17 ноября 2018 г.

Reva P., Volobuieva O., Saukh D., Bahno O.

|  |
| --- |
|  |
| Architecture Tradeoff Analysis Method |
|  |
| ProFinder |

Architecture Tradeoff Analysis Method

ProFinder

Content

[**ATAM. What is it?** 2](#_Toc531555614)

[**What are the outputs of ATAM?** 3](#_Toc531555615)

[**Participants in ATAM** 4](#_Toc531555616)

[**Phases of ATAM** 4](#_Toc531555617)

[**The steps of the ATAM** 5](#_Toc531555618)

[**Step 1 - Present the ATAM** 5](#_Toc531555619)

[**Step 2 - Present Business Drivers** 6](#_Toc531555620)

[**Step 3 - Present Architecture** 7](#_Toc531555621)

[**Step 4 - Identify Architectural Approaches** 7](#_Toc531555622)

[**Step 5 - Generate Quality Attribute Utility Tree** 7](#_Toc531555623)

[**Step 6 - Analyze Architectural Approaches** 8](#_Toc531555624)

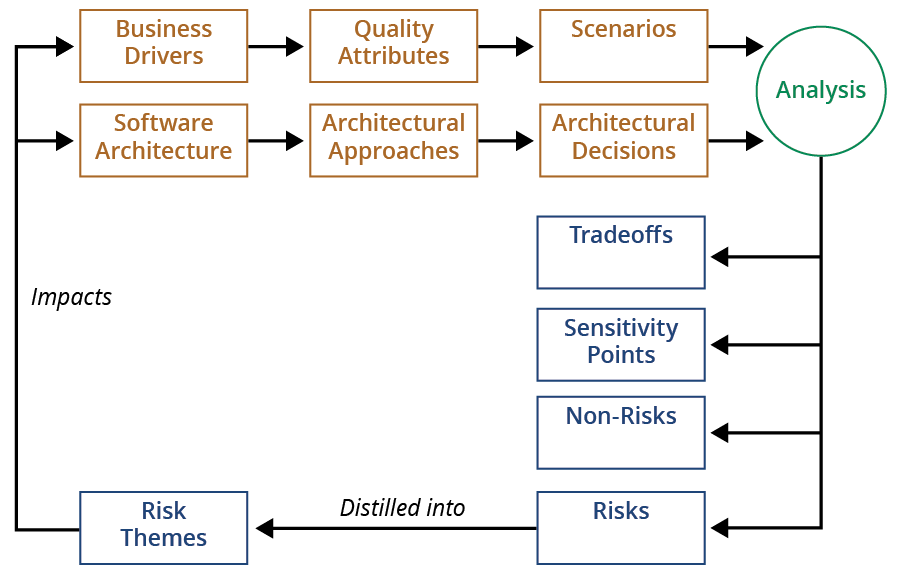
[**Step 7 - Brainstorm and Prioritize Scenarios** 9](#_Toc531555625)

[**Step 8 - Analyze Architectural Approaches** 10](#_Toc531555626)

[**Step 9 - Present Results** 10](#_Toc531555627)

**ATAM. What is it?**

The Architecture Tradeoff Analysis Method (ATAM) is a method for evaluating software architectures relative to quality attribute goals. ATAM evaluations expose architectural risks that potentially inhibit the achievement of an organization's business goals. The ATAM gets its name because it not only reveals how well an architecture satisfies particular quality goals, but it also provides insight into how those quality goals interact with each other—how they trade off against each other.

The ATAM is the leading method in the area of software architecture evaluation. An evaluation using the ATAM typically takes three to four days and gathers together a trained evaluation team, architects, and representatives of the architecture's various stakeholders.

**What are the outputs of ATAM?**

– A set of **architectural approaches** identified and or applied: Sometimes we can identify architectural approaches that cannot be applied on our architecture

– **A Utility Tree**: a top-down mechanism for directly and efficiently translating the business drivers of a system into concrete quality attribute scenarios.

– A **set of scenarios** identified and the subset that had been effectively mapped in the architecture.

– A **set of questions about the quality attributes** in the architecture and the answers to these questions. In our case our questions are a set of metrics and the answers are the values measured.

