Software Architecture Evaluation

Software Architecture

3rd Year – Semester 1

Lecture 14

Why Software Architecture Evaluation?

- How can you be sure whether the architecture chosen for your software is the right one?

- How can you be sure that it won't lead to disaster?

- Architecture evaluation aims to improve and find deficiencies in an architecture as early as possible in the development process.
  - Modifying the architecture during the architectural design phase is cheap, modifying the architecture later is costly!

- Architecture determines the structure of the project: schedules and budgets, performance goals, team structure, documentation organization, and testing and maintenance activities.

How

- Evaluation can be based on questioning or measuring techniques.
- Qualitative Evaluation and Quantitative Evaluation.
- Method Types
  - Scenario-based
  - Mathematical model-based
  - Simulation based
  - Experience-based reasoning

When

- The classical application of architecture evaluation occurs when the architecture has been specified but before implementation has begun.

- Early
  - At any stage in the architecture creation process to examine those architectural decisions already made and choose among architectural options.
  - Based on specification and description of the software architecture, and other sources of information, such as interviews with architects.

- Late
  - Takes place when the architecture is nailed down and the implementation is complete. Mainly used when architecture is inherited from legacy system.
  - Based on metrics. E.g. cohesion and coupling of architectural components

Challenges

- Requires an expert evaluation team to deal with unpredictable risks along with the known risks

- Lack of common understanding of the high level design

- Different stakeholders have different interests

- The quality requirements for a system may not be written properly or may not be finished when the architecture is designed

Who is involved?

- Evaluation Team
  - People who will conduct the evaluation and perform the analysis.

- Stakeholders
  - People who have a vested interest in the architecture and the system.

Planning & Process

- Before you start an evaluation, you need
  - Clearly articulated goals and requirements for the architecture
  - Select an evaluation method
  - Controlled scope (a small number of explicit goals)
  - Cost-effectiveness (it might not be necessary to do full-scale reviews for small projects)
  - Ensure the key personnel (representatives of each group of stakeholder) are available
  - Have a competent evaluation team, ideally separate from the architects and developers
  - Managed expectations
- The results of the review need to be circulated to all stakeholders afterwards in draft form, with a ranked list of potential issues found.

What are the outputs?

- Prioritized Statement of Quality Attribute Requirements
  - Having a prioritized statement of the quality attributes serves as an excellent documentation record to accompany any architecture and guide it through its evolution
- Mapping of Approaches to Quality Attributes
  - Produces a mapping that shows how the architectural approaches achieve (or fail to achieve) the desired quality attributes
- Risks and Non-risks
  - Risks are potentially problematic architectural decisions
  - Non-risks are good decisions that rely on assumptions that are frequently implicit in the architecture

Advantages

- Forces an Articulation of Specific Quality Goals
- Results in the Prioritization of Conflicting Goals
- Puts Stakeholders in the Same Room
- Improves the Quality of Architectural Documentation
- Uncovers Opportunities for Cross-Project Reuse
- The average architecture evaluation adds no more than a few days to the project schedule (small % on total project time)
- Architecture created in haste will precipitate disaster: performance goals not met, Security goals falling, customer dissatisfaction, system that is too hard to change, and schedules and budgets through the roof

How to validate a software architecture?

- Most of the common evaluation methods evaluate only 1 Quality Attribute

- There are more advanced evaluation methods available to evaluate multiple Quality Attributes

- Identifying Trade-Offs

- A suite of three methods, all developed at the Software Engineering Institute.
    - ARID: Active Reviews for Intermediate Designs
    - SAAM: Software Architecture Analysis Method
    - ATAM: Architecture Tradeoff Analysis Method

Active Reviews for Intermediate Designs (ARID)

- Method for reviewing preliminary software designs (such as for a component or a subsystem) for suitability in its intended usage context and environment

- Result in a high-fidelity design review coupled with high-quality familiarization with the design

- Stakeholder Centric: Requires active stakeholder participation

- Easy and Lightweight approach

- Does not require complete documentation

ARID: Steps

- Identify Reviewers
  - The designer who has commissioned the review works with the ARID facilitator to identify the best reviewers. These are usually the software engineers who will be expected to use the design, as they are in the best position to judge its adequacy

- Overview & Presentation
  - The designer prepares a briefing explaining the design, which is reviewed by the ARID facilitator. The designer presents the overview to the reviewers and walks through examples of using the design. A scribe captures questions and answers

- Brainstorming
  - The reviewers brainstorm scenarios for using the design to solve problems they expect to face. After a facilitated prioritization, the resulting set of scenarios operationally defines what it means for the design to be usable: If it performs well under the adopted scenarios, then it must be agreed that the design has passed the review

- Artifacts
  - The reviewers begin to jointly craft code (or pseudo-code) that uses the design's services to solve the problem posed by each high-priority scenario. The scribe records issues, problems, and places where the stakeholders get stuck

ARID: Benefits

- Helps the designers of architecture engage stakeholders and get their buy-in early in the design process

