

OVERVIEW **Architecture Evaluation and ATAM**

Architecture is shaped by functional requirements, quality requirements and business constraints, and it should be evaluated before construction, when problems are still cheap to fix. Evaluation methods (ATAM, SAAM, ARID) all rest on scenarios and quality attributes. ATAM, the main one, assesses how architectural decisions affect quality goals, with an evaluation team, decision makers and stakeholders working through nine steps: from presenting the method and business drivers, through a quality-attribute utility tree and scenario analysis, to a report of risks, non-risks, sensitivity points and trade-offs. A lightweight ATAM compresses this into a day or less for smaller projects.

Why evaluate

Prioritises conflicting goals, forces a clear explanation, improves documentation, uncovers reuse, improves practices.

Common ground

ATAM, SAAM and ARID all use scenarios and quality attributes.

ATAM's 9 steps

Present ATAM → business drivers → architecture → identify approaches → utility tree → analyse → brainstorm scenarios → analyse again → present results.

Findings

Risks, non-risks, sensitivity points (decisions that strongly affect one quality) and trade-offs (decisions that affect several qualities in opposite ways).

HOW THE LECTURE IS ORGANIZED

Slides 1–5 Why evaluate the architecture, and the benefits

Slides 6–11 Evaluation methods; ATAM purpose, quality attributes, participants, roles

Slides 12–15 ATAM steps, phases and output

Slides 16–28 The nine ATAM steps in detail, with utility trees and an analysis form

Slides 29–31 Lightweight ATAM and summary

EXAM FOCUS

Final 252 Q8 (5 marks with Chapter 9) asked three questions from this chapter: two benefits of architecture evaluation, the difference between ATAM and lightweight ATAM, and the shared point of all evaluation methods. Learn the nine ATAM steps in order as well.

HOW TO READ THIS PAGE

Every original slide is shown unchanged. **Yellow marks** on a slide point to a numbered note: hover or tap a mark to jump to its note, and hover a note to see its marks. Questions tagged **Past exam** appeared on a real paper; **Prediction** questions are likely angles, not confirmed.



Software Design and Architecture

[Architecture and Design Evaluation] – Chapter 08, L01

NOTES

- 1 Chapter 8 returns to the third key task of architectural design from Chapter 2: **evaluating and validating** the architecture before it is built.

SLIDE 2 / 31 Lecture Outlines

Lecture Outlines

- **Previously- Design Patterns**
- **Architecture Evaluation**
- **Benefits of Architecture Evaluation**
- **Methods of Architecture Evaluation**
- **ATAM Evaluation**

NOTES

- 1 Why architectures are evaluated.
- 2 The benefits of evaluation.
- 3 The main evaluation methods.
- 4 **ATAM** in detail, and its lightweight version.

SLIDE 3 / 31 Software Architecture Evaluation*

Software Architecture Evaluation*

- The **architecture** design is the **foundation** of any software system.
- It defines the structure of the system through its underlying components and their relationship as well as the properties and behavior that are exposed to the external world.
- The software architecture is basically shaped by the following drivers: **functional requirements, non-functional requirements** and **business constraints**.

Prince Sultan University | Slides prepared from "Software Engineering Design: Theory and Practice" book and slides

3

*Source: <https://johnolamendy.wordpress.com/2011/08/12/evaluate-the-software-architecture-using-atam/>

NOTES

- 1 The architecture is the **foundation** of a system: it defines the structure (components and relationships) and the properties and behaviour exposed to the outside world.
- 2 It is shaped by three **drivers**: **functional requirements, non-functional (quality) requirements** and **business constraints** (budget, schedule, regulations). An evaluation checks the architecture against all three.

Prediction

Q1

Which of these is NOT one of the three drivers that shape a software architecture?

- A. The programming language's syntax
- B. Functional requirements
- C. Non-functional requirements
- D. Business constraints

Answer: A. The programming language's syntax

SLIDE 4 / 31 Software Architecture Evaluation

Software Architecture Evaluation

- In order to run a successful project with a high quality software product, it is important to **validate the architectural design in advance**
- The earlier we find a problem in the design phase, the better (**less** cost to fix an error)
- Software architecture (*architectural decisions*) should be evaluated **before the construction phase begins**.

NOTES

- 1 For a successful project and a high-quality product, the architectural design must be **validated in advance**.
- 2 The earlier a problem is found, the **cheaper** it is to fix: a wrong decision found on paper costs a meeting; found in code, it costs a rewrite.
- 3 So architectural decisions should be evaluated **before construction begins**.

