

# Architecture Tradeoff Analysis Method<sup>®</sup> - ATAM<sup>®</sup>

## SKA Telescope Manager Architecture Evaluation

Copyright 2018 Carnegie Mellon University. All Rights Reserved.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

Architecture Tradeoff Analysis Method® and ATAM® are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM18-0493

# Evaluation Team

## **Internal:**

- Nick Rees (Chair)
- Maurizio Miccolis (Secretary)
- Juande Santander Vela
- Lorenzo Pivetta
- Antonio Chrysostomou

## **External:**

- Jeff Kern (NRAO – USA)
- Tim Jenness (LSST– USA)
- Mary Popeck (Software Engineering Institute - USA)
- John Klein (Software Engineering Institute – USA)

# Outline

## Why Analyze an Architecture?

### The ATAM

- Phase 0 – Partnership and Preparation
- Phase 1 – Initial Evaluation
- Phase 2 – Complete Evaluation
- Phase 3 – Follow-Up

### Rules of Engagement

### Summary

# Architecture Analysis

## Why Analyze an Architecture?

- *All* design involves tradeoffs.
- A software architecture is an artifact that embodies significant design decisions.

## The ATAM

- The Architecture Tradeoff Analysis Method (ATAM) was developed by the Software Engineering Institute.
- The purpose of the ATAM is *to assess the consequences of architectural decisions in light of quality attribute requirements and business goals.*

# Purpose of the ATAM

**SKAO will use the ATAM findings as one input to the TM Element CDR.**

The ATAM is a method that helps stakeholders ask the right questions to discover potentially problematic architectural decisions.

Discovered risks can then be made the focus of mitigation activities—for example, further design, further analysis, and prototyping.

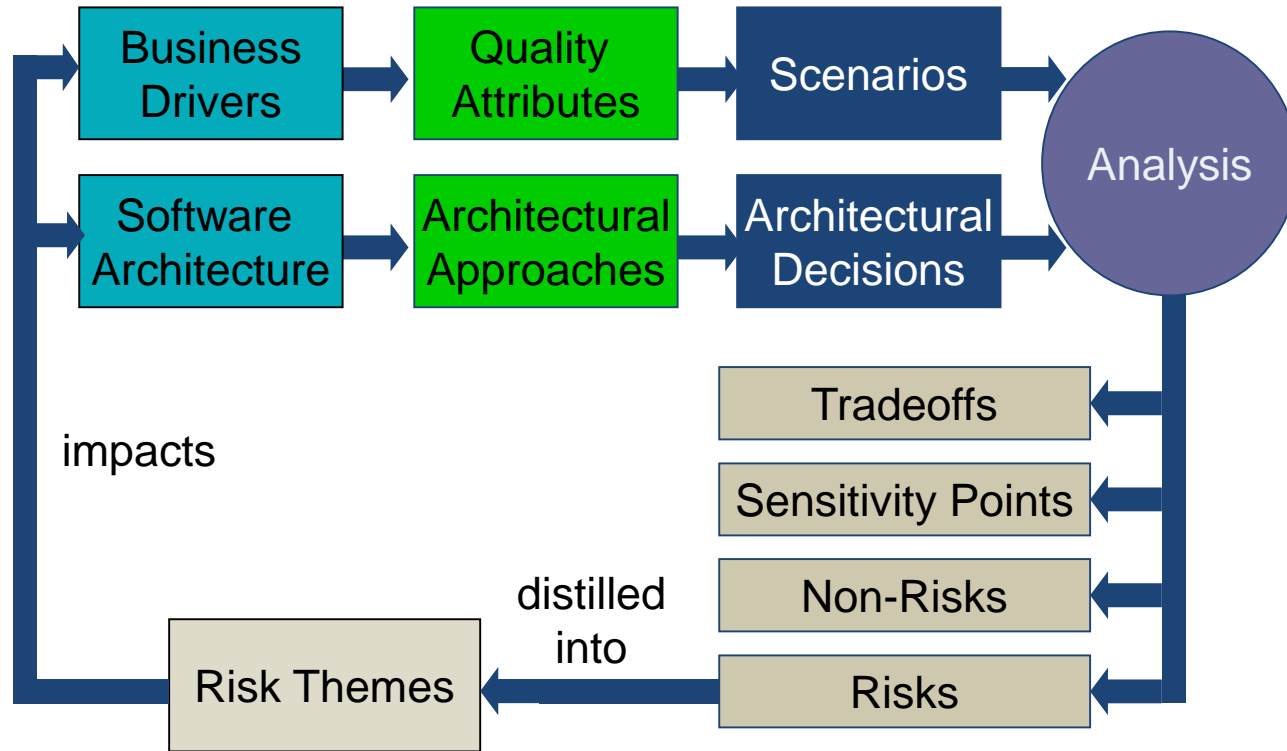
Tradeoffs can be explicitly identified and documented.

