Software Architecture Evaluation

March 2014

Ying SHEN

SSE, Tongji University

Objectives of this lecture

This lecture will enable students to

- understand the steps of ATAM method

Why Analyze Software Architectures?

All design involves tradeoff in system qualities

- System qualities are largely dependent on architectural decisions
- Promoting one quality often comes at the expense of another quality

A software architecture is the earliest life-cycle artifact that embodies significant design decisions: choices and tradeoffs.

- Choices are easy to make, but hard to change once the system is implemented

The ATAM

SEI has developed the Architecture Tradeoff Analysis Method (ATAM) over several years.

The purpose of ATAM is to:

- assess the consequences of architectural decisions in light of quality attribute requirements and business goals
- discover risks created by architectural decisions in the system and software architectures of systems

Purpose of ATAM

We need a method in which the right questions are asked *early* to:

- Discover risks - alternatives that might create future problems in some quality attribute
- Discover non-risks - decisions that promote qualities that help realize business/mission goals
- Discover sensitivity points - alternatives for which a slight change makes a significant difference in some quality attribute
- Discover tradeoffs - decisions affecting more than one quality attribute

Purpose of ATAM

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

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

Discovered risks can then be made the focus of mitigation activities:

- e.g. further design, further analysis, prototyping.

Surfaced tradeoffs can be explicitly identified and documented.

ATAM Benefits

There are a number of benefits from performing ATAM analyses:

- Clarified quality attribute requirements
- Improved architecture documentation
- Documented basis for architectural decisions
- Identified risks early in the life-cycle
- Increased communication among stakeholders

The results are improved architectures.

Output of ATAM

An outbrief presentation and/or a written report including the major findings of the evaluation:

- A concise presentation of the architecture
- Articulation of the business goals
- Prioritized quality attribute requirements expressed as quality attribute scenarios
- A set of identified risks and non-risks
- A set of risk themes
- Mapping of architectural decisions to quality requirements
- A set of identified sensitivity and tradeoff points

Preconditions for an ATAM

1. Clients must have a Software Architecture
   - Scope/scale must be manageable
   - ATAM *will not work* if the software architecture has not been created yet
   - ATAM team members will review architectural artifacts, and may help refine documentation
   - Architect must prepare an architecture presentation
2. Clients must prepare a business/mission goals presentation
3. ATAM will review architecture artifacts, presentations, and read ahead material to become familiar with domain

Evaluation Team

Each ATAM team consists of a leader and at least three other team members

- Domain expertise is not necessary
- ATAM team members must be experienced architects
- ATAM leaders must have EXCELLENT communication and facilitation skills

The ATAM team members fill multiple roles during the course of the evaluation.

Evaluation Team Roles

Team Leader

- sets up the evaluation;
- coordinates with client;
- establishes evaluation contract;
- forms evaluation team
- sees that final report is produced and delivered.

Evaluation Leader

- Runs evaluation;
- facilitates elicitation of scenarios;
- administers scenario selection/prioritization process;
- facilitates evaluation of scenarios against architecture;
- facilitates on-site analysis.

Evaluation Team Roles

Scenario scribe(s)

- writes scenarios on flip-charts or whiteboards;
- capture agreed-on wording of each scenario

Proceedings scribe

- captures proceedings (raw scenarios, other issues, and resolution of each scenario) on a laptop computer
- generate a printed list of adopted scenarios to all participants

Questioner(s)

- raise issues of architectural interests, usually related to the quality attributes in which he/she has expertise

Basic Rules for ATAM Team Members

Keep the process moving!

Ask questions

Propose scenarios

Write down exactly what stakeholders say; do not “edit” their words!

ATAM 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

Phase I

7. Brainstorm and prioritize scenarios
8. Analyze architectural approaches

Phase II

9. Present results

1. Present the ATAM

Evaluation Team presents an overview of the ATAM including:

- ATAM steps in brief
- Techniques
  - utility tree generation
  - architecture elicitation and analysis
  - scenario brainstorming/mapping
- Outputs
  - architectural approaches
  - utility tree
  - scenarios
  - risks and “non-risks”
  - sensitivity points and tradeoffs

2. Present Business Drivers

ATAM customer representative describes the system’s business drivers including:

- The system’s most important functions
- Any relevant technical, managerial, economic, or political constraints
- Business goals and contexts
- The major stakeholders
- Architectural drivers

3. Present Architecture

Architect presents an overview of the architecture including:

- Technical constraints such as an OS, hardware, or middle-ware prescribed for use
- Other systems with which the system must interact
- Architectural approaches/styles used to address quality attribute requirements

Evaluation team begins probing for and capturing risks.

