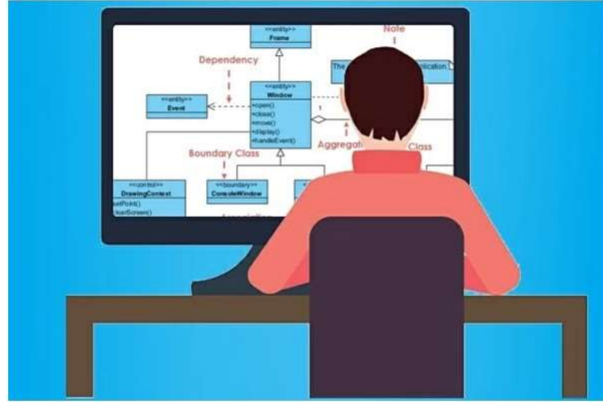


Software Design & Architecture

Spring 2022 - Week-14



مدرس: مهندس ماجد کلیم
جامعہ بحریہ، واقعہ گاہ کراچی
Engr. Majid Kaleem

WEEKLY AGENDA

TENTATIVE WEEKLY DATES		TENTATIVE TOPICS
1	Mar 7 th – Mar 11 th	INTRODUCTION TO THE COURSE; DEFINING SOFTWARE ARCHITECTURE & DESIGN CONCEPTS
2	Mar 14 th – Mar 18 th	DESIGN PRINCIPLES; OBJECT-ORIENTED DESIGN WITH UML
3	Mar 21 st – Mar 25 th	SYSTEM DESIGN & SOFTWARE ARCHITECTURE; OBJECT DESIGN, MAPPING DESIGN TO CODE
4	Mar 28 th – Apr 1 st	FUNCTIONAL DESIGN; UI DESIGN; WEB APPLICATIONS DESIGN ASSIGNMENT & QUIZ #1
5	Apr 4 th – Apr 8 th	MOBILE APPLICATION DESIGN; PERSISTENCE LAYER DESIGN
6	Apr 11 th – Apr 15 th	CREATIONAL DESIGN PATTERNS
7	Apr 18 th – Apr 22 nd	STRUCTURAL DESIGN PATTERNS ASSIGNMENT & QUIZ #2
8	Apr 25 th – Apr 29 th	BEHAVIORAL DESIGN PATTERNS
← MID TERM EXAMINATIONS →		
9	May 9 th – May 13 th	INTERACTIVE SYSTEMS WITH MVC ARCHITECTURE; SOFTWARE REUSE
10	May 16 th – May 20 th	ARCHITECTURAL DESIGN ISSUES; ARCHITECTURE DESCRIPTION LANGUAGES (ADLS)
11	May 23 rd – May 27 th	ARCHITECTURAL STYLES/PATTERNS & DESIGN QUALITIES
12	May 30 th – Jun 3 rd	ARCHITECTURAL STYLES/PATTERNS & DESIGN QUALITIES ASSIGNMENT & QUIZ #3
13	Jun 6 th – Jun 10 th	QUALITY TACTICS; ARCHITECTURE DOCUMENTATION
14	Jun 13 th – Jun 17 th	ARCHITECTURAL EVALUATION TECHNIQUES
15	Jun 20 th – Jun 24 th	MODEL DRIVEN DEVELOPMENT ASSIGNMENT (PRESENTATIONS) & QUIZ #4
16	Jun 27 th – Jul 1 st	REVISION WEEK
← FINAL TERM EXAMINATIONS →		

EVALUATING A SOFTWARE ARCHITECTURE

- 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 calamity?
- *"Marry your architecture in haste and you can repent in leisure"*
—Barry Boehm

3

WHY EVALUATE AN ARCHITECTURE?

- The earlier you find a problem in a SW project, the better off you are (the cost to fix an error in early design phase is much smaller than the cost to fix the same error in implementation/testing)
- Architecture determines the structure of the project: schedules and budgets, performance goals, team structure, documentation organization, and testing and maintenance activities.
- Architecture is the earliest point in the project where trade-offs are visible
- Architecture determines the structure of the project: schedules, budgets, performance indicators, team structure, testing and maintenance activities
- Risk management

4

WHEN CAN AN ARCHITECTURE BE EVALUATED?

- The classical application of architecture evaluation occurs when the architecture has been specified but before implementation has begun.
- Two useful variations from: Early and Late
 - **Early** - at any stage in the architecture creation process to examine those architectural decisions already made and choose among architectural options.
 - **Late** - takes place when the architecture is nailed down and the implementation is complete. Mainly used when architecture is inherited from legacy system.

5

WHO IS INVOLVED?

- **Evaluation team** - these are the people who will conduct the evaluation and perform the analysis.
- **Stakeholders** - stakeholders are people who have a vested interest in the architecture and the system.

6

OUTPUTS OF AN ARCHITECTURE EVALUATION?

- 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.

7

ARCHITECTURAL EVALUATION TECHNIQUES

1. *Scenario-Based Architecture Analysis Method (SAAM)*
2. *Architecture Tradeoff Analysis Method (ATAM)*
3. SAAM founded on Complex Scenarios (SAACS)
4. Extending SAAM Integration in the Domain (ESAAMI)
5. Software Architecture Analysis for Evolution and Reusability (SAAMER)
6. Scenario-Based Architecture Reengineering (SBAR)
7. Architecture Level Prediction of Software Maintenance (ALPSM)
8. Software Architecture Evaluation Model (SAEM)
9. *Active Reviews for Intermediate Designs (ARID)*
10. The Cost Benefit Analysis Method (CBAM)

14-Jun-2022

Engr. Majid Kaleem

8

HOW TO VALIDATE A SOFTWARE ARCHITECTURE?