The purpose of the ATAM is **NOT** to provide precise analyses of every function or requirement. . .

the purpose **IS** to discover risks created by architectural decisions.

We want to find *trends*: correlations between architectural decisions and predictions of system properties.

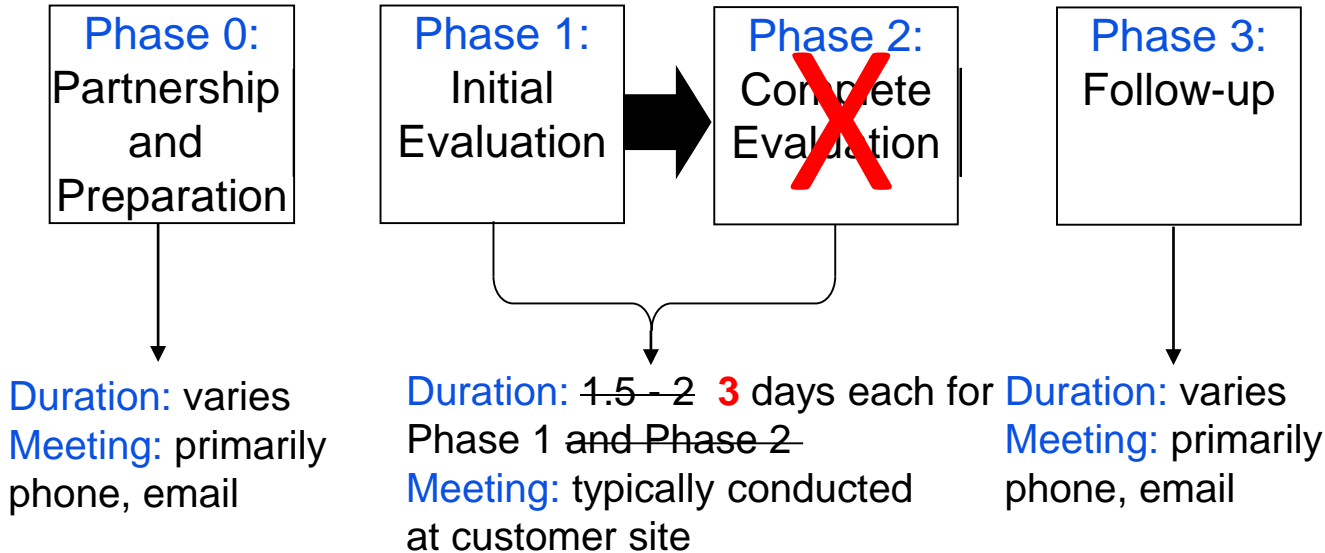
# Conceptual Flow of the ATAM



# Nominal ATAM Phases

We are tailoring the ATAM process for the TM Evaluation

ATAM evaluations are conducted in ~~four~~ **three** phases.



# ATAM Phase 1

**Phase 1:** this phase involves a small group of predominantly technically oriented stakeholders.

Phase 1 is

- architecture-centric
- focused on eliciting detailed architectural information and analyzing it
- top-down analysis

# ATAM Phase 1 Steps

1. Present the ATAM.
2. Present business drivers.
3. Present architecture.
4. Identify architectural approaches.
5. Generate quality attribute utility tree.
6. Analyze architectural approaches.
- ~~7. Brainstorm and prioritize scenarios.~~
- ~~8. Analyze architectural approaches.~~
9. Present results.

