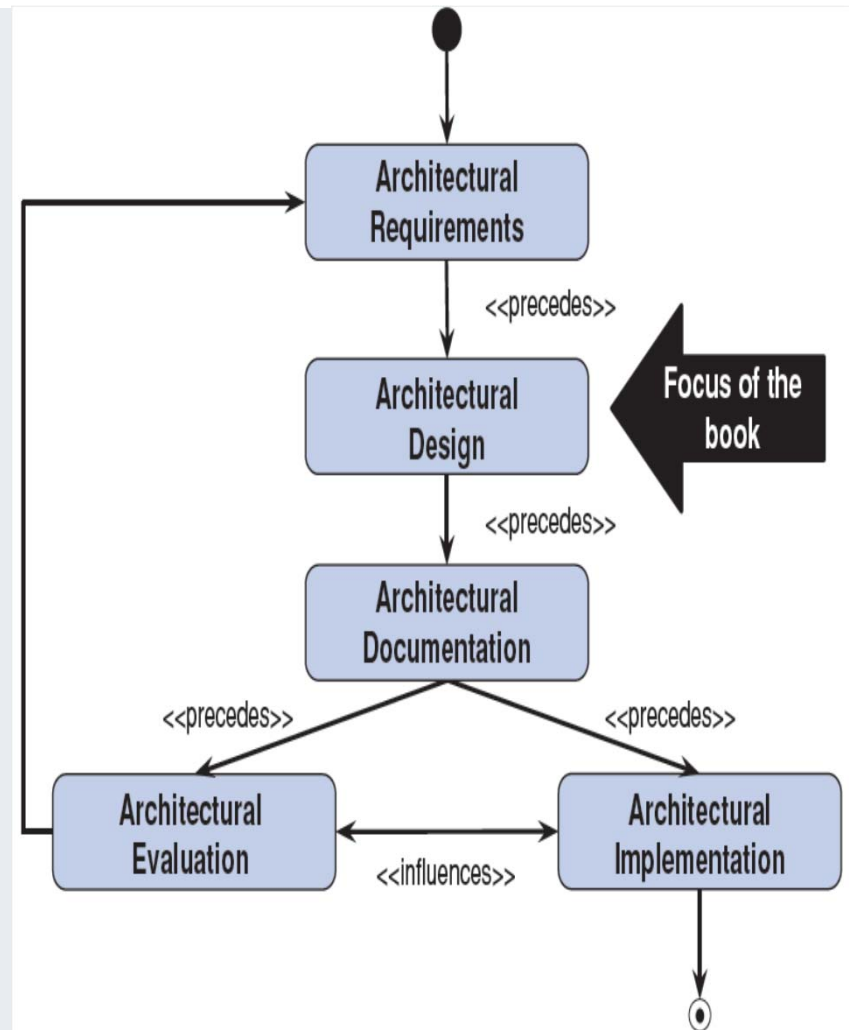
2.1 The Importance of Software Architecture(cont')

- An architecture is the key artifact that allows the architect and the project manager to reason about cost and schedule.
- An architecture can be created as a transferable, reusable model that forms the heart of a product line.
- Architecture-based development focuses attention on the assembly of components, rather than simply on their creation.
- By restricting design alternatives, architecture channels the creativity of developers, reducing design and system complexity.
- An architecture can be the foundation for training a new team member.

2. Software Architecture

2.2 SA Life Cycle Activities

- (1) Architectural requirements
- ASR(Architecturally Significant Req.)
 - Very important functionalities, constraints, and **QAs**
 - ex) high performance, high availability, ease of evolution, high security, etc
 - Guide you to choose one set of architectural structures and components over another.
 - *Architecture design drivers*



2. Software Architecture

2.2 SA Life Cycle Activities

(2) Architectural design

- Design is a translation, from the needs (requirements) to the solutions, in terms of structures
- Composed of code, frameworks, and components.
- A good design is one that *satisfies* the drivers.

2. Software Architecture

2.2 SA Life Cycle Activities

(3) Architectural documentation

- Refers to the creation of a more formal document from rough sketches.
- When the project is large, distributed teams are collaborating, or significant technical challenges exist.
- While documentation is often avoided and derided by programmers, it is a standard, non-negotiable deliverable in almost every other engineering discipline.

2. Software Architecture

2.2 SA Life Cycle Activities

(4) Architectural evaluation

- Ensure that the decisions made are appropriate to address the critical requirements
- Similar with delivering code after testing
- Typically evaluation is done informally and internally, but for truly important projects advisable formal evaluation by an external team.

2. Software Architecture

2.2 SA Life Cycle Activities

(5) Architectural implementation/conformance checking

- Need to tweak the design as the system grows and as requirements evolve.
- Major responsibility during implementation is to ensure conformance of the code to the design.
- Without conformance checking, we have no way of ensuring the quality of what is being subsequently constructed.
- None of these architecture activities is incompatible with ***Agile practices***.
- "Should I do Agile or architecture?" (X)
- "How much architecture should I do up front *versus* how much should I defer until the project's requirements have solidified somewhat?"

3. The Role of the Architect

An architect is much more than “just” a designer. Has a long list of duties, skills, and knowledge that must be satisfied if it is to be successful.

- **Leadership:** mentoring, team-building, establishing a vision, coaching
- **Communication:** both technical and nontechnical, encouraging collaboration
- **Negotiation:** dealing with internal and external stakeholders and their conflicting needs and expectations
- **Technical skills:** life-cycle skills, expertise with technologies, continuous learning, coding
- **Project skills:** budgeting, personnel, schedule management, risk management
- **Analytical skills:** architectural analysis, general analysis mindset for project management and measurement

4. A Brief History of ADD

- 1st version of ADD: (ADD 1.0, originally called ABD, for “Architecture-Based Design”) January 2000,
 - 2nd version: (ADD 2.0) was published in November 2006,
 - 3rd version: (ADD 3.0) 2016 for agile development with quick design iteration.
- ADD is the most comprehensive and most widely used documented architecture design method.
 - The first design method to focus specifically on quality attributes and their achievement
 - It includes architecture analysis and documentation as an integral part of the design process.

Key points

- Software architecture is a description of a set of major elements, communications between them, and control of them.
- Software architecture has important roles for system analysis, construction and maintenance.
- Architect has many duties and responsibilities, especially for architecture design.
- ADD is one of prominent structured method for AD.