

The Architecture Tradeoff Analysis Method[®] (ATAM[®])

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



Software Engineering Institute

Carnegie Mellon

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution

© 2018 Carnegie Mellon University

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.

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

DM18-0698



Goals for this meeting

SDP software architecture team understands:

- The process that SKAO will be using to perform the SDP software architecture evaluation
- What the SDP software architecture team will do during the June pre-CDR meeting



Evaluation Team

Internal:

- Nick Rees (Chair)
- Maurizio Miccolis (Secretary)
- Juande Santander Vela
- Miles Deegan
- Anna Bonaldi

External:

- Andre' Offringa (ASTRON)
- Bernd Mohr (Jüelich Supercomputing Centre)
- Norbert Eicker (Jüelich Supercomputing Centre)
- 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~~ Complete Evaluation
- ~~Phase 2 – Complete Evaluation~~
- Phase 3 – Follow-Up

Rules of Engagement

Summary



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 – 1

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.



Purpose of the ATAM – 2

The purpose of the ATAM is **NOT** to provide precise analyses. . .
the purpose **IS** to discover risks created by architectural decisions.

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



ATAM Benefits

Performing ATAM evaluations has these benefits:

- identified risks
- clarified quality attribute requirements
- improved architecture documentation
- documented basis for architectural decisions
- increased communication among stakeholders

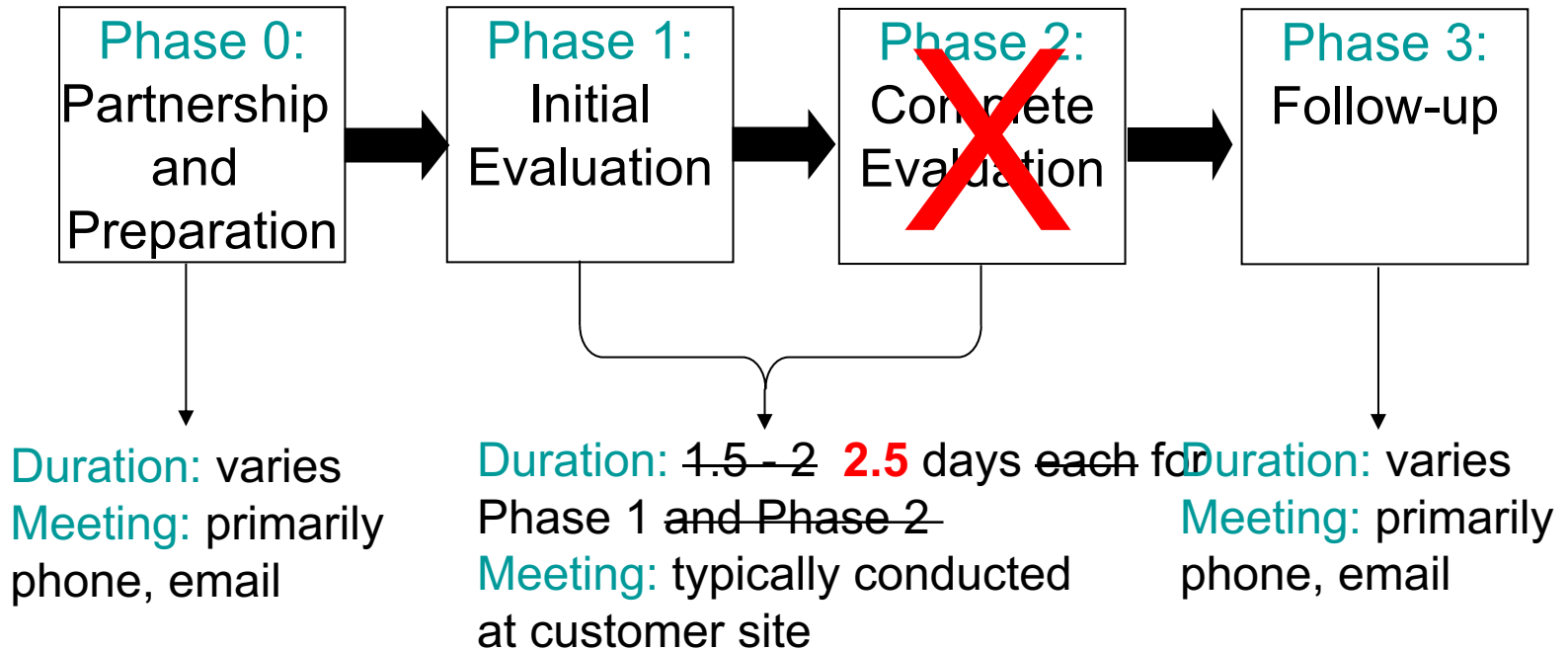
The end result is improved architectures.



