Introduction

Arnav Gupta

September 4, 2024

Contents

1 Introduction

1.1 Terminology

Architecture analysis exists between the problem space (requirements) and the solution space (implementation and testing).

Architecture: organization of a system embodied in components and their relationships to each other and their environment

System: collection of components organized to accomplish a specific function or set of functions

Environment: determines the setting and circumstances of influences on a system

Mission: use or operation for which a system is intended by stakeholders to meet objectives

Stakeholder: individual, team, or organization with interests/concerns relative to a system

Software architecture: conceptual fabric that defines a system, is a type of design, related to parts of a system that would be difficult to change once a system is built

Architectures capture structure, communication, and nonfunctional requirements. Architecture helps with stakeholder communication, system analysis, and large-scale reuse.

System architecture is structure, **conceptual (prescriptive) software architecture** is abstract structure, and **concrete (descriptive) software architecture** is actual structure.

Live architecture: the mental model, which can be abstract or concrete, but may be fuzzy, incomplete, etc.

Complexity: architecture simplifies a system through focus on structure rather than content

Reverse Engineering: extraction of design from implementation and from developers

1.2 Architectural Degradation

When a system evolves, prescriptive architecture is modified first. In practice, descriptive architecture is often directly modified due to sloppiness, short deadlines, lack of documentation, desire for optimization, or inadequate tool support.

Architectural drift: introduction of principal design decisions into descriptive architecture that are not included in prescriptive architecture but do not violate the prescriptive architecture

Architectural erosion: introduction of architectural design decisions into a system's descriptive architecture that violate prescriptive architecture

If architectural degradation occurs, system architecture must be recovered. **Architectural recovery** is the process of determining a software system's architecture from its implementation-level artifacts.

1.3 Expressing Software Architecture

System architecture needs to express:

- decomposition into subsystems (code dependency)
- process interactions (how code calls each other)

- distribution of subsystems across networked devices (where to put them)

To form an architectural model, one must use several views of a software architecture:

- **logical view**: shows key abstractions in the system (objects, object classes)
- **process view**: shows how, at runtime, the system is composed of interacting processes
- **development view**: shows how the software is decomposed for development
- **physical view**: shows the system hardware and how software components are distributed across processors in the system
- these all related using use cases or scenarios

Architectural style is the form of structure, **reference architecture** is the general architectural for an application domain, and **product line architecture** is the architecture for a line of similar software products.

1.4 Architecture vs Design

Architecture is the structure of a system (components and connectors) and should be high level and hard to change.

Design is the inner structure of components and more low level and allows change.