**Phase 1**

**Usually Phase 2 - Covered here in Steps 5 & 6**

**Added to  
TM Phase 1**

# Steps 1 – 4

We are doing Step 1 now 😊

Step 2 (Business Drivers) and Step 3 (Architecture Presentation) will be very abbreviated

- We are all familiar with the SKA drivers at this point
- The TM Architecture team has provided good architecture documentation, and the evaluation team has reviewed this before our meeting today

Step 4 (Architecture Drivers) will be a brief check of the evaluation teams understanding of the architecture

We will spend most of our time on Steps 5 (Utility Tree) and Step 6 (Architecture Analysis).

# Step 5: Generate Quality Attribute Utility Tree

Identify, prioritize, and refine the most important quality attribute goals by building a *utility tree*.

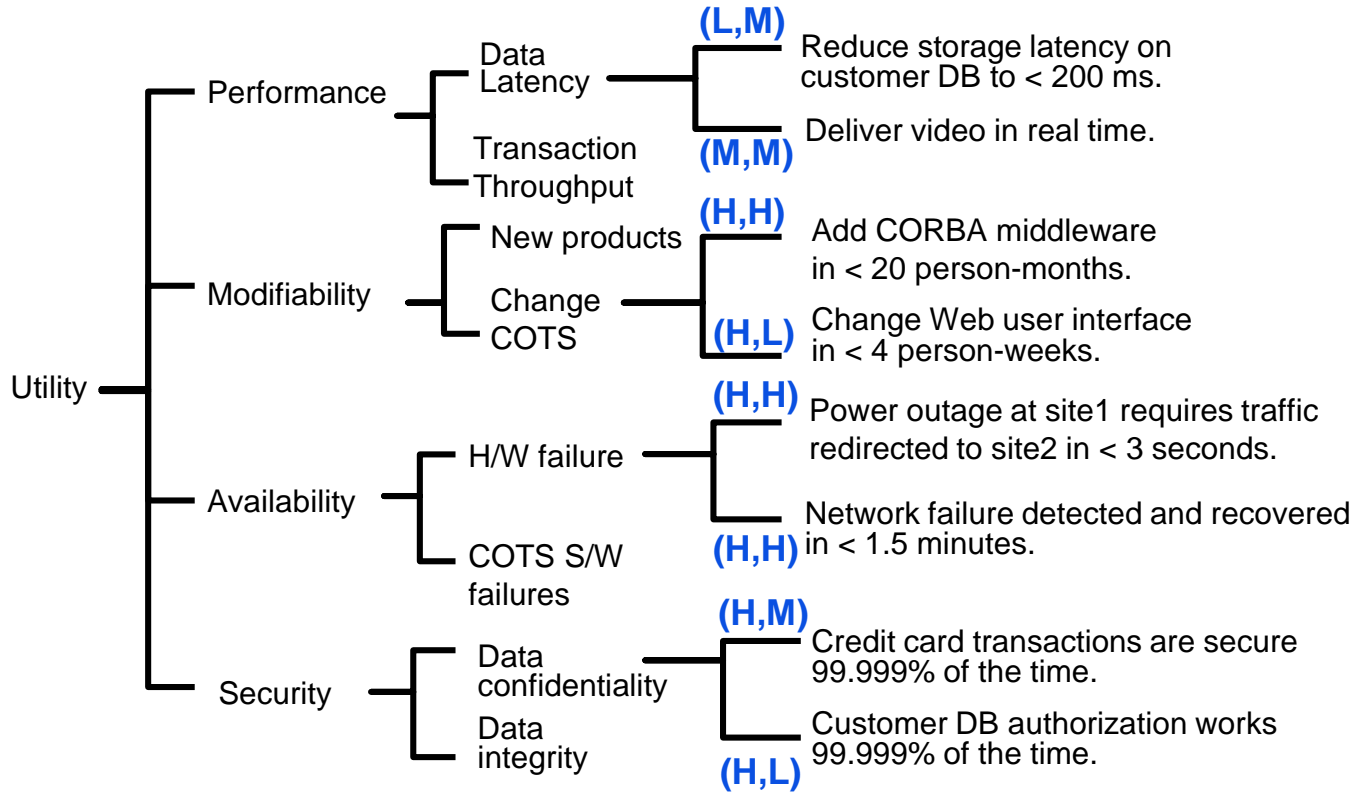
- A utility tree is a top-down vehicle for characterizing and prioritizing the “driving” attribute-specific requirements.
- The driving quality attributes are the high-level nodes (typically performance, modifiability, security, and availability).
- Scenarios are the leaves of the utility tree.

