UNIVERSITÄT DES SAARLANDES

30 YEARS DFKI Human Centric AI.

Architecture Trade-off Analysis Method

ATAM

Architectural Thinking for Intelligent Systems

Winter 2019/2020

Prof. Dr. habil.Jana Koehler

Agenda

- Evaluation of architectures with the Architecture Trade-off Analysis Method (ATAM)
- Scenarios and Utility Tree
- Risks & Non-Risks
- Sensitivity Points and Trade-offs
- Applying ATAM and ATAM-light

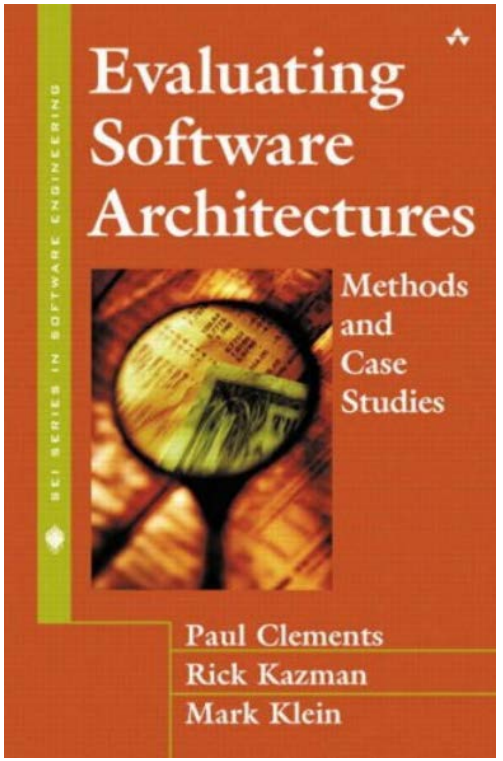
Tutorial Assignment 11:

- Our architecture has been developed to such a level of detail that we can think about a first evaluation based on the ATAM method.

Architecture Trade-off Analysis Method (ATAM)

"There is no such thing as an inherently good or bad architecture. Architectures are either more or less fit for some purpose."

Bass, Clements, Katzman



How Good is my Architecture?

- Architectural decisions have been taken to address requirements
- Constraints are satisfied
- Styles, patterns, tactics were applied
- The most risky software components are prototyped
- Patterns etc. are recommended practices, but need to be adapted to the system under consideration
- Experience helps to reduce risks
- Working with proven practices and solutions reduces the risk of failing to achieve the desired system qualitites

➢ How well does an architectural draft meet qualities?

Evaluation of Architectures

- Is the architecture good **enough?**

- Software architecture can only be evaluated qualitatively
  - No useful quantities **directly** measurable

- Quantitative measurement possible for
  - Scope of change and frequency of requirement changes
  - Prototyping and testing coverage
  - Bug fixes during implementation, backlog
  - Code metrics

➢ Do not provide an answer to our question!

➢ But we have **response measures** from scenarios!