Benefits of Architecture Evaluation

- Results in prioritization of conflicting goals
- Forces clear explanation of architecture
- Improves quality of architecture documentation
- Uncovers opportunities for cross-project reuse
- Results in improved architecture practices

NOTES

- 1 It results in **prioritisation of conflicting goals**: stakeholders must agree which qualities matter most when they clash.
- 2 It forces a **clear explanation of the architecture**: the architect must be able to present and justify it.
- 3 It improves the **quality of the architecture documentation**, because gaps show up during the review.
- 4 It uncovers opportunities for **cross-project reuse** of components and approaches.
- 5 It leads to **improved architecture practices** in the organisation over time.

EXAM ANGLE

Past exam · Final 252 · Q8-1 · 1 mark

Q1

What are the benefits of architecture evaluation? State two only.

Answer: Any two: it forces **prioritisation** of conflicting goals; it forces a **clear explanation** of the architecture; it improves the quality of architecture **documentation**; it uncovers opportunities for cross-project **reuse**; it results in improved architecture **practices**. (Finding problems early, when they are cheap to fix, is also accepted.)

Major Architectural Evaluation Methods

- Today, we have several methods to evaluate the software architecture :
 - ✓ **ATAM**: Architecture Tradeoff Analysis Method
 - ✓ **SAAM**: Software Architecture Analysis Method
 - ✓ **ARID**: Active Reviews for Intermediate Designs
- These methods, have in common, the use of scenarios and quality attribute in the evaluation process

NOTES

- 1 **ATAM**: the Architecture Tradeoff Analysis Method (the main one in this chapter).
- 2 **SAAM** (Software Architecture Analysis Method) and **ARID** (Active Reviews for Intermediate Designs) are other established methods.
- 3 What they all have **in common**: they evaluate the architecture using **scenarios** and **quality attributes**. They differ in depth and focus, not in the underlying idea.

EXAM ANGLE

Past exam · Final 252 · Q8-3 · 1 mark

Q1

What is the shared, common point between all architecture evaluation methods?

Answer: All methods (ATAM, SAAM, ARID) are built on the same two ingredients: **scenarios** and **quality attributes**. They differ in depth and focus, not in the underlying idea.

ATAM Evaluation Method

- The **Architecture Tradeoff Analysis Method (ATAM)**, is one of the most used architectural decision method.
- The purpose of **ATAM** is to assess the consequences of architectural decisions in light of quality attribute requirements and business goals.
- This means discovering risks where a quality attribute of interest is affected by architectural decisions (a trade-off between the quality attributes).

NOTES

- 1 The **Architecture Tradeoff Analysis Method (ATAM)** is one of the most widely used evaluation methods.
- 2 Its purpose: **assess the consequences of architectural decisions** in light of the quality-attribute requirements and the business goals.
- 3 In practice that means **discovering risks**: places where an architectural decision affects a quality attribute of interest, often through a **trade-off** between qualities.

EXAM ANGLE

Prediction

Q1

What does ATAM stand for, and what is its purpose?

Answer: Architecture Tradeoff Analysis Method. It assesses the consequences of architectural decisions against the quality-attribute requirements and business goals, discovering risks and trade-offs between quality attributes.

ATAM Evaluation Method – Quality attributes Re-visited

- One important concept related to any architecture evaluation method is the **quality attribute**.
- In complex systems, quality attributes can never be achieved in isolation.
- In order to achieve one quality, other quality is affected (sometimes negative and sometimes positive).
- So ... the architectural decisions are really a **trade-off** between the quality attributes in order to support the business goals.

NOTES

- 1 The key concept in every evaluation method is the **quality attribute** (Chapter 3).
- 2 In complex systems qualities can **never be achieved in isolation**: improving one affects others, sometimes negatively (encryption adds security but costs performance), sometimes positively (modularity helps both modifiability and testability).
- 3 So architectural decisions are really **trade-offs** between qualities in support of the business goals, which is exactly what ATAM analyses.

ATAM Evaluation Method - Quality attributes Re-visited

- We can group the quality attributes into three main categories:
 - ✓ **End-user perspective**: performance, availability, usability and security
 - ✓ **Technical perspective**: modifiability, portability, reusability, testability, interoperability
 - ✓ **Business community perspective**: time to market, cost and benefits, projected life time, project budget
- In order to evaluate the software architecture using quality attributes, we need to characterize them in a proper way using **quality attribute scenario**. (see Chapter 3 slides for details about attribute scenarios)
- Then, a quality attribute scenario can be used to prioritize the quality attributes.