- Ask for clarification based on examination of documentation and knowledge of business drivers

ATAM 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

Phase I

7. Brainstorm and prioritize scenarios
8. Analyze architectural approaches

Phase II

9. Present results

4. Identify Architectural Approaches

Start to identify places in the architecture that are key for realizing quality attribute goals.

Identify any predominant architectural approaches.

Examples:

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

5. Generate Quality Attribute Utility Tree

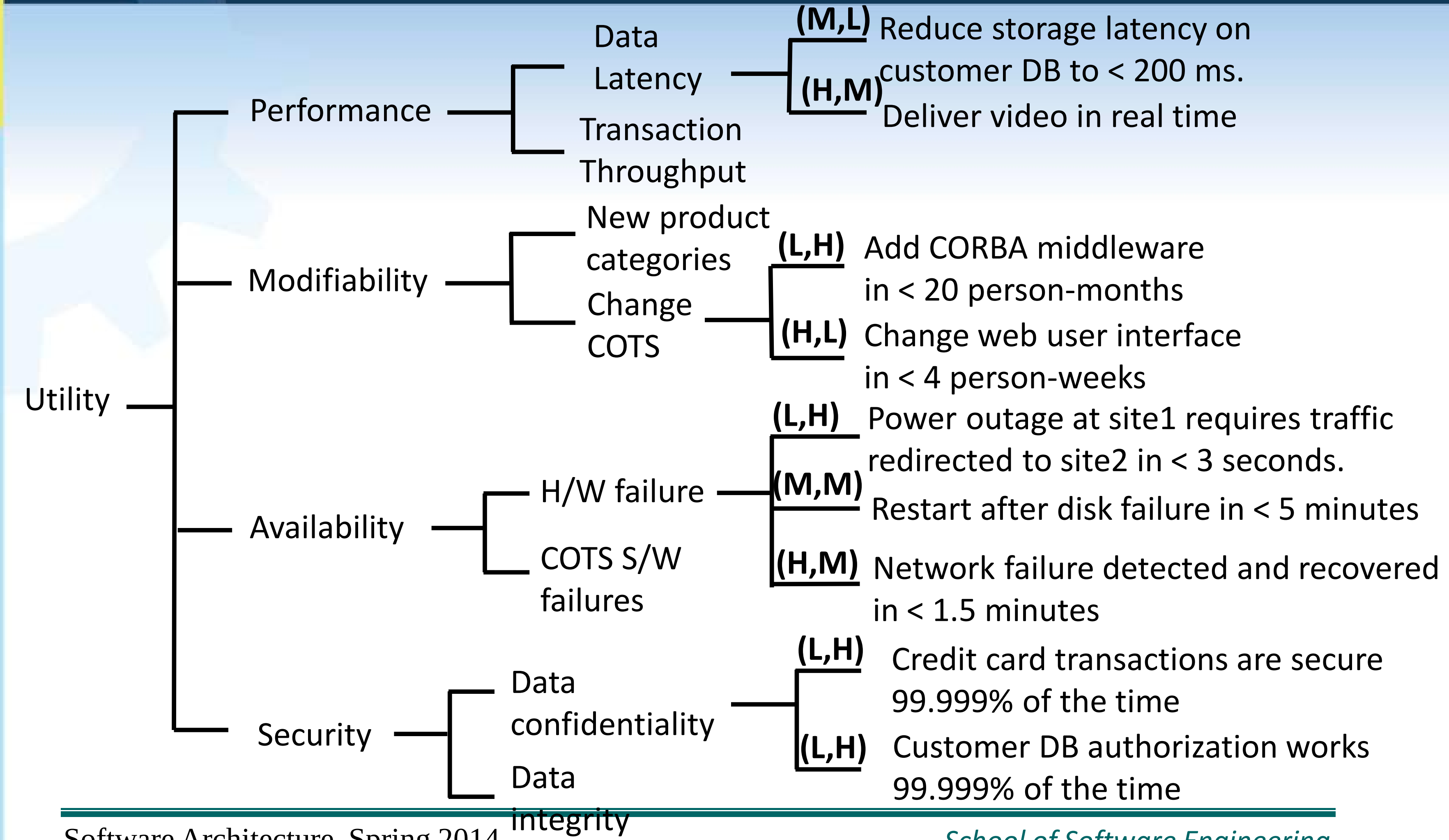
Identify, prioritize, and refine the most important quality attribute goals by building a utility tree.