Output: a characterization and a prioritization of specific quality attribute requirements

The evaluation team has started to develop a utility tree.

We will review that work and extend it today.

# Utility Tree Construction



# Scenarios

Scenarios are used to

- represent *stakeholders'* interests
- understand quality attribute requirements

Scenarios should cover a range of information:

- use case scenarios: anticipated uses of the system
- growth scenarios: anticipated changes to the system
- exploratory scenarios: unanticipated stresses to the system

A good scenario makes clear what stimulus causes it and what responses are of interest.

- Specific case
- Positive assertion, not a question

# Stimuli, Environment, Responses

## Use case scenario

*Remote user requests a database report via the Web **during a peak period** and receives it within 5 seconds.*

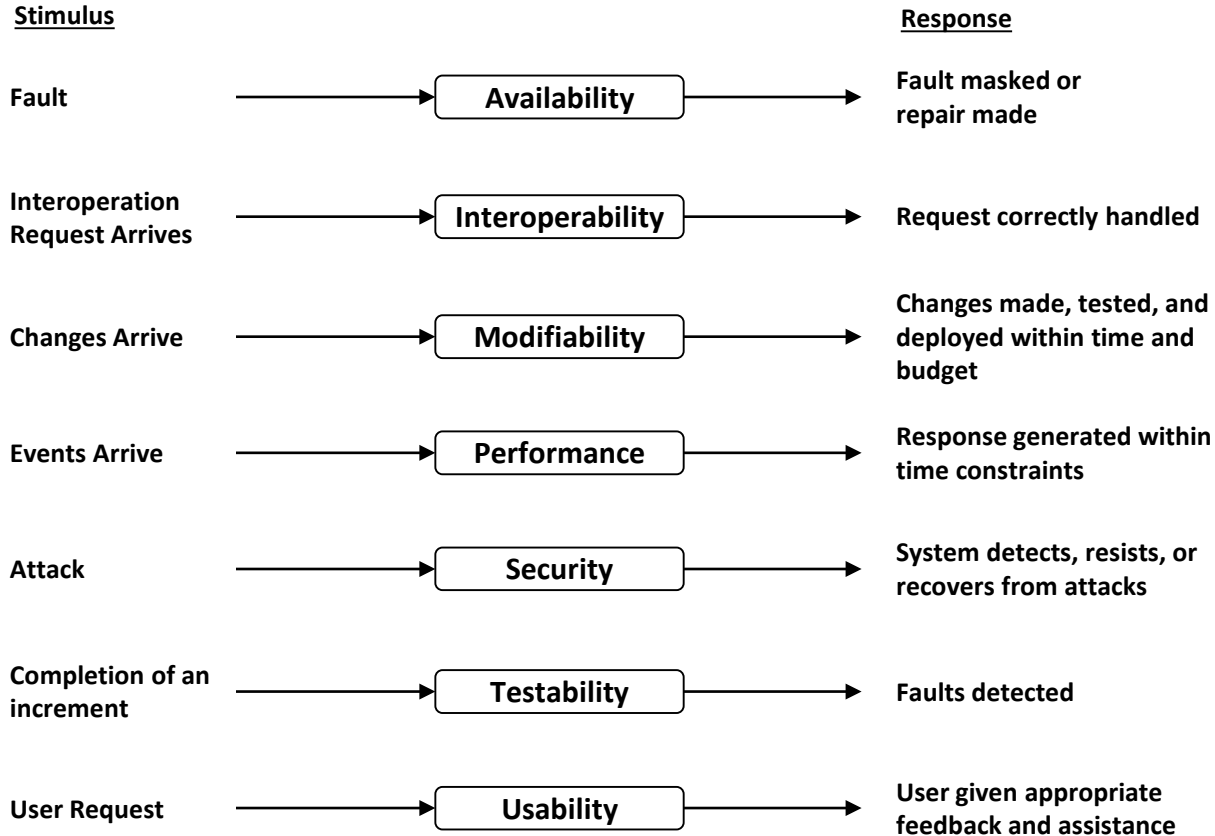
## Growth scenario

*Add a new data server **to reduce latency in scenario 1 to 2.5 seconds** within 1 person-week.*