NOTES

- 1 **End-user** perspective: performance, availability, usability, security.
- 2 **Technical** perspective: modifiability, portability, reusability, testability, interoperability.
- 3 **Business community** perspective: time to market, cost and benefits, projected lifetime, project budget.
- 4 To evaluate an architecture against them, each must be characterised as a **quality attribute scenario** (Chapter 3); the scenarios are then used to prioritise the qualities.

ATAM Evaluation Method – Participants

- In order to execute the ATAM process correctly, we need three groups cooperating each other:
 - ✓ **Evaluation team:** A group of experienced architects (three to five people)
 - ✓ **Project decision makers:** People with the authority to make changes in the project (project manager, customer, manager)
 - ✓ **Architecture stakeholders:** People interested in a good architecture for doing correctly their job (developers, testers, integrators, maintainers, performance engineering and users)

NOTES

- 1 The **evaluation team**: three to five experienced architects who run the evaluation.
- 2 **Project decision makers**: people with authority to change the project (project manager, customer, manager).
- 3 **Architecture stakeholders**: people who need a good architecture to do their jobs (developers, testers, integrators, maintainers, performance engineers, users).

EXAM ANGLE

Prediction

Q1

Name the three groups that take part in an ATAM evaluation.

Answer: The evaluation team (3–5 experienced architects), the project decision makers (manager, customer), and the architecture stakeholders (developers, testers, integrators, maintainers, users).

ATAM Evaluation - Team Roles

Role	Responsibilities
Team leader	Sets up the evaluation; coordinates with client, making sure client's needs are met; establishes evaluation contract; forms evaluation team; sees that final report is produced and delivered (although the writing may be delegated)
Evaluation leader	Runs evaluation; facilitates elicitation of scenarios; administers scenario selection/prioritization process; facilitates evaluation of scenarios against architecture; facilitates on-site analysis
Scenario scribe	Writes scenarios on flipchart or whiteboard during scenario elicitation; captures agreed-on wording of each scenario, halting discussion until exact wording is captured
Proceedings scribe	Captures proceedings in electronic form on laptop or workstation: raw scenarios, issue(s) that motivate each scenario (often lost in the wording of the scenario itself), and resolution of each scenario when applied to architecture(s); also generates a printed list of adopted scenarios for handout to all participants
Questioner	Raises issues of architectural interest, usually related to the quality attributes in which he or she has expertise

NOTES

- 1 Team leader:** sets up the evaluation contract with the client, forms the team, and makes sure the final report is produced and delivered. (The garbled text on the slide is overlapping lines in the PDF.)
- 2 Evaluation leader:** runs the evaluation, facilitates scenario elicitation, runs scenario selection and prioritisation, and facilitates evaluating scenarios against the architecture.
- 3 Scenario scribe** writes scenarios on the board and captures their agreed wording, stopping discussion until it is exact; the **proceedings scribe** records everything electronically (scenarios, the issues behind them, how each was resolved) and prints the adopted scenarios.
- 4 Questioner:** raises issues of architectural interest, usually about the quality attributes they specialise in.

ATAM Evaluation Method – Steps

➤ The ATAM evaluation process consist of 9 steps:

1. Present the ATAM method
2. Present the business drivers
3. Present the architecture
4. Identify architectural approaches
5. Generate quality attribute utility tree
6. Analyze architectural approaches

NOTES

- 1 Steps **1–3** set the context: present ATAM, present the business drivers, present the architecture.
- 2 Steps **4–6** do the first analysis: identify architectural approaches, generate the quality-attribute **utility tree**, analyse the approaches against its top scenarios.

EXAM ANGLE

Prediction

Q1

List the nine steps of ATAM in order.

Answer: 1 Present the ATAM method. 2 Present the business drivers. 3 Present the architecture. 4 Identify architectural approaches. 5 Generate the quality attribute utility tree. 6 Analyse architectural approaches. 7 Brainstorm and prioritise scenarios. 8 Analyse architectural approaches (again, with the new scenarios). 9 Present results.

Prediction

Q2

In which ATAM step is the quality attribute utility tree generated?

- A. Step 5
- B. Step 2
- C. Step 7
- D. Step 9

Answer: A. Step 5

ATAM Evaluation Method – Steps (cont.)