– The **risks identified**: risks that the architecture is able to mitigate and the risks that threaten the system and the business goals.

Architectural Approaches

Attribute - Specific Questions

High-Priority Scenarios

Risks

Tradeoff Points

Sensitivity Points

**Participants in ATAM**

– The evaluation team

* Team leader
* Evaluation leader
* Scenario scribe
* Proceedings scribe
* Timekeeper
* Process observer
* Process enforcer
* Questioner

– Project decision makers

– Architecture stakeholders

* Developers, testers
* Users
* Builders of systems interacting with this one

**Phases of ATAM**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Activity** | **Participants** | **Duration** |
| 0 | Preparation | Team leadership/key project decision makers | Over a few weeks |
| 1 | Evaluation | Evaluation team and project decision makers | 1 day + hiatus of 2 or 3 weeks |
| 2 | Evaluation with Stakeholders | Ditto + stakeholders | 2 days |
| 3 | Follow-up | Evaluation team and client | 1 week |

**The steps of the ATAM**

The ATAM process consists of nine steps, as was briefly presented in Section 3. Sometimes there must be dynamic modifications to the order of steps to accommodate the availability of personnel or architectural information. Although the steps are numbered, suggesting linearity, this is not a strict waterfall process. There will be times when an analyst will return briefly to an earlier step, or will jump forward to a later step, or will iterate among steps, as the need dictates. The importance of the steps is to clearly delineate the activities involved in ATAM along with the outputs of these activities.

The amount of time it takes to carry out an ATAM varies depending on the size of the system and the maturity of the architecture and the state of its description. We have conducted ATAMs by executing the steps in three consecutive days. However, we have found that the most effective ATAMs have been those in which the ATAM team and the customer establish a common understanding of the architecture during an initial phase of the evaluation and then later assemble a larger group of stakeholders for a more formal evaluation meeting.

**Step 1 - Present the ATAM**

In this step the evaluation team lead presents the ATAM to the assembled stakeholders. This time is used to explain the process that everyone will be following, allows time to answer questions, and sets the context and expectations for the remainder of the activities. It is important that everyone knows what information will be collected, how it will be scrutinized, and to whom it will be reported. In particular, the presentation will describe

• the ATAM steps in brief

• the techniques that will be used for elicitation and analysis: utility tree generation, architectural approach-based elicitation/analysis, and scenario brainstorming/mapping

• the outputs from the evaluation: the scenarios elicited and prioritized, the questions used to understand/evaluate the architecture, a “utility” tree, describing and prioritizing the driving architectural requirements, the set of identified architectural approaches and styles, the set of risks and non-risks discovered, the set of sensitivity points and tradeoffs discovered.

**Step 2 - Present Business Drivers**

The system to be evaluated needs to be understood by all participants in the evaluation. In this step the project manager presents a system overview from a business perspective. A sample outline for such a presentation is given in Figure 1. The system itself must be presented, initially at a high level of abstraction, typically describing:

• its most important functional requirements

• its technical, managerial, economic, or political constraints

• its business goals and context

• its major stakeholders

• the architectural drivers (major quality attribute goals that shape the architecture)

Figure 1:

Business Context/Drivers Presentation (~ 12 slides; 45 minutes)

- Description of the business environment, history, market differentiators, driving requirements, stakeholders, current need and how the proposed system will meet those needs/ requirements (3-4 slides)

- Description of business constraints (e.g., time to market, customer demands, standards, cost, etc.) (1-3 slides)

- Description of the technical constraints (e.g., COTS, interoperation with other systems, required hardware or software platform, reuse of legacy code, etc.) (1-3 slides)

- Quality attributes desired (e.g., performance, availability, security, modifiability, interoperability, integrality) and what business needs these are derived from (2-3 slides)

- Glossary (1 slide)

**Step 3 - Present Architecture**

The architecture will be presented by the lead architect (or architecture team) at an appropriate level of detail. What is an appropriate level? This depends on several factors: how much information has been decided upon and documented; how much time is available; how much risk the system faces. This is an important step as the amount of architectural information available and documented will directly affect the analysis that is possible and the quality of this analysis. Frequently the evaluation team will need to specify additional architectural information that is required to be collected and documented before a more substantial analysis is possible.