## Exploratory scenario

*Half of the servers go down **during normal operation** without affecting overall system availability.*

# Scenario Templates



# Prioritize Scenarios – First Dimension Importance

The importance dimension is rated high, medium, or low

- High – If this scenario can't be satisfied the system will be useless.
- Medium – It would be highly desirable for the system to satisfy this scenario.
- Low – Satisfying this scenario would be a “nice-to-have” feature.

SKAO (Nick and Robert) will make this prioritization.

# Secenario Prioritization – Second Dimension Difficulty

The difficulty dimension is rated high, medium, or low

- High – The architect and technical team are very concerned about satisfying this scenario.
  - You are joking, right?
  - lack experience, lack expertise, have no idea how to satisfy the scenario
- Medium – The architect and technical team are concerned about satisfying this scenario.
  - difficult, time consuming, but generally understand how to satisfy the scenario
- Low – Satisfying this scenario is relatively easy.

TM Architects will make this prioritization.

# Step 6: Analyze Architectural Approaches

Purpose: To analyze the highest priority scenarios.

- Begin by reading the highest priority scenarios.
  - We are planning to analyze 8-10 scenarios
- Spend some time to refine the scenario into its six parts. This will produce better understanding of the scenario.
- Ask the architect(s) to explain how the architecture responds to the stimuli.
  - Architects should use existing pictures and/or they may draw new pictures as necessary.
- Evaluators will ask quality-attribute-specific questions to uncover risks, non-risks, sensitivity points, and tradeoffs.

# Quality Attribute Questions

Quality attribute questions probe architectural decisions that bear on quality attribute requirements.

## Performance

- How are concurrent write operations handled?
- What are the message-arrival rates?

## Modifiability

- Are there any places where layers/facades are circumvented?
- What components rely on detailed knowledge of message formats?

# Risks, Non-Risks, Sensitivities, and Tradeoffs

A **risk** is a potentially problematic architectural decision.

**Non-risks** are good architectural decisions that are frequently implicit in the architecture.

A **sensitivity point** is a property of one or more components (and/or component relationships) that is critical for achieving a particular quality attribute response.

A **tradeoff point** is a property that affects more than one attribute and is a sensitivity point for more than one attribute.

# Risks and Tradeoffs

## Example risk

*“Rules for writing business logic modules in the second tier of your 3-tier architecture are not clearly articulated. This could result in replication of functionality, thereby compromising modifiability of the third tier.”*

## Example tradeoff

*“Changing the level of encryption could have a significant impact on both security and performance.”*

# Sensitivity Points and Non-Risks

## Example sensitivity point

*“The average number of person-days of effort it takes to maintain a system might be sensitive to the degree of encapsulation of its communication protocols and file formats.”*

## Example non-risk

*“Assuming message arrival rates of once per second, a processing time of less than 30 ms, and the existence of one higher priority process, a 1-second soft deadline seems reasonable.”*

# Step 9: Present Results

Recapitulate all the steps of the ATAM and present the ATAM outputs, including

- architectural approaches
- utility tree
- scenarios
- risks and non-risks
- sensitivity points and tradeoffs
- risk themes

We will do this on Thursday afternoon.

ATAM is a transparent process – you will see the findings before you leave.

# SKA Organisation Code of Conduct for Meetings

These meetings will be conducted in compliance with the **SKA Organisation Code of Conduct for Meetings**

<https://www.skatelescope.org/ska-organisation-code-of-conduct-for-meetings/>

# For Additional Information

## **ATAM Process:**

John Klein [jklein@sei.cmu.edu](mailto:jklein@sei.cmu.edu)

Mary Popeck [mpopeck@sei.cmu.edu](mailto:mpopeck@sei.cmu.edu)

## **CDR Process:**

Nick Rees [N.Rees@skatelescope.org](mailto:N.Rees@skatelescope.org)

Marco Bartolini [M.Bartolini@skatelescope.org](mailto:M.Bartolini@skatelescope.org)