➤ The rest of the ATAM evaluation process steps:

7. Brainstorm and prioritize scenarios
8. Analyze architectural approaches
9. Present results

NOTES

- 1 Steps 7–8 widen the view: all stakeholders **brainstorm and prioritise** scenarios, and the approaches are **analysed again** against them.
- 2 Step 9: **present the results.**

SLIDE 14 / 31 ATAM Evaluation – Phases

ATAM Evaluation – Phases

Phase	Activity	Participants	Typical duration
0	Partnership and preparation: Logistics, planning, stakeholder recruitment, team formation	Evaluation team leadership and key project decision-makers	Proceeds informally as required, perhaps over a few weeks
1	Evaluation: Steps 1-6	Evaluation team and project decision-makers	1-2 days followed by a hiatus of 2-3 weeks
2	Evaluation: Steps 7-9	Evaluation team, project decision makers, stakeholders	2 days
3	Follow-up: Report generation and delivery, process improvement	Evaluation team and evaluation client	1 week

NOTES

- 1 **Phase 0**, partnership and preparation: logistics, planning, recruiting stakeholders, forming the team. Evaluation-team leadership and key decision makers; informal, perhaps over a few weeks.
- 2 **Phase 1**, evaluation steps 1–6: the evaluation team and project decision makers; 1–2 days, followed by a 2–3 week break.
- 3 **Phase 2**, evaluation steps 7–9: the evaluation team, decision makers **and stakeholders**; 2 days.
- 4 **Phase 3**, follow-up: report generation and delivery, process improvement; evaluation team and client; 1 week.

SLIDE 15 / 31 ATAM Evaluation Method Output

ATAM Evaluation Method – Output

- The **output** of the ATAM method must include at least the following artifacts:
 - ✓ The **documentation of the selected architecture**: the key artifact to specify the architecture is the **SAD (Software Architecture Document)**
 - ✓ An **evaluation report** that:
 - Recaps the ATAM process
 - Captures the scenario analysis (quality requirements captured in the form of scenarios).
 - Explains the candidate architectures and the underlying rationale in the architectural decision process to select the right architecture and a summary of all the work done.

NOTES

- 1 ATAM must produce at least the **documentation of the selected architecture**, whose key artefact is the **SAD (Software Architecture Document)** (Chapter 9).
- 2 And an **evaluation report** that recaps the ATAM process, captures the scenario analysis (quality requirements written as scenarios), and explains the candidate architectures and the rationale for choosing one, with a summary of the work done.

ATAM Steps .. In more details ...

NOTES

- 1 Section divider: slides 17–28 go through the nine steps one by one.

Step 1. Present the ATAM method

- The evaluation team presents an overview of the ATAM process such as:
 - ✓ The **key steps**, techniques (e.g. utility tree generation, architecture elicitation and analysis, scenario brainstorming).
 - ✓ The expected **output** (architectural approaches, utility tree, scenarios, risks, sensitivity points).

NOTES

- 1 **Step 1:** the evaluation team presents an overview of ATAM: its **key steps** and techniques (utility-tree generation, architecture elicitation and analysis, scenario brainstorming)...
- 2 ...and the expected **output**: architectural approaches, the utility tree, scenarios, risks and sensitivity points.

SLIDE 18 / 31 Step 2. Present the business drivers

Step 2. Present the business drivers

- The **project decision maker** presents the system from the business point of view including the following information:
 - ✓ Business **goals** and context
 - ✓ Major **stakeholders**
 - ✓ High-level **functional requirements** (described as use cases or user stories) that impact on the system architecture
 - ✓ Most important **quality attributes** (described as quality scenarios) that impact on the system architecture
 - ✓ **Constraints** such as technical, managerial, economic and political

NOTES

- 1 **Step 2:** the **project decision maker** presents the system from the business point of view.
- 2 That covers the business **goals** and context, and the major **stakeholders**.
- 3 The high-level **functional requirements** (use cases, user stories) and the most important **quality attributes** (as quality scenarios) that affect the architecture.
- 4 And the **constraints**: technical, managerial, economic and political.

Step 3. Present architecture

- The **lead architect** makes a presentation describing the **architectural** approach used to meet the requirements and the constraints.
- In order to describe the architecture, it's very useful to use the **4+1** architectural view model (see Chapter 02, L03)
- This view model describes the **contextual view** (e.g. relationship with humans and other systems), the logical view (module, layers, relationship), **process view** (e.g. process, threads, data flow, events) and **deployment view** (CPU, storage, devices, network).
- And finally, describe **risks** associated in order to meet the architectural requirements.

NOTES

- 1 **Step 3:** the **lead architect** presents the architectural approach used to meet the requirements and constraints.
- 2 The **4+1 view model** (Chapter 2, Lecture 3) is a good way to structure the presentation.
- 3 It covers the **contextual** view (relationships with people and other systems), the logical view (modules, layers), the process view (processes, threads, data flow, events) and the deployment view (CPUs, storage, devices, network).
- 4 Finally, the architect describes the **risks** in meeting the architectural requirements.