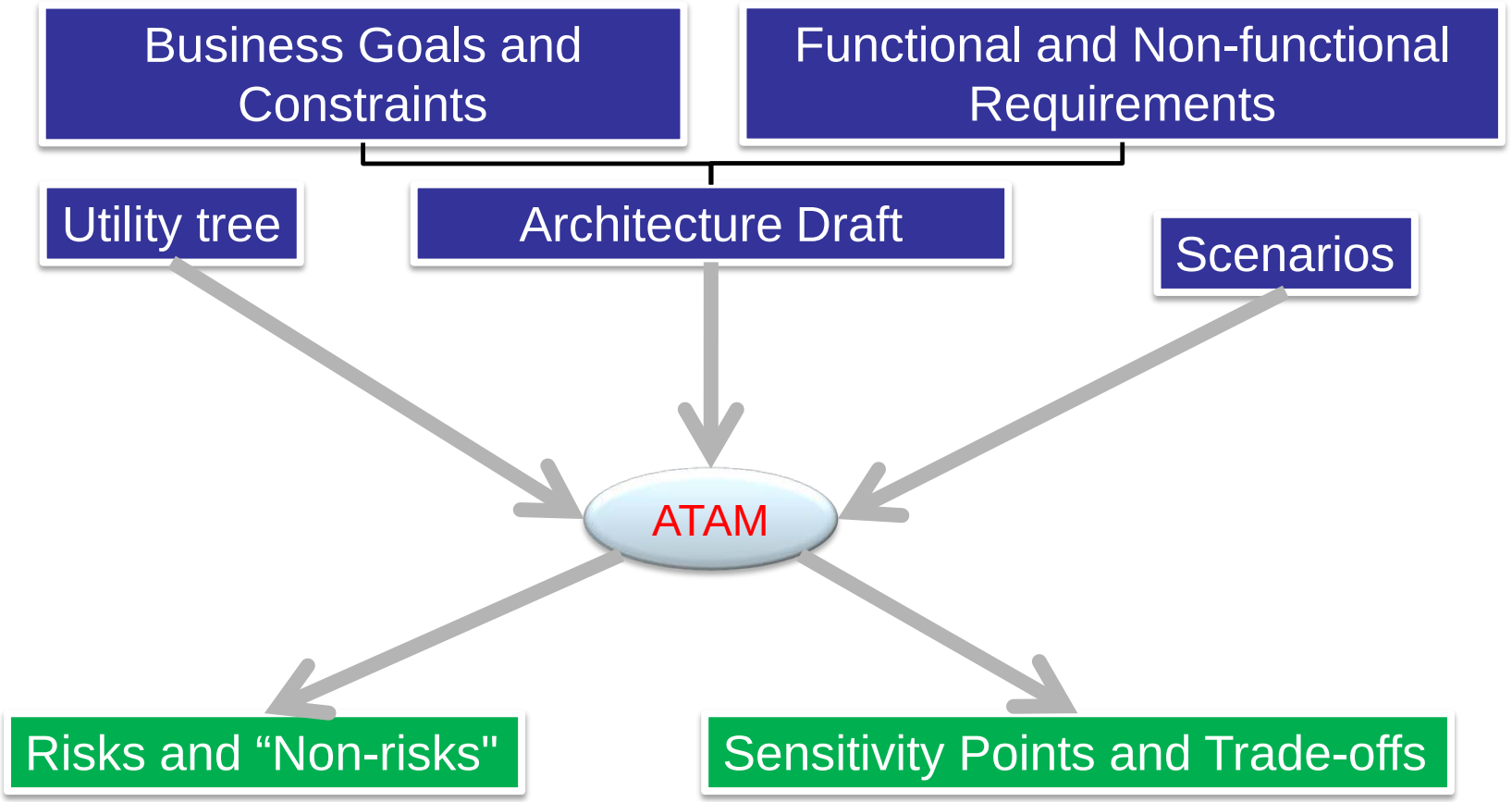
Analysis as a Prerequisite for Evaluation

- Whenever a key decision is made or a milestone is reached, the choices made and possible alternatives should be carefully documented and analyzed
  - Importance of the decision
  - Possible alternatives
  - Do we need to revise the decision?
  - "Good enough" compared to "Perfect"

- Analysis costs should not exceed costs of a wrong decision
  - Use thought experiments, simulations, prototypes

ATAM

- Evaluators do not need to know the architecture
- The system does not have to exist yet
- There can be many stakeholders
- 3 groups of participants
  1. Evaluators: outside the project, ATAM experts
  2. Project decision maker: architect, project manager, product owner
  3. Architecture stakeholders:
     - have an interest in a well working architecture
     - define the quality attributes they require from a good system (developers, testers, users)
     - 12-15 people for complex systems



Risk analysis as aim of ATAM:

Find a causal relation between architectural decisions and the to-be-expected system behavior

Results of ATAM

1. A precise and understandable representation of the architecture
   - can be presented in one hour
2. Definition of business goals and objectives
3. Prioritized list of quality attributes captured in scenarios
4. Risks
   - An architectural decision that has negative impact on a scenario represents a risk
   - Architectural risk mitigation plan