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

9

ARCHITECTURE TRADEOFF ANALYSIS METHOD (ATAM)

- A structured technique for understanding the *tradeoffs* inherent in the architectures of software-intensive systems.
- ATAM provides a principled way to evaluate a software architecture's fitness with respect to multiple competing quality attributes.
- ATAM is a *spiral* model of design: one of postulating candidate architectures followed by analysis and risk mitigation, leading to refined architectures.

10

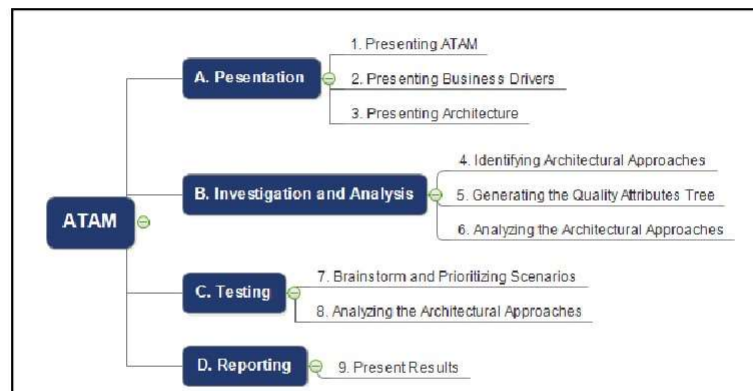
ARCHITECTURE TRADEOFF ANALYSIS METHOD (ATAM)

ATAM Steps:

ATAM Steps	Description
Present the ATAM	Evaluation team leader presents a method overview to the participants.
Present business drivers	Client or representative of system whose architecture is being evaluated presents the business drivers underlying the architecture.
Present architecture	Architect makes presentation.
Identify architectural approaches	Evaluation team catalogs architectural approaches used, as basis for subsequent analysis
Generate quality attribute utility tree	Participants build utility tree to identify quality attributes (and the scenarios that express them) of interest. Evaluation team facilitates.
Analyze architectural approaches	Evaluation team and architect perform analysis based on qualities desired and approaches used.
Brainstorm and prioritize scenarios	The architecture's stakeholders adopt an additional set of scenarios expressing architectural requirements. Evaluation team facilitates.
Analyze architectural approaches	Evaluation team and architect perform analysis based on these new scenarios.
Present results	Evaluation team leader makes presentation of analysis results, lists identified risks, sensitivity points, and tradeoffs in the architecture.

11

ARCHITECTURE TRADEOFF ANALYSIS METHOD (ATAM)



12

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.
- Based on scenarios
 - A scenario represents a description of a stakeholder's interaction with the system
- Scenarios are created depending on the point of view of each stakeholder:
 - **Developer** – interested in reusability, implementation, maintenance
 - **Project Manager** – interested in time, cost, quality, extensibility
 - **Tester** – interested in usability, mapping to requirements

13

SOFTWARE ARCHITECTURE ANALYSIS METHOD (SAAM)

• SAAM Steps:

Develop scenario	Involve all stakeholders Involve experts
Classify scenarios	Direct scenarios - can be "walked through" the architecture Indirect scenarios - require modification to the system
Perform scenario evaluations	A weighting of the difficulty of the change(s) Estimated cost of the changes A description of the set of changes required
Reveal scenario interaction	High interaction between semantically <i>unrelated</i> scenarios indicates: <ul style="list-style-type: none"> • Low cohesion • High structural complexity
Overall evaluation	A " <i>importance-weight</i> " should be assigned to each scenario and the scenario interactions. This is a subject process involving all of the stakeholders of the system. The weighting is used to determine an overall ranking of candidate architectures

14

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

15

ACTIVE REVIEWS FOR INTERMEDIATE DESIGNS (ARID)

- ARID Steps:

Phase 1: Pre-meeting	Step 1: Identify reviewers
	Step 2: Prepare design presentation
	Step 3: Prepare seed scenarios
	Step 4: Prepare for the review meeting
Phase 2: Review meeting	Step 5: Present ARID method
	Step 6: Present design
	Step 7: Brainstorm and prioritize scenarios
	Step 8: Perform review
	Step 9: Present conclusions

16

WHAT ARE THE BENEFITS AND COSTS?

- 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.

17

CONCLUSION

- The average architecture evaluation adds no more than a few days to the project schedule.
- 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.

18

```
If(anyQuestions)
{
    askNow();
}
else
{
    thankYou();
    submitAttendance();
    endClass();
}
```

14-Jun-2022

Engr. Majid Kaleem

19

REFERENCES

1. **Software Architecture**, *Perspectives on an Emerging Discipline* By Mary Shaw & David Garlan
2. **The Art of Software Architecture**, *Design Methods & Techniques* By Stephen T. Albin
3. **Essential Software Architecture**, By Ian Gorton
4. **Microsoft Application Architecture Guide**, By Microsoft
5. **Design Patterns**, *Elements of Reusable Object-Oriented Software* By Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides
6. **Refactoring, Improving the Design of Existing Code**, By Martin Fowler & Kent Beck

14-Jun-2022

Engr. Majid Kaleem

20