Step 4. Identify architectural approaches

- The evaluation team tries to identify what key architectural approaches (**patterns**) are used for realizing the requirements and constraints.
- Possible architectural approaches like: **client-server**, **multi-layer** application, pipes-filters, etc.
- In this step, the evaluation team **analyzes** deeply the **architecture** presented in the step 3, and then it has a good idea of what patterns and approaches the architect used at designing the system.

NOTES

- 1 **Step 4:** the evaluation team identifies the key **architectural approaches (patterns)** used to realise the requirements and constraints.
- 2 For example client-server, multi-layer, pipes and filters (Chapter 4).
- 3 The team **analyses** the architecture from step 3 in depth until it knows which patterns and approaches the architect used. These approaches are what the later analysis tests.

Step 5. Generate quality attribute utility tree

- In this step, the evaluation team (along with the project decision maker) **identifies, prioritizes and refines** the most **important** quality attribute goals (expressed by quality scenario) by building a **utility tree**.
- A **utility tree** is a **top-down approach** for **characterizing** the quality attribute requirements, selecting the most important quality goals to be the high-level nodes (*performance, security, availability, modifiability, maintainability*) and the leaves of the tree are quality scenarios evaluated by importance (success of the system) and **difficulty** (architect's assessment).

NOTES

- 1 **Step 5:** the evaluation team, with the decision maker, **identifies, prioritises and refines** the most important quality-attribute goals, expressed as scenarios, by building a utility tree.
- 2 A **utility tree** is a **top-down** way of characterising the quality requirements: the root is "utility", the next level the most important qualities (performance, security, availability, modifiability, maintainability), and the **leaves** are concrete quality scenarios.
- 3 Each leaf is rated for **importance** to the system's success and **difficulty** to achieve (the architect's assessment), usually H/M/L. High-importance, high-difficulty (H,H) scenarios are analysed first.

EXAM ANGLE

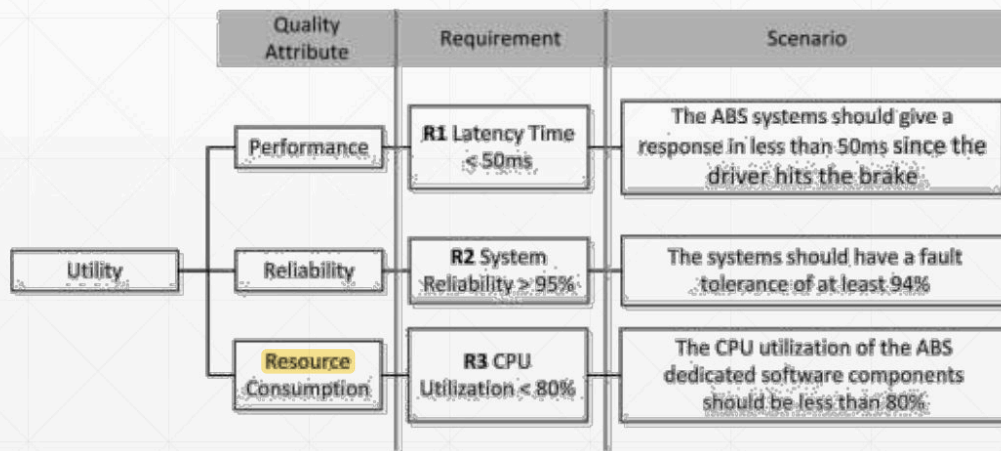
Prediction

Q1

What is a quality attribute utility tree, and how are its leaves rated?

Answer: A top-down tree: utility at the root, important quality attributes as the next level (refined into sub-concerns), and concrete quality scenarios as leaves. Each leaf is rated (H/M/L) for its importance to the system's success and its difficulty to achieve, so the most important and difficult scenarios are analysed first.

Step 5. Generate quality attribute utility tree (cont.)