In this presentation the architecture should cover

• technical constraints such as an OS, hardware, or middleware prescribed for use

• other systems with which the system must interact

• architectural approaches used to meet quality attribute requirements

At this time the evaluation team begins its initial probing of architectural approaches.

**Step 4 - Identify Architectural Approaches**

The ATAM focuses on analyzing an architecture by understanding its architectural approaches. In this step they are identified by the architect, and captured by the analysis team, but are not analyzed.

We concentrate on identifying architectural approaches and architectural styles 1 because these represent the architecture’s means of addressing the highest priority quality attributes; that is, the means of ensuring that the critical requirements are met in a predictable way. These architectural approaches define the important structures of the system and describe the ways in which the system can grow, respond to changes, withstand attacks, integrate with other systems, and so forth.

**Step 5 - Generate Quality Attribute Utility Tree**

In this step the evaluation team works with the architecture team, manager, and customer representatives to identify, prioritize, and refine the system’s most important quality attribute goals. This is a crucial step in that it guides the remainder of the analysis. Analysis, even at the level of software architecture, is not inherently bound in scope. So we need a means of focusing the attention of all the stakeholders on the aspects of the architecture that are most critical to the system’s success. We do this by building a utility tree.

The output of the utility tree generation process is a prioritization of specific quality attribute requirements, realized as scenarios. This prioritized list provides a guide for the remainder of the ATAM. It tells the ATAM team where to spend its limited time, and in particular where to probe for architectural approaches and their consequent risks, sensitivity points, and tradeoffs. Additionally, the utility tree serves to concretize the quality attribute requirements, forcing the evaluation team and the customer to define their “ility” requirements precisely.

**Step 6 - Analyze Architectural Approaches**

Once the scope of the evaluation has been set by the utility tree elicitation, the evaluation team can then probe for the architectural approaches that realize the important quality attributes. This is done with an eye to documenting these architectural decisions and identifying their risks, sensitivity points, and tradeoffs.

What are we eliciting in this step? We are eliciting sufficient information about each architectural approach to conduct a rudimentary analysis about the attribute for which the approach is relevant. What are we looking for in this rudimentary analysis? We want to be convinced that the instantiation of the approach in the architecture being evaluated holds significant promise for meeting the attribute-specific requirements for which it is intended.

The major outputs of this phase are a list of architectural approaches or styles, the questions associated with them, and the architect’s response to these questions. Frequently a list of risks, sensitivity points, and tradeoffs are generated. Each of these is associated with the achievement of one or more utility tree sub-factors, with respect to the quality attribute questions that probed the risk. In effect, the utility sub-factor tells us where to probe the architecture (because this is a highly important factor for the success of the system), the architect (hopefully) responds with the architectural approach that answers this need, and we use the quality attribute-specific questions to probe the approach more deeply. The questions help us to

• understand the approach

• look for well-known weaknesses with the approach

• look for the approach’s sensitivity points

• find interactions and tradeoffs with other approaches

In the end, each of these may provide the basic material for the description of a risk and this is recorded in an ever-growing list of risks.

The first action in this step is to associate the highest priority quality attribute requirements (as identified in the utility tree of Step 5) with the architectural approaches (from Step 4) used to realize them. For each highly ranked scenario generated by the utility-tree-creation step, the architect should identify the components, connectors, configuration, and constraints involved.

The evaluation team and the architecture team address each architectural approach presented by asking a set of approach-specific and quality-attribute-specific questions. These questions might come from documented experience with styles (as found in ABASs and their associated quality attribute characterizations), from books on software architecture or from the prior experiences of the assembled stakeholders. In practice we mine all three areas for questions.

**Step 7 - Brainstorm and Prioritize Scenarios**

Scenarios are the motor that drives the testing phase of the ATAM. Generating a set of scenarios has proven to be a great facilitator of discussion and brainstorming, when greater numbers of stakeholders are gathered to participate in the ATAM. Scenarios are examples of architectural stimuli used to both

• represent stakeholders’ interests

• understand quality attribute requirements

The stakeholders now undertake two related activities: brainstorming use case scenarios (representing the ways in which the stakeholders expect the system to be used) and change scenarios (representing the ways in which the stakeholders expect the system to change in the future). Use case scenarios are a kind of scenario where the stakeholder is an end user, using the system to execute some function. Change scenarios represent changes to the system and are sub-divided into two categories: growth scenarios and exploratory scenarios.