Nominal ATAM Phases

We are tailoring the ATAM process for the SDP Evaluation

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



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



ATAM Phase 0

Phase 0: this phase precedes the technical evaluation. During it, the customer and a subset of the evaluation team

- exchange their understanding about the method and the system whose architecture is to be evaluated
- come to an agreement about performing the evaluation
- field a core evaluation team



Outline

Why Analyze an Architecture?

The ATAM

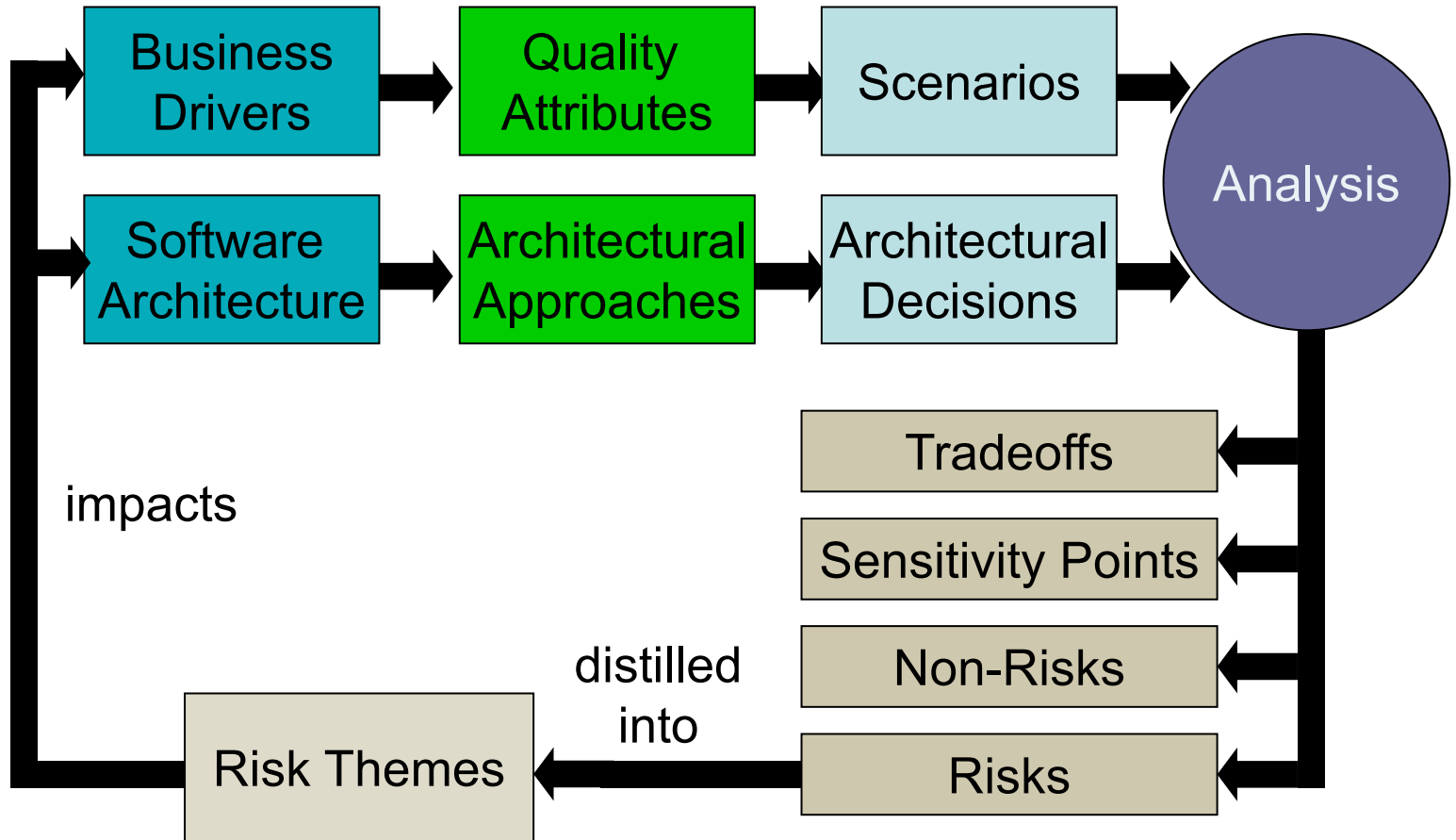
- Phase 0 – Partnership and Preparation
- **Phase 1 – Initial Complete Evaluation**
- ~~Phase 2 – Complete Evaluation~~
- Phase 3 – Follow-Up