Utility tree example

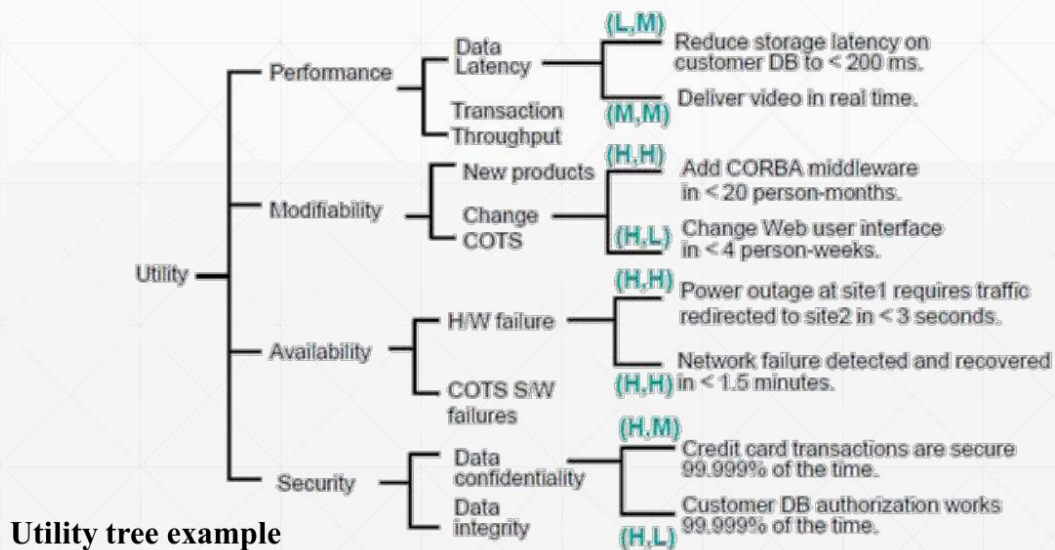
Prince Sultan University | Slides prepared from "Software Engineering Design: Theory and Practice" book and slides

22

NOTES

- 1 A utility tree for a car's anti-lock braking system (ABS), with three columns: quality attribute → requirement → scenario.
- 2 **Performance** → R1 latency < 50 ms → "the ABS should respond in less than 50 ms after the driver hits the brake".
- 3 **Reliability** → R2 system reliability > 95% → "fault tolerance of at least 94%" (note the slide's numbers don't quite match). **Resource consumption** → R3 CPU utilisation < 80% → "CPU use of the ABS software below 80%".

Step 5. Generate quality attribute utility tree (cont.)



NOTES

- 1 The textbook utility tree with ratings (importance, difficulty). **Performance:** data latency → reduce storage latency on the customer DB to < 200 ms (L,M); transaction throughput → deliver video in real time (M,M).
- 2 **Modifiability:** new products → add CORBA middleware in < 20 person-months (H,H); change COTS → change the web UI in < 4 person-weeks (H,L).
- 3 **Availability:** hardware failure → power outage at site 1 redirects traffic to site 2 in < 3 s (H,H); COTS software failure → network failure detected and recovered in < 1.5 min (H,H).
- 4 **Security:** data confidentiality → credit-card transactions secure 99.999% of the time (H,M); data integrity → customer DB authorisation works 99.999% of the time (H,L). The (H,H) leaves go to step 6 first.

Step 6 – Analyze Architectural Approaches

- In this step, the evaluation team examines the **highest-ranked** scenarios one at a time in order to understand how the proposed architecture supports each one.
- The team also identify and document the **architecture decisions** and its rationale, **risks, non-risks, sensitivity points and tradeoffs** with regards to quality attributes.
- This information can be recorded using the **form shown in the next slide**, where the analysis of the architecture approach for a scenario is captured.

NOTES

- 1 **Step 6:** the evaluation team examines the **highest-ranked** scenarios one at a time to understand how the architecture supports each one.
- 2 For each, it documents the relevant **architecture decisions** and their rationale, and classifies the findings as **risks** (decisions that may prevent a quality goal), **non-risks** (decisions judged safe), **sensitivity points** (decisions that strongly affect one quality's response) and **trade-offs** (decisions that affect several qualities in opposite directions).
- 3 The findings are recorded on an analysis **form** (next slide).

Prediction

Q1

In ATAM, what is the difference between a sensitivity point and a trade-off point?

Answer: A sensitivity point is an architectural decision that strongly affects the response of one quality attribute. A trade-off point is a decision that affects several quality attributes, improving one while degrading another (e.g. encryption: better security, worse performance).

Prediction

Q2

A decision that improves one quality attribute while degrading another is recorded in ATAM as a:

- A. Trade-off point
- B. Non-risk
- C. Business driver
- D. Utility tree leaf

Answer: A. Trade-off point

SLIDE 25 / 31 Step 6 – Analyze Architectural Approaches: the analysis form

Step 6 – Analyze Architectural Approaches (cont.)