- A utility tree is a top-down vehicle for characterizing the “driving” attribute-specific requirements
- Select the most important quality goals to be 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.

- High/Medium/Low business value
- High/Medium/Low architectural impact

Utility Tree Construction



Utility

Performance

Data Latency

(M,L) Reduce storage latency on customer DB to < 200 ms.

(H,M) Deliver video in real time

Transaction Throughput

Modifiability

New product categories

Change COTS

(L,H) Add CORBA middleware in < 20 person-months

(H,L) Change web user interface in < 4 person-weeks

Availability

H/W failure

(L,H) Power outage at site1 requires traffic redirected to site2 in < 3 seconds.

(M,M) Restart after disk failure in < 5 minutes

(H,M) Network failure detected and recovered in < 1.5 minutes

COTS S/W failures

Security

Data confidentiality

(L,H) Credit card transactions are secure 99.999% of the time

(L,H) Customer DB authorization works 99.999% of the time

Data integrity

Scenarios

Scenarios are used to

- Represent *stakeholders*' interests
- Understand quality attribute requirements

Scenarios should cover a range of

- Anticipated uses of (use case scenarios),
- Anticipated changes to (growth scenarios), or
- Unanticipated stresses (exploratory scenarios) to the System.

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

Stimuli, Environment, **Responses**

Use Case Scenario

- *Remote user requests a database report via the Web* **during 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.

6. Analyze Architectural Approaches

The architect is asked to explain how architecture supports the highest-ranked scenarios.

Evaluation Team probes architectural approaches to identify risks.

- Identify the approaches that pertain to the highest priority quality attribute requirements
- Generate quality-attribute specific questions for highest priority quality attribute requirement
- Ask quality-attribute specific questions
- Identify and record risks and non-risks, sensitivity points and tradeoffs

Quality Attribute Questions

Quality attribute questions probe styles to elicit architectural decisions which bear on QA requirements.

For well known approaches

- How the architect overcame known weaknesses in the approach or
- how the architect gained assurance that the approach suffice?

The goal is to convince evaluation team that the approach is appropriate for meeting QA requirements.

Risks, Non-Risks, Sensitivity, and Tradeoff

Example Risks

- Rules for writing business logic tier of your 3-tier style are not clearly articulated.
- There is no way of detecting the “live” failure of a critical component.

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.

Sensitivities and Tradeoffs

Example Sensitivity

- Changing the timing scheme from a harmonic framework to a non-harmonic framework would be easy, but due to implied timing dependencies, there would be far reaching impacts to other modules.

Example Tradeoffs

- In order to achieve the required level of performance in the discrete event generation component, assembly language had to be used thereby reducing the portability of this component.

ATAM 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

Phase I

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

Phase II

7. Brainstorm and Prioritize Scenarios

Stakeholders generate scenarios using a facilitated brainstorming process.

- Scenarios at the leaves of the utility tree serve as examples to facilitate the step.
- The new scenarios are added to the utility tree

Each stakeholder is allocated a number of votes roughly equal to 0.3 x #scenarios.

8. Analyze Architectural Approaches

Identify the architectural approaches impacted by the scenarios generated in the previous step.

This step continues the analysis started in step 6 using the new scenarios.

Continue identifying risks and non-risks.

Continue annotating architectural information.

