NTNU | Norwegian University of Science and Technology

TDT4240 Software Architecture

Chapter 21:
Evaluating an Architecture

Professor Alf Inge Wang

Learning Outcome

- Knowledge on evaluation of architectures.
- Know how to evaluate an architecture using the Architecture Tradeoff Analysis Methods (ATAM)

Who can Perform the Evaluation?

- Evaluation by the architect within the design process:
  - Evaluate every key decision
- Evaluation by peers within the design process:
  - Review architecture for all important quality attribute scenarios
- Analysis by outsiders once the architecture has been designed:
  - Objective eye on an architecture

Contextual factors for evaluation

- ***Which stakeholders*** will participate?
- ***Who performs*** the evaluation?
- ***What artifacts*** are available?
- What are the ***business goals***?
- ***Who sees*** the results?

Architecture Tradeoff Analysis Method (ATAM)

- Motivation:
  - Unsuitable architecture is a disaster
  - Uncover cross-project reuse
  - Force stakeholders to meet
  - Prioritize conflicting goals
  - Find problems early
  - Assurance

Architecture Tradeoff Analysis Method (ATAM)

- What is ATAM?
  - A method to evaluate if an architecture meets quality requirements by analyzing trade-offs.
    - Example: Does the architecture ensure 99.9% uptime?
- When to evaluate?
  - **Early:** Before full specification to assess decisions.
  - **Late:** Assess an existing system for integration with other systems.

Who are Involved?

- Evaluation team:
  - External group conducting the evaluation and analysis.
- Project decision makers:
  - Project manager, customer, architect.
- Architecture stakeholders:
  - Developers, testers, integrators, maintainers, performance engineers, users.

What Result Does ATAM Produce?

- Evaluation Report Answers:
  - Is the architecture suitable for its intended purpose?
  - Which competing architecture is the best fit?
- An architecture is suitable if:
  - The resulting system meets quality goals.
  - It can be built with available resources.

ATAM Outputs

- Decision mapping: Tactics linked to quality requirements
- Utility tree: Prioritized quality attributes as scenarios
- Identified sensitivity & tradeoff points
- Concise architecture presentation
- Defined business goals
- Risks & non-risks
- Risk themes

The ATAM process (Phases)

Phase	Activity	Participants
0	Partnership & preparation	Evaluation team leadership, Key project decision makers
1	Evaluation	Evaluation team, Project decision makers
2	Evaluation (continue)	Evaluation team, Project decision makers, Stakeholders
3	Follow-up	Evaluation team, Evaluation client

ATAM Steps (1-3)

- **Step 1: Present the ATAM** [Evaluation team]:
  - Steps, techniques, output
- **Step 2: Present the Business Goals** [Project decision makers]:
  - Most important functionality, constraints, business goals, major stakeholders, architectural significant requirements (ASRs)
- **Step 3: Present the Architecture** [Architect]:
  - Architecture (views), technical constraints, interaction with other systems, architectural approaches

Step 4: Identify the Architectural Approaches

- **Responsible: Evaluation team**
- Capture tactics & patterns:
  - Document approaches from Step 3
  - Ask architect to name approaches
- Architectural approaches:
  - Define important system structures, and how the system can respond to quality attributes
- Architectural patterns:
  - Elements, layout, relations, constraints

Step 5: Generate a Quality Attribute Utility Tree

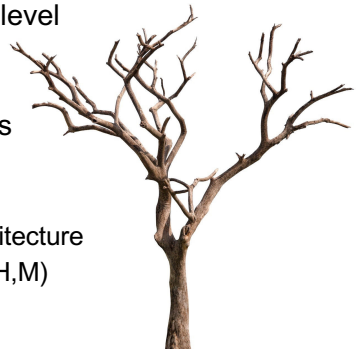
- **Responsible: Evaluation team, Project decision makers**
- **Goal:** Identify, prioritize, & refine key quality attribute goals
- **Output:** Prioritized quality attribute scenarios table
- **Process:** Translate business drivers into measurable quality requirements

- Utility: System's usefulness or goodness

Step 5: Generate QA Utility Tree (2)