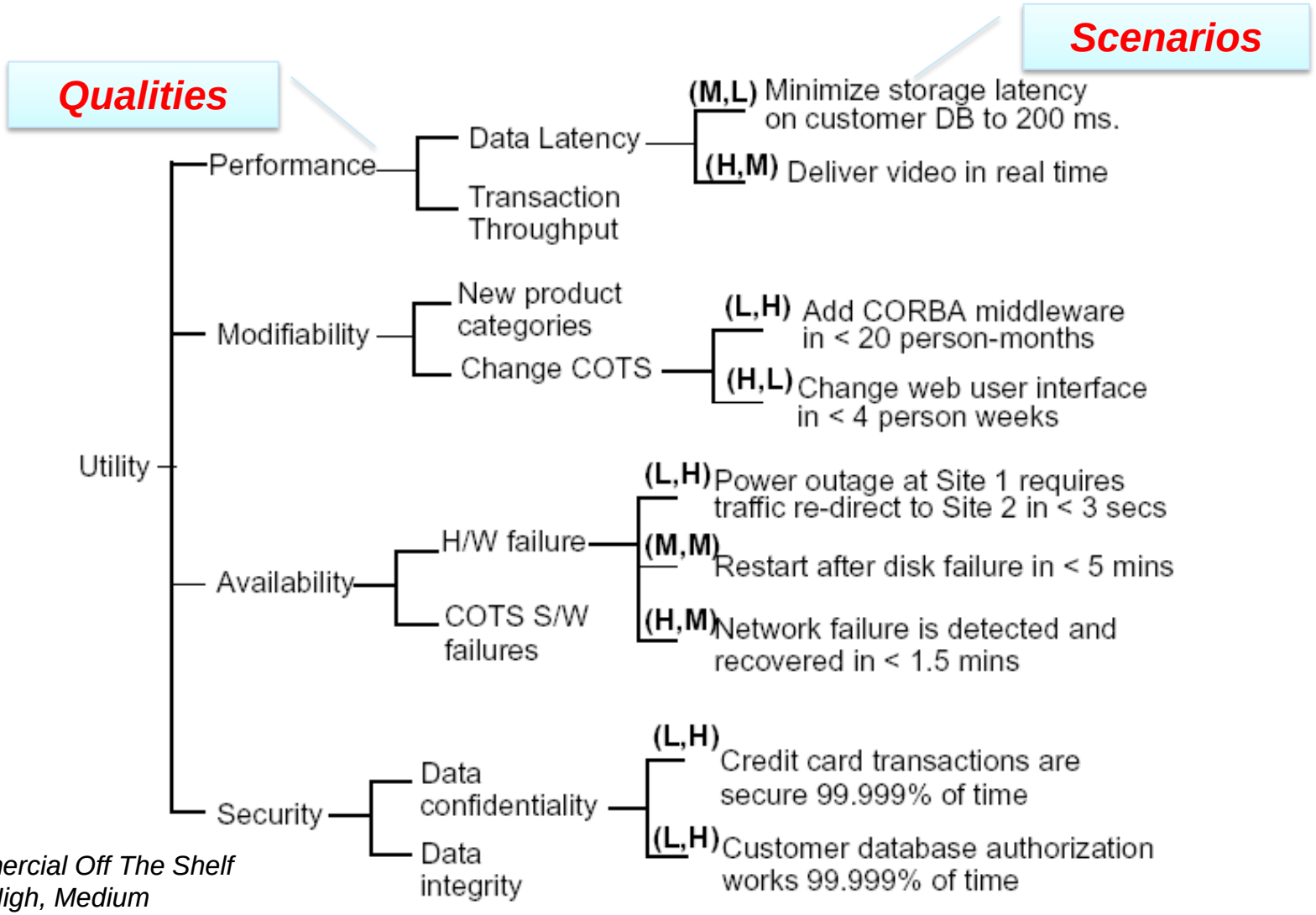
Results of ATAM (cont'd)

5. Risk Investigation: Which systematic weaknesses in the architecture (or in the team or process) lead to the risks?

6. Causal relationship between architectural decisions and quality attributes
   - Which decisions support which qualities?