Rules of Engagement

Summary



Conceptual Flow of the ATAM



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

**Added to
SDP Phase 1**



2. Present Business Drivers

The ATAM customer representative describes the system's business drivers including the

- business context for the system
- high-level functional requirements
- high-level quality attribute requirements
 - architectural drivers: quality attributes that “shape” the architecture
 - critical requirements: quality attributes that are most central to the system's success

Nick Rees is going to handle this prior to the June pre-CDR meeting.



3. Present Architecture

The architect presents an overview of the architecture including

- technical constraints such as an OS, hardware, or middleware prescribed for use
- other systems with which the system must interact
- architectural approaches used to address quality attribute requirements

The SDP architecture team will make a brief presentation: Scope boundaries, major components and responsibilities, strategies and principles, key concepts.



4. Identify Architectural Approaches

Identify predominant architectural approaches such as

- client-server
- 3-tier
- watchdog
- publish-subscribe
- redundant hardware

The evaluators begin to identify places in the architecture that are key to realizing the system's quality attribute goals.

Evaluation team is starting this now, based on the architecture documentation. We will validate this at the June pre-CDR meeting.



5. Generate Quality Attribute Utility Tree

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

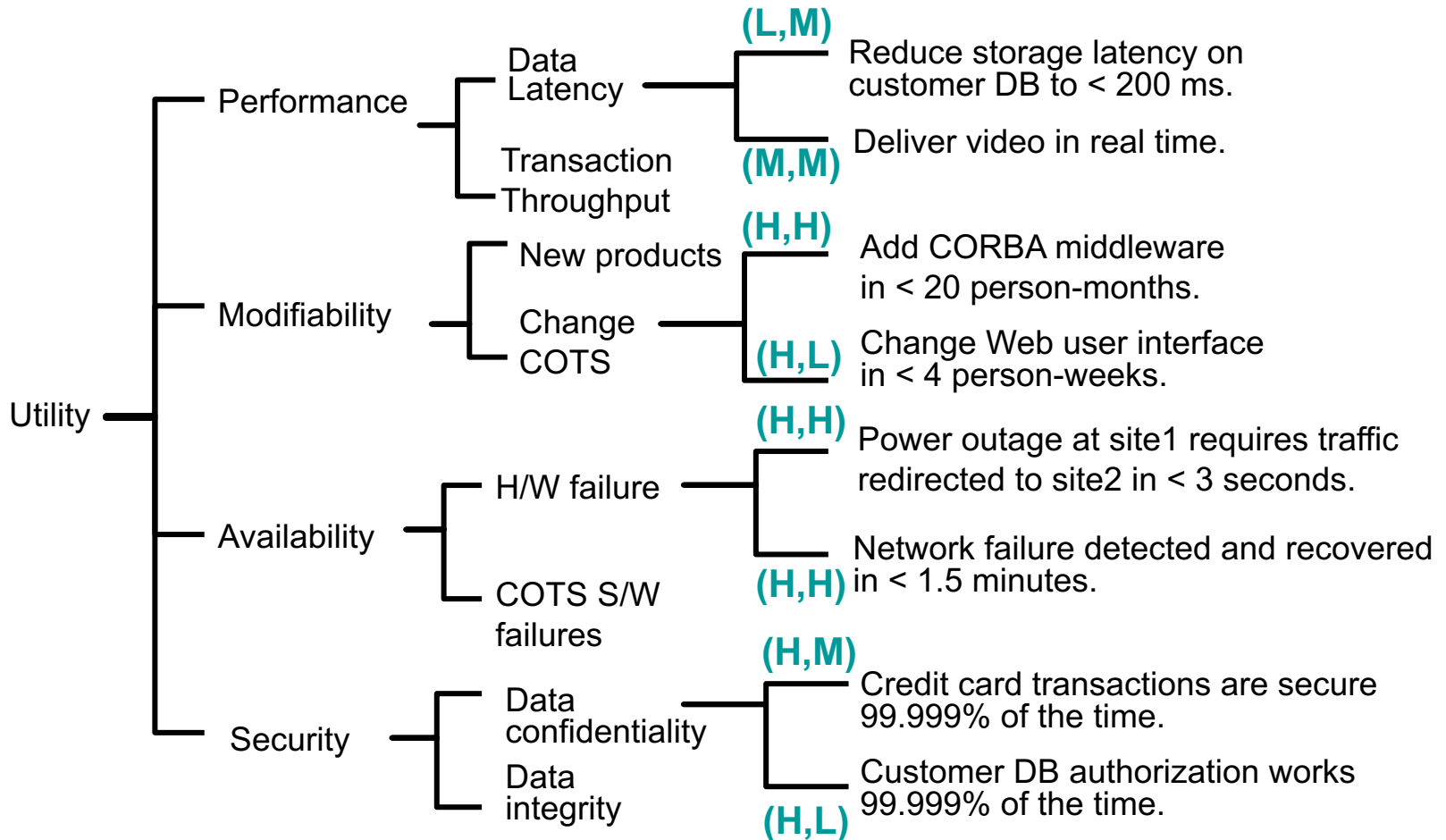
- A utility tree is a top-down vehicle for characterizing and prioritizing the “driving” attribute-specific requirements.
- The driving quality attributes are the high-level nodes (typically performance, modifiability, security, and availability).
- Scenarios are the leaves of the utility tree.