- Informs designers about whether their design is suitable for the overall system being developed

- Valuable early insight into the design's viability and allows for timely discovery of errors, inconsistencies, and inadequacies

Software Architecture Analysis Method (SAAM)

- Aims to predict the quality of a system before it has been developed

- The quality of the architecture is validated by analyzing the impact of predefined scenarios on architectural components

- Addresses concerns at the architecture design level which inherently crosscut multiple architectural components

- Scenario based evaluation

SAAM: Steps

- Specify
  - Collects Requirements & Constraints and Develop Scenarios
- Describe Architectures
  - Present Candidate Architecture(s), static & dynamic representation of the system
- Elicit Scenarios
  - Scenarios are simulated with the presence of the relevant stakeholders for Brainstorming
- Classify and Prioritize Scenarios
  - Direct & Indirect Scenarios. Voting can be used to prioritize/identify what is most likely to occur
- Evaluate
  - Evaluate Architecture with respect to Scenarios. Impact on Scenarios for Architecture is exposed
- Results
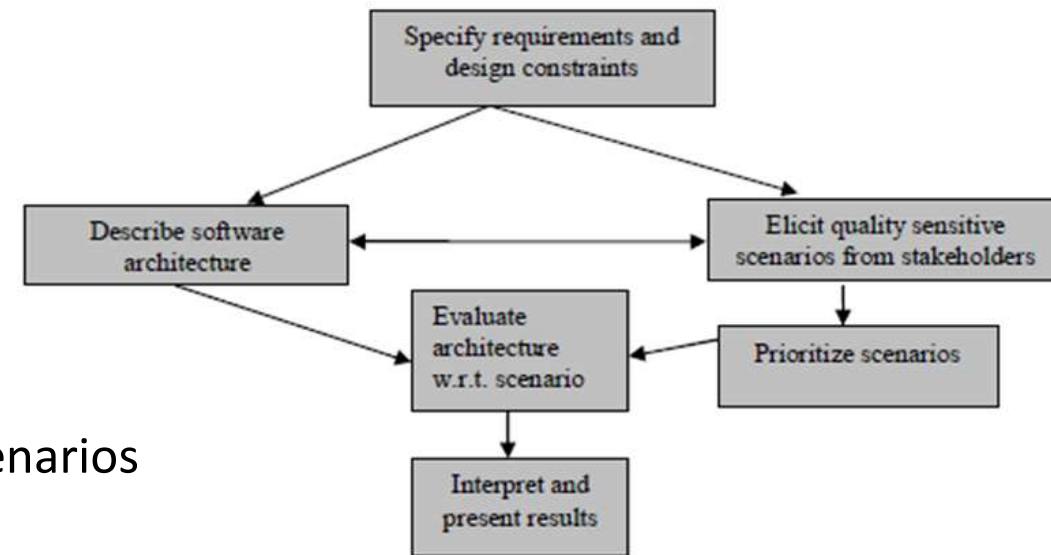
SAAM: Benefits

- This analysis helps assess the risks inherent in an architecture

- Compare candidate software architectures

- Guides the inspection of the architecture, focusing on potential trouble spots such as requirement conflicts or incomplete design specification from a particular stakeholder's perspective

Architecture Tradeoff Analysis Method (ATAM)

- A structured technique for understanding the tradeoffs inherent in the architectures of software-intensive systems

- Provides a principled way to evaluate a software architecture's fitness with respect to multiple competing quality attributes

- Is a spiral model of design: one of postulating candidate architectures followed by analysis and risk mitigation, leading to refined architectures

ATAM

- Architecture Tradeoff Analysis Method

NEXT LECTURE:

     Trade-offs

     ATAM

Other Architecture Evaluation Methods

- SALUTA: Scenario-based Architecture Level <u>Usability</u> Analysis

- ALMA: Architecture Level <u>Modifiability</u> Analysis

- Software Architecture-based <u>Reliability</u> Analysis

- Software Architecture-based <u>Performance</u> Analysis
  - Analyze to estimate performance attributes quantitatively

- FAAM: Family-Architecture Assessment Method
  - Focus on two related quality aspects (e.g. interoperability and extensibility)

Mathematical Model-based Software Architecture Evaluation

- It is important to quantitatively assess operational quality attributes

- These methods model software architectures using well-known mathematical equations

- Then, these methods use the models to obtain architectural statistics, for instance, mean execution time of a component

- These architectural statistics are used to estimate operational quality attributes. Reliability and performance are two important operational quality attributes

Late Evaluation Methods Applied to Software Architecture

- Late software architecture evaluation methods identify the difference between the actual and planned architectures

- These methods provide useful guidelines of how to reconstruct the actual architecture, so that it conforms to the planned architecture

- During the testing phase, late software architecture evaluation methods are also applied to check the compliance of the source code to the planned design

References

- https://www.sei.cmu.edu/architecture/tools/evaluate/
- http://www.win.tue.nl/oas/architecting/aimes/papers/Scenario-Based%20SWA%20Evaluation%20Methods.pdf