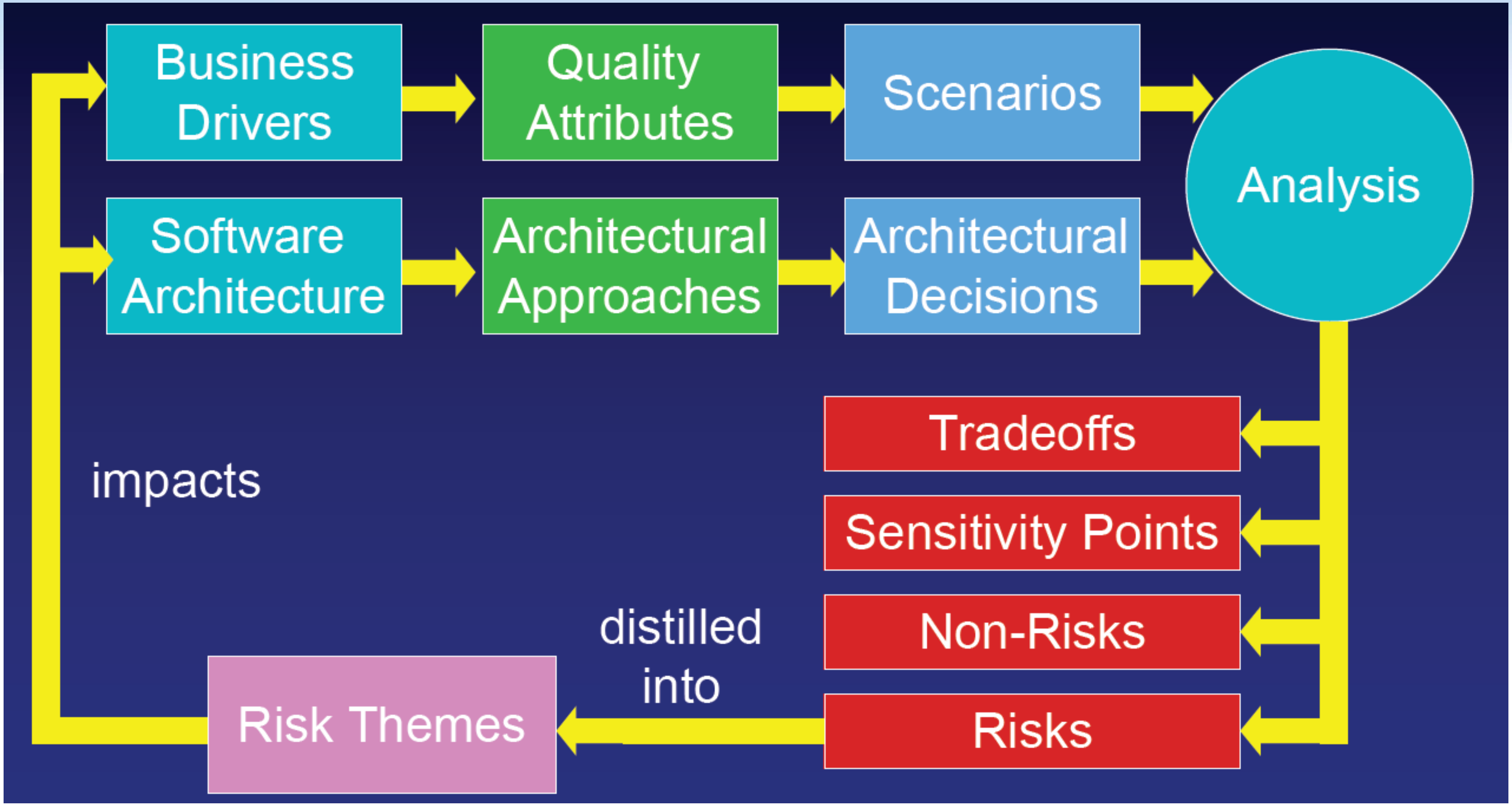
9. Present Results

Recapitulate steps of the ATAM

Present ATAM outputs

- architectural approaches
- utility tree
- scenarios
- risks and “non-risks”
- sensitivity points and tradeoffs