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

SKAO has started work on the utility tree. The evaluation team will extend this prior to the June pre-CDR meeting, and we will revisit this in the session.



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.



Example Scenarios

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.

Scenarios should be as specific as possible.



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

Stimulus

Response

Fault



Availability

Fault masked or repair made

Interoperation Request Arrives



Interoperability

Request correctly handled

Changes Arrive



Modifiability

Changes made, tested, and deployed within time and budget

Events Arrive



Performance

Response generated within time constraints

Attack



Security

System detects, resists, or recovers from attacks

Completion of an increment



Testability

Faults detected



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 will make this prioritization during the June pre-CDR meeting.



Scenario Prioritization – Second Dimension

Architecture Difficulty/Impact

The difficulty dimension is rated high, medium, or low

- High – The architecture team is (or was) very concerned about satisfying this scenario.
 - Architecture driver, required significant tradeoffs, shaped design
 - lack experience, lack expertise, don't know 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.

SDP Architects will make this prioritization during the June pre-CDR meeting.



6. Analyze Architectural Approaches – 1

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.**



6. Analyze Architectural Approaches – 2

The scenario was created for a particular quality attribute

Analysis will focus on that quality attribute, but there may be questions about other quality attributes, especially if those quality attributes and concerns were identified as high priority

Analysis completion criteria:

- Architect has traced from stimulus to response, or we have identified risks that the response will not be achieved
- No more questions from the evaluation team

First scenario will take the longest (60-90 minutes). Things usually move faster as we complete more analyses.



Quality Attribute Questions

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

Performance

- How are priorities assigned to processes?
- 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 **tradeoff point** is a property that affects more than one attribute and is a sensitivity point for more than one attribute.

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. **In practice, most sensitivity points are trivial, and we don't identify many of these.**