Growth scenarios represent ways in which the architecture is expected to accommodate growth and change in the moderate near term: expected modifications, changes in performance or availability, porting to other platforms, integration with other software, and so forth. Exploratory scenarios, on the other hand, represent extreme forms of growth: ways in which the architecture might be stressed by changes: dramatic new performance or availability requirements (order of magnitude changes, for example), major changes in the infrastructure or mission of the system, and so forth. Growth scenarios are a way of showing the strengths and weaknesses of the architecture with respect to anticipated forces on the system. Exploratory scenarios are an attempt to find sensitivity points and tradeoff points. The identification of these points help us assess the limits of the system with respect to the models of quality attributes that we build.

Once the scenarios have been collected, they must be prioritized. We typically do this via a voting procedure where each stakeholder is allocated a number of votes equal to 30% of the number of scenarios, rounded up. So, for instance, if there were eighteen scenarios collected, each stakeholder would be given six votes. These votes can be allocated in any way that the stakeholder sees fit: all six votes allocated to one scenario, two votes to each of three scenarios, one vote to each of six scenarios, etc. In addition, at this point any stakeholder can suggest merging multiple scenarios if they are felt to represent the same stimulus/response behavior.

The prioritization and voting can be an open or a secret balloting procedure. Once the votes have been made, they are tallied and the scenarios are prioritized. A cutoff is typically made that separates the high-priority scenarios from the lower ones, and only the high-priority ones are considered in future evaluation steps. For example, a team might only consider the top five scenarios.

**Step 8 - Analyze Architectural Approaches**

After the scenarios have been collected and so analyzed, the architect then begins the process of mapping the highest ranked scenarios onto whatever architectural descriptions have been presented. Ideally this activity will be dominated by the architect’s mapping of scenarios onto previously discussed architectural approaches. In fact, the whole point of the hiatus between the two phases is to ensure that this is the case. If this is not the case then either the architect has no approach- or style-based (and hence no architecture-wide) solution for the stimulus that the scenario represents, or the approach exists but was not revealed by any activities up until this point.

In this step we reiterate Step 6, mapping the highest ranked newly generated scenarios onto the architectural artifacts thus far uncovered. Assuming Step 7 didn’t produce any high-priority scenarios that were not already covered by previous analysis, Step 8 is a testing activity: We hope and expect to be uncovering little new information. This is a testing activity: at this point we hope and expect to be uncovering little new information.

If we do uncover new information, then this was a failing of our utility tree exercise and the architectural approaches that it led us to investigate. At this point we would need to go back to Step 4 and work through it, as well as Steps 5 and 6 until no new information is uncovered.

**Step 9 - Present Results**

Finally, the collected information from the ATAM needs to be summarized and presented back to the stakeholders. This presentation typically takes the form of a verbal report accompanied by slides but might, in addition, be accompanied by a more complete written report delivered subsequent to the ATAM. In this presentation we recapitulate the steps of the ATAM and all the information collected in the steps of the method including: the business context, driving requirements, constraints, and the architecture. Most important, however, is the set of ATAM outputs:

• the architectural approaches/styles documented

• the set of scenarios and their prioritization

• the set of attribute-based questions

• the utility tree

• the risks discovered

• the non-risks documented

• the sensitivity points and tradeoff points found

Each of these findings will be described and in some cases we might offer some mitigation strategies. Because we are systematically working through and trying to understand the architectural approaches it is inevitable that, at times, we make some recommendations on how the architecture might have been designed or analyzed differently. These mitigation strategies may be process related (e.g., a database administrator stakeholder should be consulted before completing the design of the system administration user interface), they may be managerial (e.g., three sub-groups within the development effort are pursuing highly similar goals and these should be merged), or they may be technical (e.g., given the estimated distribution of customer input requests, additional server threads need to be allocated to ensure that worst-case latency does not exceed five seconds). However, offering mitigation strategies is not an integral part the ATAM. The ATAM is about locating architectural risks. Addressing them may be done in any number of ways.