Utility is the root node

- **Quality attribute** is the second level (e.g., Performance)
  - **Quality Attribute sub-domain** is the third level (e.g., data latency, transaction throughput)
    - **Quality Attribute Scenarios** is the fourth level prioritized in two dimensions (scale: Low, Medium, High):
      - Business value
      - Technical difficulty to achieve with architecture
      - E.g., 20,000 transactions per second (H,M)



Step 5: Generate the QA Utility Tree (3)

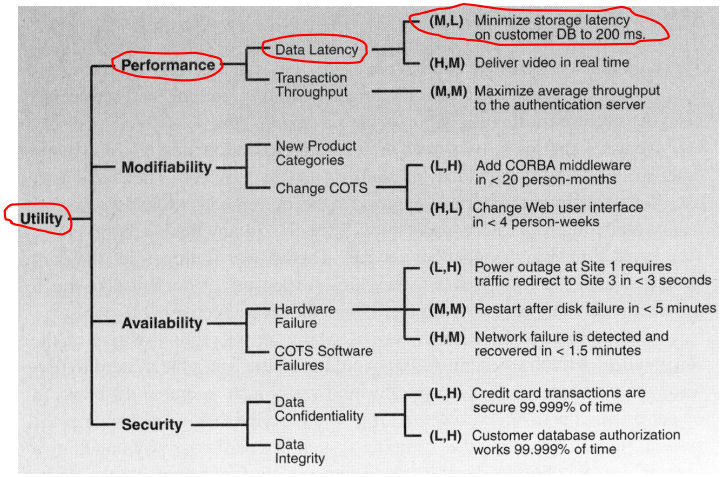
- **Use case** scenarios:
  - Specific stakeholder desires
- **Growth** scenarios:
  - Expected system changes
- **Exploratory** scenarios:
  - Stress test to reveal limits
  - Goal: Identify design boundaries, as systems rarely handle all modifications

- Each scenario highlights different architectural aspects

Step 5: Generate the QA Utility Tree (4)

- E-commerce example:
  - Business drivers:
    - **Security:** Protect customer data
    - **Modifiability:** Adapt to a fast-changing market
  - Key Considerations:
    - Business drivers must be concrete
    - Evaluation team must balance these with other goals (e.g., performance)
    - Utility tree aids in prioritization

Step 5: Generate the QA Utility Tree (4)



Step 6: Analyze Architectural Approaches

- **Responsible: Evaluation team**
- Analyze top-ranked scenarios individually:
  - Evaluate architectural approaches (tactics, patterns) that support the quality attribute scenario
  - Document key architectural decisions
  - Identify risks, non-risks, sensitivity points & tradeoffs:
    - Architect justifies architectural decisions for scenarios
    - May require deeper analysis.

Step 6: Analysis outputs

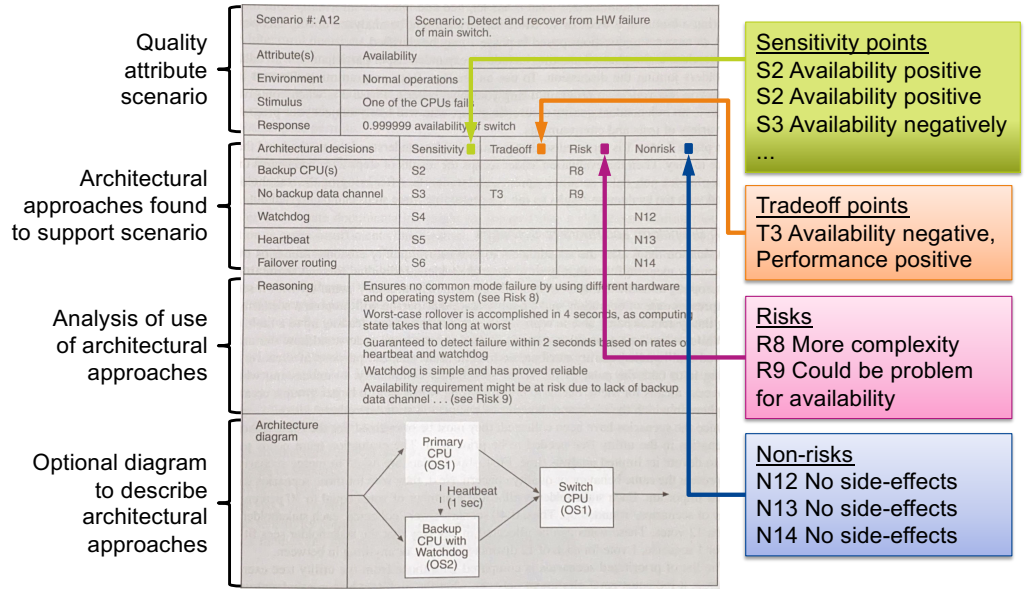
- ***Mapping*** architectural decisions or approaches to high-priority quality requirements.
- Analysis ***questions*** for each approach.
- Architect's ***responses*** to these questions.