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

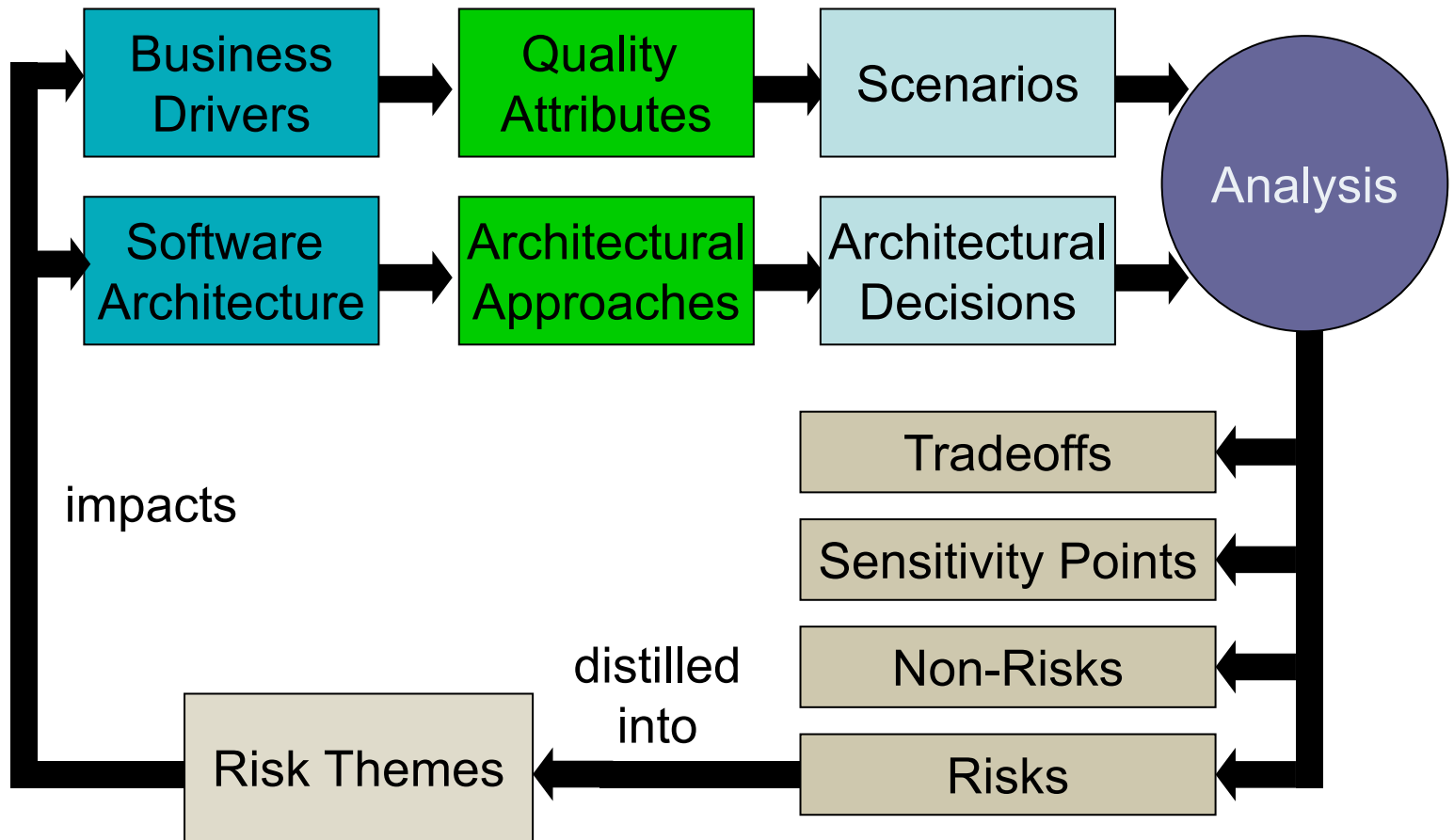
“The average number of person-days of effort it takes to maintain a system might be sensitive to the degree of encapsulation of its communication protocols and file formats.”

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.”



Conceptual Flow of the ATAM



The Final Report

The evaluation team will typically create a final report that includes

- an executive summary
- a description of the ATAM
- a description of the business drivers and architecture
- the Phase 1 scenarios and the utility tree
- the Phase 1 analysis: architectural approaches, decisions, risks, sensitivities, tradeoffs, and non-risks
- the risk themes
- the next steps

For the SDP evaluation, the final report will be sections in the pre-CDR report.



Outline

Why Analyze an Architecture?

The ATAM

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

Rules of Engagement

Summary



ATAM Principles

Transparency

- Live scribing of proceedings
- All findings are created and documented during the working sessions

Evidence-based findings, not subjective

Stakeholder-centric

- Quality attributes
- Scenarios
- Prioritization

Reasoning chain: qualities → scenarios → decisions → findings



Rules of Engagement

To complete everything and get the most from the evaluation, we will need to follow some rules of engagement:

- Your participation is essential.
- Feel free to ask questions at any time.
- Discussion is good, but we may need to cut some discussions short in the interest of time.
- Please stay focused and limit side discussions.



Summary

The ATAM is

- a method for evaluating an architecture with respect to multiple quality attributes
- an effective strategy for discovering the consequences of architectural decisions
- a method for identifying trends, not for performing precise analyses



For Additional Information

ATAM Process:

John Klein jklein@sei.cmu.edu

Mary Popeck mpopeck@sei.cmu.edu

Pre-CDR Process:

Nick Rees N.Rees@skatelescope.org

Marco Bartolini M.Bartolini@skatelescope.org