Scenario #: A12		Scenario: Detect and recover from HW failure of main switch.			
Attribute(s)	Availability				
Environment	Normal operations				
Stimulus	One of the CPUs fails				
Response	0.999999 availability of switch				
Architectural decisions		Sensitivity	Tradeoff	Risk	Nonrisk
Backup CPU(s)		S2		R8	
No backup data channel		S3	T3	R9	
Watchdog		S4			N12
Heartbeat		S5			N13
Failover routing		S6			N14
Reasoning		Ensures no common mode failure by using different hardware and operating system (see Risk 8) Worst-case rollover is accomplished in 4 seconds as computing state takes that long at worst Guaranteed to detect failure within 2 seconds based on rates of heartbeat and watchdog Watchdog is simple and has proved reliable Availability requirement might be at risk due to lack of backup data channel ... (see Risk 9)			
Architecture diagram		<pre>graph LR P[Primar CPU OS1] -- heartbeat 1 sec --> B[Backup CPU with Watchdog OS2] P --> S[Switch CPU OS1] B --> S S --> Out[]</pre>			

NOTES

- 1 The analysis form for one scenario. **Scenario A12:** detect and recover from a hardware failure of the main switch. Attribute: availability; environment: normal operations; stimulus: one of the CPUs fails; response: 0.999999 availability of the switch.
- 2 **Architectural decisions** with their classification: backup CPU(s) (sensitivity S2, risk R8); no backup data channel (S3, trade-off T3, risk R9); watchdog (S4, non-risk N12); heartbeat (S5, N13); failover routing (S6, N14).

- 3 **Reasoning:** different hardware and OS avoid common-mode failure (see risk 8); worst-case rollover is 4 seconds; failure is detected within 2 seconds from the heartbeat and watchdog rates; the watchdog is simple and proven; availability might be at risk without a backup data channel (risk 9). The **architecture diagram** shows primary and backup CPUs with heartbeat (1 s) to the switch CPU.

SLIDE 26 / 31 **Step 7 Brainstorm and prioritize scenarios**

Step 7 – Brainstorm and prioritize scenarios

- In this step, the **stakeholders brainstorm scenarios** that are operationally meaningful with respect to the stakeholders' individual roles.
- The purpose of scenario brainstorming is to **take the pulse of the larger stakeholder community**: to understand what system success means for them.
- Once the scenarios have been collected, they are **prioritized by voting**.
- The list of prioritized scenarios is compared with those from the utility tree exercise:
 - ✓ If they agree, it indicates good alignment between what the architect had in mind and what the stakeholders actually wanted.
 - ✓ If additional driving scenarios are discovered—and they usually are—**this may itself be a risk**, if the conflict is large. This would indicate that there was some disagreement in the system's important goals between the stakeholders and the architect.

NOTES

- 1 **Step 7:** the wider group of **stakeholders brainstorms scenarios** that matter for their own roles.
- 2 The purpose is to **take the pulse** of the larger stakeholder community: what does success mean for them?
- 3 The scenarios are **prioritised by voting**, then compared with the utility tree from step 5. If they agree, the architect and stakeholders are aligned.
- 4 If new driving scenarios appear (they usually do), that **may itself be a risk** when the gap is large: the architect and stakeholders disagree about the system's important goals.

Step 8 – Analyze Architectural Approaches

- In this step, the evaluation team guides the architect in the process of carrying out the **highest ranked scenarios from step 7**.
- The architect identifies how the architectural approaches are **impacted** by the scenarios generated in the previous step.
- Risks, non-risks, sensitivity points and tradeoffs **continue to be** identified and architectural decisions are specified.

NOTES

- 1 **Step 8:** the evaluation team guides the architect through the **highest-ranked scenarios from step 7**.
- 2 The architect explains how the architectural approaches are **affected** by these new scenarios.
- 3 Risks, non-risks, sensitivity points and trade-offs **continue to be identified**, and architectural decisions are specified. It is step 6 repeated with the stakeholders' scenarios.

Step 9 – Present Results

- And finally, the information generated by the ATAM process needs to be **presented to stakeholders**.
- The evaluation team can write a **report** and present the ideas with slides.
- The lead architect must report the **business context and drivers**, requirements and constraints.
- As well as the documentation of the selected architecture, the set of prioritized scenarios, the utility tree and the discovered **risks, non-risks**, sensitivity points and tradeoffs.

NOTES

- 1 **Step 9:** the information generated by ATAM is **presented to the stakeholders**.
- 2 The evaluation team writes a **report** and presents it with slides.
- 3 The lead architect reports the **business context and drivers**, requirements and constraints.
- 4 Plus the documentation of the selected architecture, the prioritised scenarios, the utility tree, and the **risks, non-risks, sensitivity points and trade-offs** discovered.