Step 6: Analysis outputs continue...

- **Risk:** Architectural decisions that introduce project risks.
- **Non-risk:** Safe decisions that don't introduce risks.
- **Sensitivity points:** Architectural decisions that impact one quality attribute, either positively or negatively.
  - E.g., ping-echo will have a positive effect on availability.
- **Tradeoff points:** Architectural decisions that impact quality attributes both positively and negatively.
  - E.g., redundancy will have positive effect on availability and performance, but negative effect on security.

Step 6: Analyze architectural approaches

- Questions from evaluation team should help to:
  - Understand the approach and its application.
  - Identify known weaknesses.
  - Assess sensitivity and tradeoff points.
  - Explore interactions and tradeoffs with other approaches.



- **Step 7: Brainstorm and Prioritize Scenarios:**
  - Invite all stakeholders to propose additional scenarios to cover any gaps.
- **Step 8: Analyze the Architectural Approaches:**
  - Reanalyze architectural approaches using the new scenarios from Step 7.

ATAM Step 9 Present/Document Results

- Report/Presentation should include:
  - Architectural approaches (tactics, patterns, decisions) linked to quality requirements.
  - The utility tree.
  - Discovered risks.
  - Discovered non-risks.
  - Identified risk themes.
  - Sensitivity and trade-off points.
- This marks the delivery of this phase of the Architecture Project, from the Evaluation team's perspective.

Mini-approach for the ATAM project phase

1. Both groups (A and B) create their own utility tree of their quality attribute scenarios (from the Requirement document):
   - Group and prioritize scenarios in a table according to quality attribute
2. Group A presents its architecture to Group B
   - Group B take notes and ask questions about architecture, patterns, tactics
3. Choose the **two highest prioritized scenarios** from Group A and analyze them according to architecture
   - Group B fills out the analysis form, list sensitivity and tradeoff points, list risks and non-risks
   - Groups switch roles, and repeat Steps 2 and 3.
4. Write the ATAM report according to template:
   - Introduction, Utility tree, Analysis of architecture, Sensitivity points, Tradeoff points, Risks and Non-risks, Own experiences, Problems & Issues, Change log

Who is evaluating who?

Matching	Matching	Matching
Group 1 and 18	Group 2 and 17	Group 3 and 16
Group 4 and 15	Group 5 and 14	Group 6 and 13
Group 7 and 12	Group 8 and 11	Group 9 and 10

ATAM problem exam august 2010

- Task:
  - Do ATAM step 6 on software for garage controller for sensors, motors, panels and buttons to operate garage door.
- Utility tree:
  - Modifiability M1: Sensor/activators from various vendors (H,M)
  - Usability U1: Learn all features within 15 min (H,M)
  - Security S1: Less than 0.01% chance of unauthorized access (L,M)
- Identified architectural tactics:
  - AT1: Increase computational efficiency in critical components
  - AT2: Schedule time-critical components wisely
  - AT3: Structure system to have semantic coherence
  - AT4: Use information hiding

In groups discuss how to do an analysis of how the tactics fulfill the quality requirements

Solution to ATAM Problem

- Utility tree:
  - Modifiability M1: Sensor/activators from various vendors (H,M)
  - Usability U1: Learn all features within 15 min (H,M)
  - Security S1: Less than 0.01% chance of unauthorized access (L,M)
- Identified architectural tactics:
  - AT1: Increase computational efficiency in critical components
  - AT2: Schedule time-critical components wisely
  - AT3: Structure system to have semantic coherence
  - AT4: Use information hiding

AT1 & AT2 are performance - not relevant
U1 & S1 has no architectural tactics support