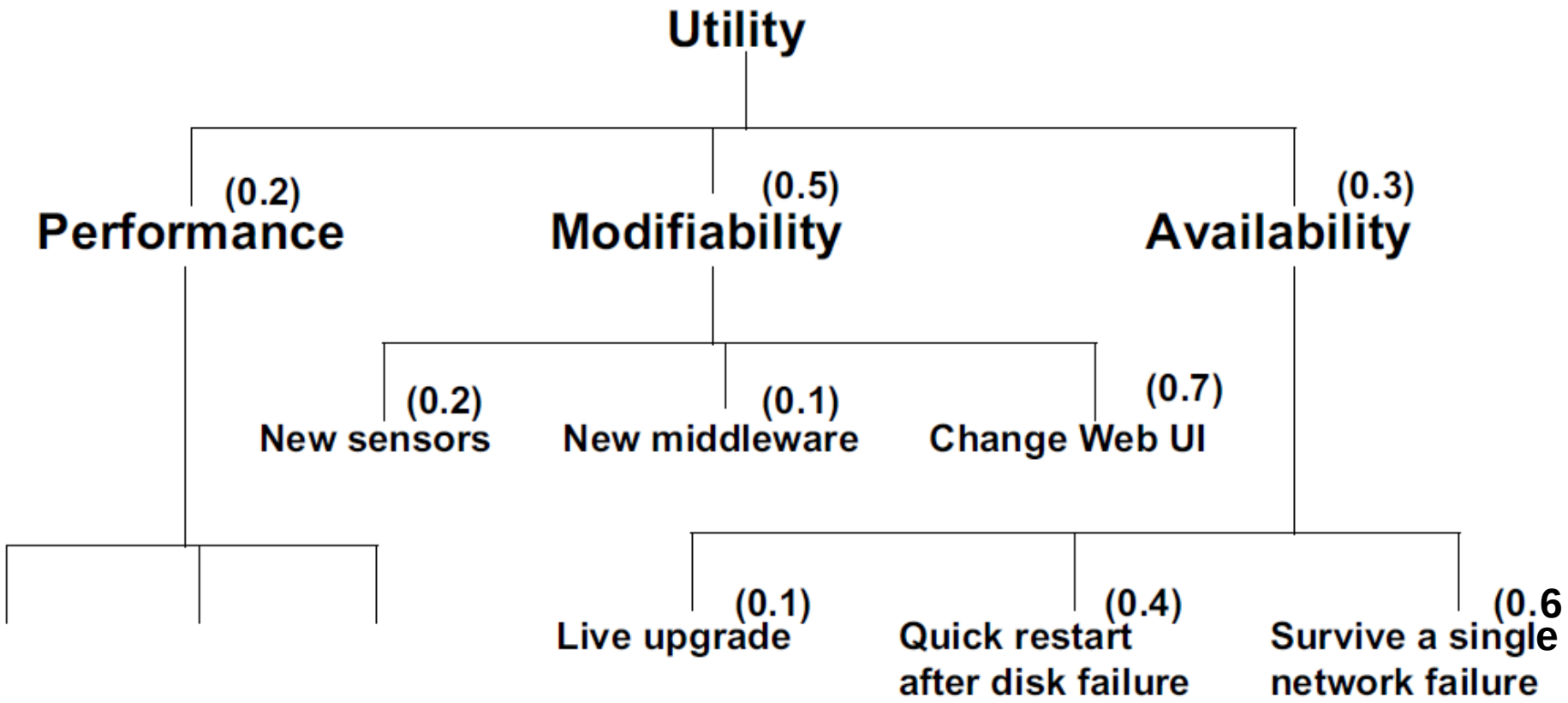
7. Sensitivity Points and Trade-offs
   - Architecture decisions that affect one or more quality attributes
   - Changing the decision impact system quality
   - Compromises had to found to balance competing requirements