A Lightweight ATAM

- A Lightweight Architecture Evaluation method have been developed which is based on the ATAM, for **smaller, less risky projects**.
 - ✓ May take place in a **single day**, or even a **half-day** meeting.
 - ✓ May be carried out entirely by members **internal to the organization**.
 - ✓ However, this **lower level of scrutiny** and objectivity may not probe the architecture as deeply.
- Because the participants are all internal to the organization and fewer in number than for the ATAM, giving everyone their say and achieving a shared understanding takes much less time.
- The steps and phases of a Lightweight Architecture Evaluation can be carried out more quickly.

NOTES

- 1 A **lightweight** evaluation method based on ATAM exists for **smaller, less risky** projects.
- 2 It may take a **single day or even half a day**.
- 3 It may be run entirely by people **internal** to the organisation. Because participants are internal and fewer, giving everyone their say and reaching shared understanding is much faster, and the steps run more quickly.
- 4 The cost: a **lower level of scrutiny and objectivity**, so it may not probe the architecture as deeply as a full ATAM with an external team.

EXAM ANGLE

Past exam · Final 252 · Q8-2 · 1 mark

Q1

What is the main difference between ATAM and lightweight ATAM?

Answer: Lightweight ATAM keeps the same nine steps but is **scaled down for smaller, lower-risk projects**: it can be done in about half a day (or a day), **using only people inside the organisation**. It is faster, but less deep and less objective than a full ATAM run with an external evaluation team.

A Lightweight ATAM - Agenda

Step	Time	Notes
1. Present the ATAM	0 hours	Participants already familiar with process.
2. Present business drivers	0.25 hours	The participants are expected to understand the system and its business goals and their priorities. A brief review ensures that these are fresh in everyone's mind and that there are no surprises.
3. Present architecture	0.5 hours	All participants are expected to be familiar with the system. A brief overview of the architecture, using at least module and C&C views, is presented. 1-2 scenarios are traced through these views.
4. Identify architectural approaches	0.25 hours	The architecture approaches for specific quality attribute concerns are identified by the architect. This may be done as a portion of step 3.
5. Generate QA utility tree	0.5- 1.5 hours	Scenarios might exist: part of previous evaluations, part of design, part of requirements elicitation. Put these in a tree. Or, a utility tree may already exist.
6. Analyze architectural approaches	2-3 hours	This step—mapping the highly ranked scenarios onto the architecture—consumes the bulk of the time and can be expanded or contracted as needed.
7. Brainstorm scenarios	0 hours	This step can be omitted as the assembled (internal) stakeholders are expected to contribute scenarios expressing their concerns in step 5.
8. Analyze architectural approaches	0 hours	This step is also omitted, since all analysis is done in step 6.
9. Present results	0.5 hours	At the end of an evaluation, the team reviews the existing and newly discovered risks, nonrisks, sensitivities, and tradeoffs and discusses whether any new risk themes have arisen.

Prince Sultan University | Slides prepared from "Software Engineering Design: Theory and Practice" book and slides

30

NOTES

- 1 The lightweight agenda. Step 1 (present ATAM) takes **0 hours**, because participants already know the process; steps 2–4 take 15–30 minutes each, because everyone knows the system (1–2 scenarios are traced through the module and C&C views in step 3).
- 2 Step 5 (utility tree) takes 0.5–1.5 hours, often reusing existing scenarios; step 6 (analysis) takes the bulk, **2–3 hours**, mapping the top scenarios onto the architecture.
- 3 Steps **7 and 8 are dropped** (0 hours): the internal stakeholders already contributed scenarios in step 5, and all analysis happens in step 6. Step 9 (results) takes 0.5 hours to review the risks, non-risks, sensitivities and trade-offs. Total: about half a day to a day.

Summary

➤ In this session, we presented:

- ✓ Architecture Evaluation
- ✓ Benefits of Architecture Evaluation
- ✓ Methods of Architecture Evaluation
- ✓ ATAM Evaluation

NOTES

- 1 Recap: why and when to evaluate an architecture, its benefits, the methods (ATAM, SAAM, ARID: all scenario- and quality-based), and ATAM's participants, nine steps, phases and outputs, plus the lightweight variant.
- 2 Next: documenting the architecture (Chapter 9).

Original slides from the SE322 lecture deck, with study notes. Check exact wording against the [original deck](#).
Notes and questions are study aids. Made by Shoug Alomran.