Conceptual Flow of ATAM

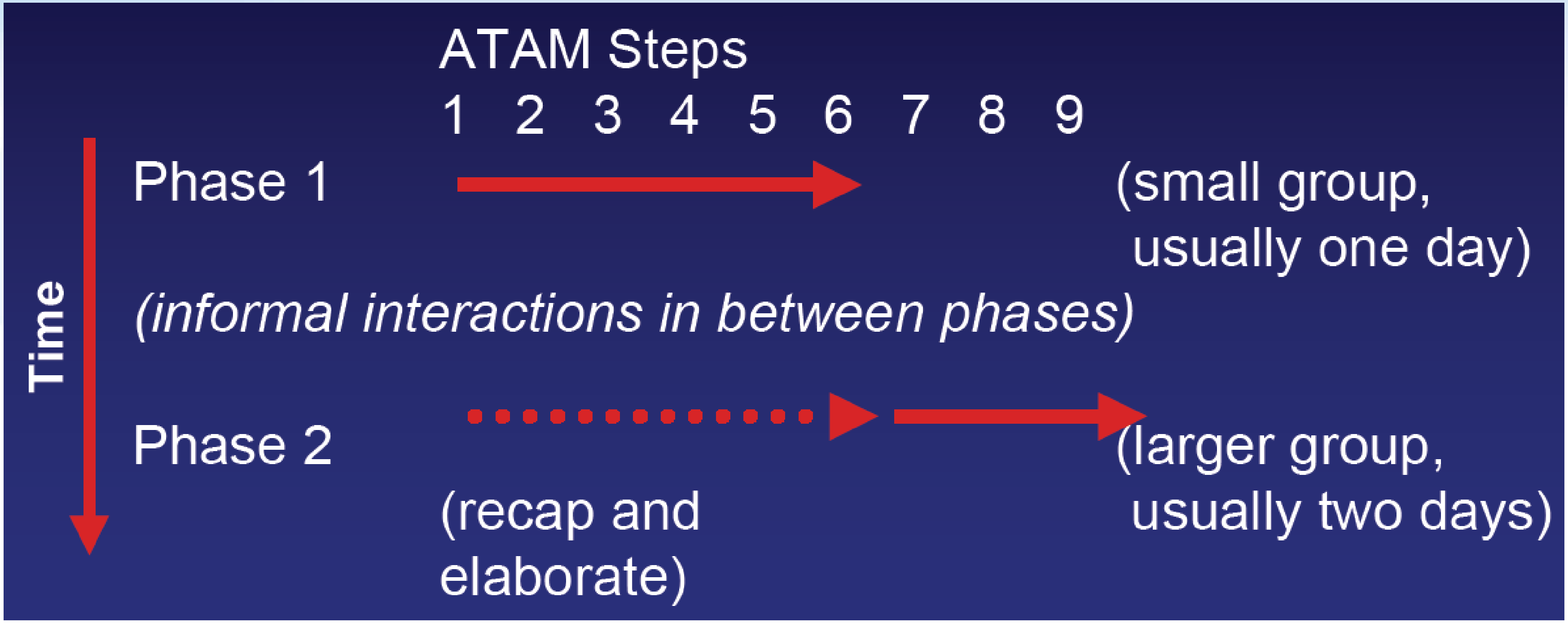


ATAM Nominal Phases

ATAM evaluations are often conducted in two stages or phases:

- During phase 1 the architect describes the quality attribute goals and how the architecture meets these goals
- During phase 2 we determine if a larger group of stakeholders agrees with the goals and the results

ATAM Nominal Phases



ATAM Steps

1 2 3 4 5 6 7 8 9

Time

Phase 1

(small group, usually one day)

(informal interactions in between phases)

Phase 2

(recap and elaborate)

(larger group, usually two days)

When to use ATAM

Academically, the time to use ATAM is right after *the architecture has been specified* when there is little or no code.

However, in practice, ATAM has been very effective in the following situations:

- Evaluating alternative candidate architectures
- Evaluating existing systems prior to committing to major upgrades
- Deciding between upgrade or replace

ATAM Summary

The ATAM is a method for evaluating an architecture with respect to multiple quality attributes.

- effective strategy for discovering the consequences of architectural decisions.

The ATAM:

- can be done early; can be done on legacy systems
- is inexpensive
- builds stakeholder confidence and buy-in
- The key to the method is looking for trends, not in making precise analyses.

ATAM Summary

The ATAM relies critically on

- Appropriate preparation by the customer
- Clearly-articulated quality attribute requirements
- Active stakeholder participation
- Active participation by the architect
- Familiarity with architectural approaches, styles and analytic models

The End