ATAM Utility Tree – Attach Weights to System Qualities



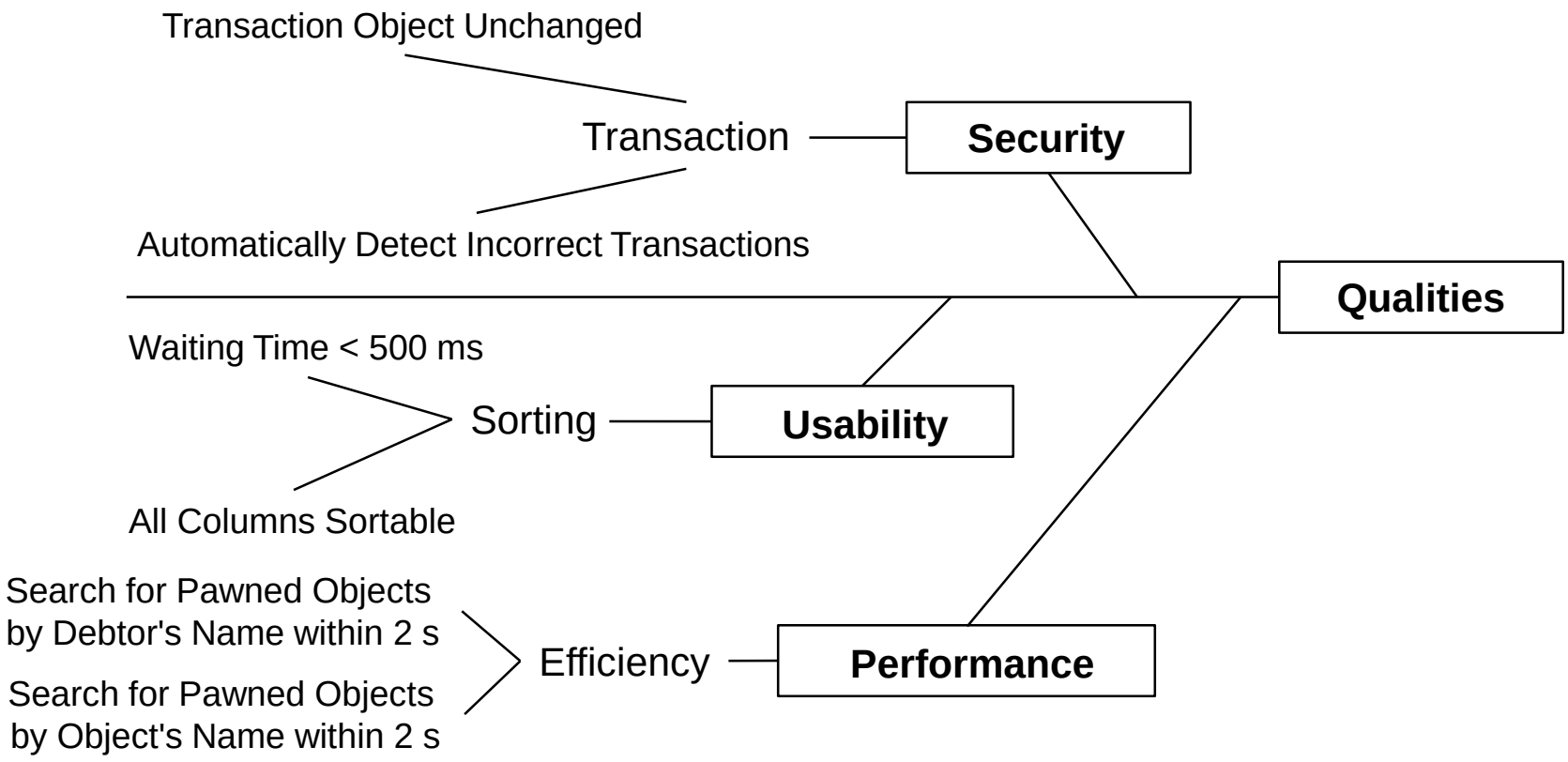
COTS - Commercial Off The Shelf
L,H,M - Low, High, Medium
Intervals: Probability of occurrence & Relevance/Risk (optional)

Alternative Representation of a Utility Tree



48

Example from a Student Project



Sytem Qualities and Stakeholders

- Performance
- Availability
- Usability
- Security

} View of the user

- Maintainability
- Portability
- Reusability
- Testability

} View of the developer

- Time To Market
- Cost and Benefits
- Projected lifetime
- Targeted Market
- Integration with Legacy System
- Roll back Schedule