Solution to ATAM problem

- Scenario: M1 Possible to use sensors and activators from various vendors.
- Attribute(s): Modifiability
- Environment: Design time
- Stimulus: Change sensors and activators
- Response: Change only one component without side-effects
- Architectural Decisions:

	Sensitivity	Tradeoff	Risk	Nonrisk
AT3 Semantic coherence		T1		N1
AT4 Information hiding	S1			N1

Solution to ATAM problem

- Reasoning:
  - AT3 ensure that code managing sensors and activators are in the same place
  - AT4 improve and simplify relationship between different parts of system
  - The architecture lacks tactics such as virtual machines to make it easier to interface various hardware components.
- Architecture diagram: None
- Sensitivity points, Tradeoff points, Risk and Nonrisk:
  - T1 AT3 is positive for modifiability, but reduce performance
  - S1 AT4 is positive for modifiability without side-effects
  - N1 Both tactics are considered safe without any problems
- Risk themes: Architecture does not have sufficient support for various hardware.

ATAM Short Version

BONUS

- Assign these roles to the evaluation team:
  - **Team and evaluation leader**: Manage contact and process.
  - **Scribe**: Responsible for taking notes.
  - **Process responsible:** Time-keeper, process observer, process enforcer and questionnaire.
- Architecture stakeholders and project decision makers:
  - Role for presenting business goals.
  - Role for presenting the architecture.

ATAM Short Version (2)

BONUS

- Step 0: Exchange reports and read through the architectural description for the other group (10 min)
- Step 1: Summary of the ATAM (5 min)
  - Make a short summary of steps and outputs.
- Step 2: Present Business drivers (5 min)
  - Main functionality, technical constraints, business goals, and architectural drivers.
- Step 3: Present the Architecture (10 min)
  - Architectural tactics, architectural patterns, architectural views, constraints.

ATAM Short Version (3)

BONUS

- Step 4: Identify Architectural Approaches (5 min)
  - Evaluation team identifies architectural patterns and tactics.
  - Also includes patterns and tactics from step 3.

ATAM Short Version (4)

BONUS

- Step 5: Generate the Quality Attribute Utility Tree (20 min)
  - Generate the utility tree for the most important quality attribute goals.
  - Prioritize scenarios in two dimensions: Importance and Difficulty (L,L), (M,L), (H,M) etc...

ATAM Short Version (5)

BONUS

- Step 6: Analyze the Architectural Approaches (40 min)
  - Examine two high-ranked scenarios one at a time:
    - Find architectural approaches/decisions to carry out scenario.
    - Identify and write down risks, non-risks, sensitivity points and trade-offs.

ATAM Short Version (6)

BONUS

- Step 9 Present/Document the results:
  - Architectural approaches (tactics, patterns, decisions) mapped to quality requirements.
  - The utility tree.
  - A set of risks discovered
  - A set of non-risks documented.
  - A set of risk themes.
  - The sensitivity and trade-off points found.
- This is also the delivery of this phase of the Architecture Project.
  - The report should be from the **Evaluation team** point of view.

Experiences from using ATAM in Norwegian companies

BONUS

- Discussions get quickly too detailed.
- The most important scenarios are often not the same as believed before ATAM.
- System stakeholders focuses often on attributes, not scenarios.
- The requirements must often be better specified.
- Performance requirements can be found by:
  - Looking at similar systems.
  - Usability studies.

Evaluation by the designer

BONUS

- Designer should evaluate every key decision.
- Evaluation based on “generate-and-test” approach.
- How much analysis?
  - *The importance of the decision*.
  - *The number of potential alternatives.*
  - *Good enough as opposed to perfect.*

Peer review

BONUS

- Architectural designs can be peer reviewed, just as code can.
- A peer review can be carried out at any point of the design process Steps:
  1. The reviewers determine a number of quality attribute scenarios to drive the review.
  2. The architect presents the portion of the architecture to be evaluated.
  3. For each scenario, the designer walks through the architecture and explains how the scenario is satisfied.
  4. Potential problems are captured.

Analysis by outsiders

- Outside evaluators can cast an objective eye on an architecture.
- “Outside” is relative; this may mean
  - outside the development project
  - outside the business unit where the project resides but within the same company
  - outside the company altogether.
- Outsiders are chosen due to specialized knowledge or (long) experience
- Managers listen to problems uncovered by an outside team.
- An outside team used to evaluate complete architectures.