} View of the Business

Scenarios

- Brief and precise description of a stakeholder's interaction with the system
  - From stimulus to measurable system response
- Important: involve all stakeholders
  - Users, Developers, Administrators, …
- Stakeholders evaluate and select the most important scenarios (approx. 10-15)

Typical Scenarios

- Availability of specific use case
  - *" Remote user accesses Report DB via Web and expects system response in less than 5 seconds."*
- Expected changes (growth scenarios)
  - *" Add another portal server within a week to reduce response time to less than 2.5 seconds."*
- Critical stress situations for the system
  - *" Half the servers go down without affecting system availability."*

Risks and Non-Risks

- Architectural decisions that are a risk (or no risk) for a given quality attribute
- If our assumptions change about a stimulus and the system response it triggers, we must review our decisions because non-risks (strengths) can become risks (weaknesses) and vice versa.
- Non-risks: "good" decisions supporting a quality attribute

- *"2-factor authentication by password and SMS token secures the system against unauthorized access.“*
- *"Simple user-selected passwords are a risk for system security."*

Sensitivity Points and Trade-offs

- Sensitivity Points
  - Architectural decisions with significant impact on a quality attribute when a minor change is made
  - Properties of critical components (or their connections) that are crucial to generating a particular response
  - *"Switching to a simplified authentication concept for users reduces system security."*

- Trade-offs
  - Architectural decisions that affect multiple quality attributes
  - *"Database backup ensures reliability, but compromises system performance."*

Example: Using a Shared File System

- Important common information of different components is stored in one file on a centrally accessible server
  - the file is small
  - no safety requirements
  - no simultaneous access of the components is possible

- As soon as one of these assumptions changes, we must revise our decision
  - the "flat file" can now become a risk

How does ATAM work?

1. Bring stakeholders together
2. Explain ATAM
3. Summarize business driver/context for system development
4. Present architecture - styles are important
5. Create Utility Tree
   a) Prioritize quality attributes and quantify response
   b) Define and prioritize scenarios
6. Compare scenarios and architecture
7. Identify risks, sensitivity points and trade-offs
8. Define actions to be taken (architecture, process, team)

Advantages of ATAM

- Clarification of non-functional requirements and quality attributes
- Improved documentation of architecture and architecture decisions
- Intensified discussion between stakeholders

- Early detection of risks at the beginning of the software life cylce
  - RISK: ATAM is time consuming!
- Desired result
  - Risk Mitigation - Introduction of measures to reduce risks
  - Improved architecture

Light-weight ATAM

- For smaller and less risky projects
- Smaller internal team

1. Explain ATAM
2. Summarize business driver/context for system development (15 min)
3. Present architecture (25 min)
4. Build UtilityTree (Utility Treee + scenarios exist, review in 1h)
   a) Hierarchize attributes and quantify meaning
   b) Define and prioritize scenarios
5. Compare scenarios and architecture (2-3h)
6. Identify risks, sensitivity points and compromises (in step 5)
7. Document results (0.5 h)

Summary

- ATAM uses Scenarios as a basis for evaluation
- Risk reduction in an early phase of the software life cycle as key focus
  - increased communication among stakeholders
  - clarified quality attribute requirements
  - improved architecture documentation
  - documented basis for architectural decisions
- ATAM does not give an absolute measure of quality, but identifies trade-offs, sensitivity points and risks
- Promotes thinking in decisions and decision alternatives
- Prototypes are important to validate system responses

Working Questions

1. When do you analyze and evaluate an architecture?
2. How can you analyze and evaluate architectures?
3. Explain the basic ideas of ATAM. What do you see as advantages/disadvantages of this method?
4. Why do we analyze and document how architectural decisions influencequality attributes?
5. What do we get as a result of ATAM?
6. What role do scenarios play in the ATAM method, in particular the Utility Tree?
7. What do we understand by risks, non-risks, sensitivity points and trade-offs in ATAM?

Working Questions

8. How can ATAM be used to compare two alternative architectural drafts with each other?
9. Your task is to validate the architecture of a system. The architect is not available and you are not allowed to talk to stakeholders. Only some documentation is available? What do you do?
10. When do you use ATAM full or lightweight?