COMP 6471
Software Design Methodologies

Fall 2011

Dr Greg Butler

http://www.cs.concordia.ca/~gregb/home/comp6471-fall2011.html

ATAM
Architecture Trade-Off Analysis Method

The purpose of the ATAM is:

- to assess the consequences of architectural decision alternatives in light of quality attribute requirements.

ATAM: Why Analyze an Architecture?

- All design involves tradeoffs

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

ATAM: Purpose

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 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

ATAM: Purpose

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.

ATAM: Benefits

There are a number of benefits from performing ATAM analyses:

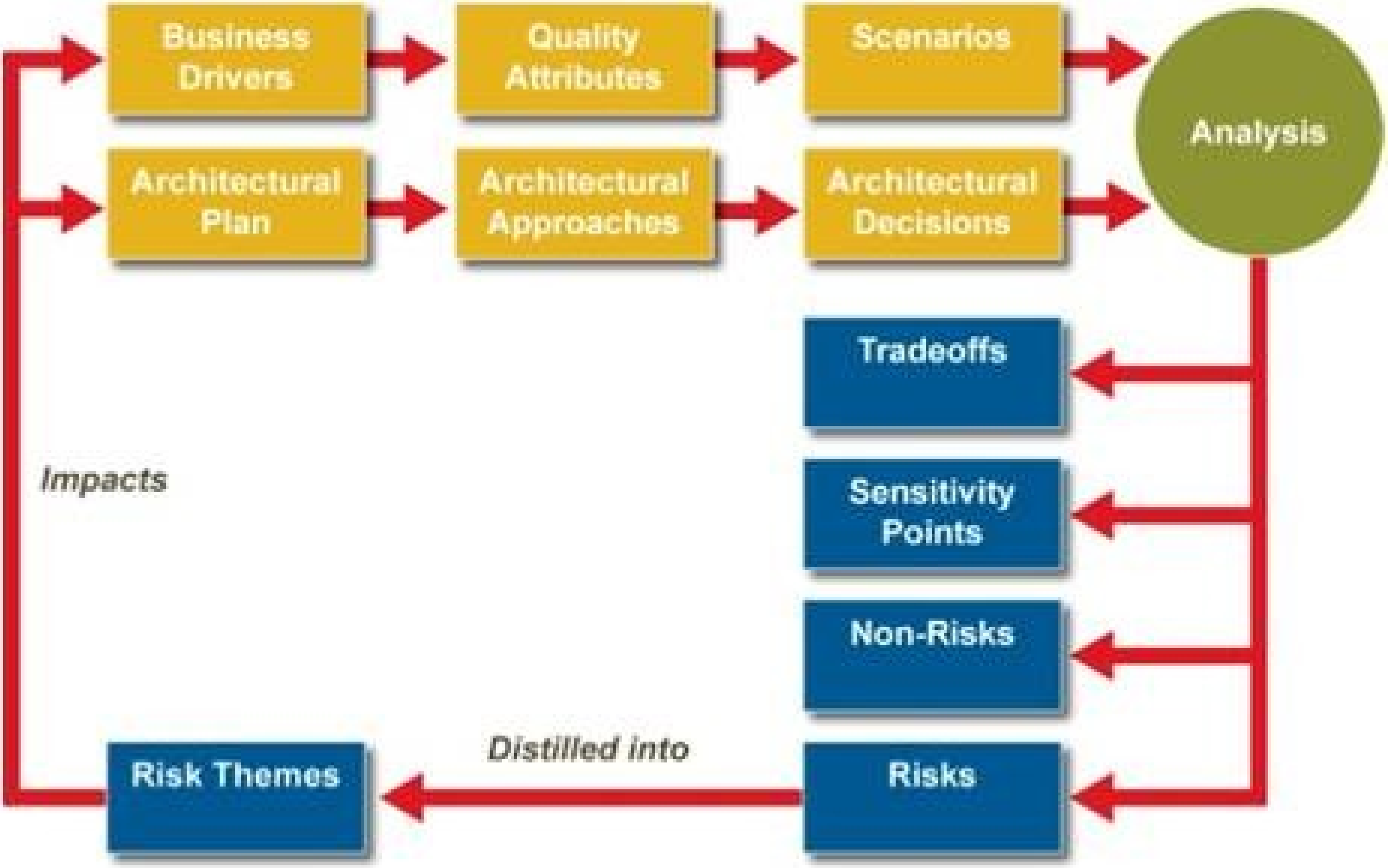
- Clarified quality attribute requirements
- Improved architecture documentation
- Documented basis for architectural decisions
- Identified risks early in the life-cycle
- Increased communication among stakeholders

The results are improved architectures.

ATAM Steps

1. Present the ATAM
2. Present business drivers
3. Present architecture
4. Identify architectural styles
5. Generate quality attribute utility tree
6. Elicit and analyze architectural styles
7. Brainstorm and prioritize scenarios
8. Analyse architectural approaches (using scenarios)
9. Present out-brief and/or write report

ATAM


Business Drivers
Quality Attributes
Scenarios
Analysis
Architectural Plan
Architectural Approaches
Architectural Decisions
Tradeoffs
Sensitivity Points
Non-Risks
Impacts
Distilled into
Risk Themes
Risks

ATAM: 1. Present the ATAM

Evaluation Team presents an overview of the ATAM including:

ATAM steps in brief

techniques
- utility tree generation
- style-based elicitation/analysis
- scenario brainstorming/mapping

outputs
- scenarios
- architectural styles
- quality attribute questions
- risks and non-risks
- utility tree

ATAM: 2. Present Business Drivers

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

- 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 most central to the system’s success

ATAM: 3. Present the 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 used to meet quality attribute requirements

Evaluation team begins probing for:

- risks
- architectural styles

ATAM: 4. identify Architectural Styles

High-level overview of architecture is completed by itemizing architectural styles found in the architecture

ATAM:
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 an AHP (analytic hierarchy process)-like model of the “driving” attribute-specific requirements

Typically

performance, modifiability, security, and availability

are the high-level nodes

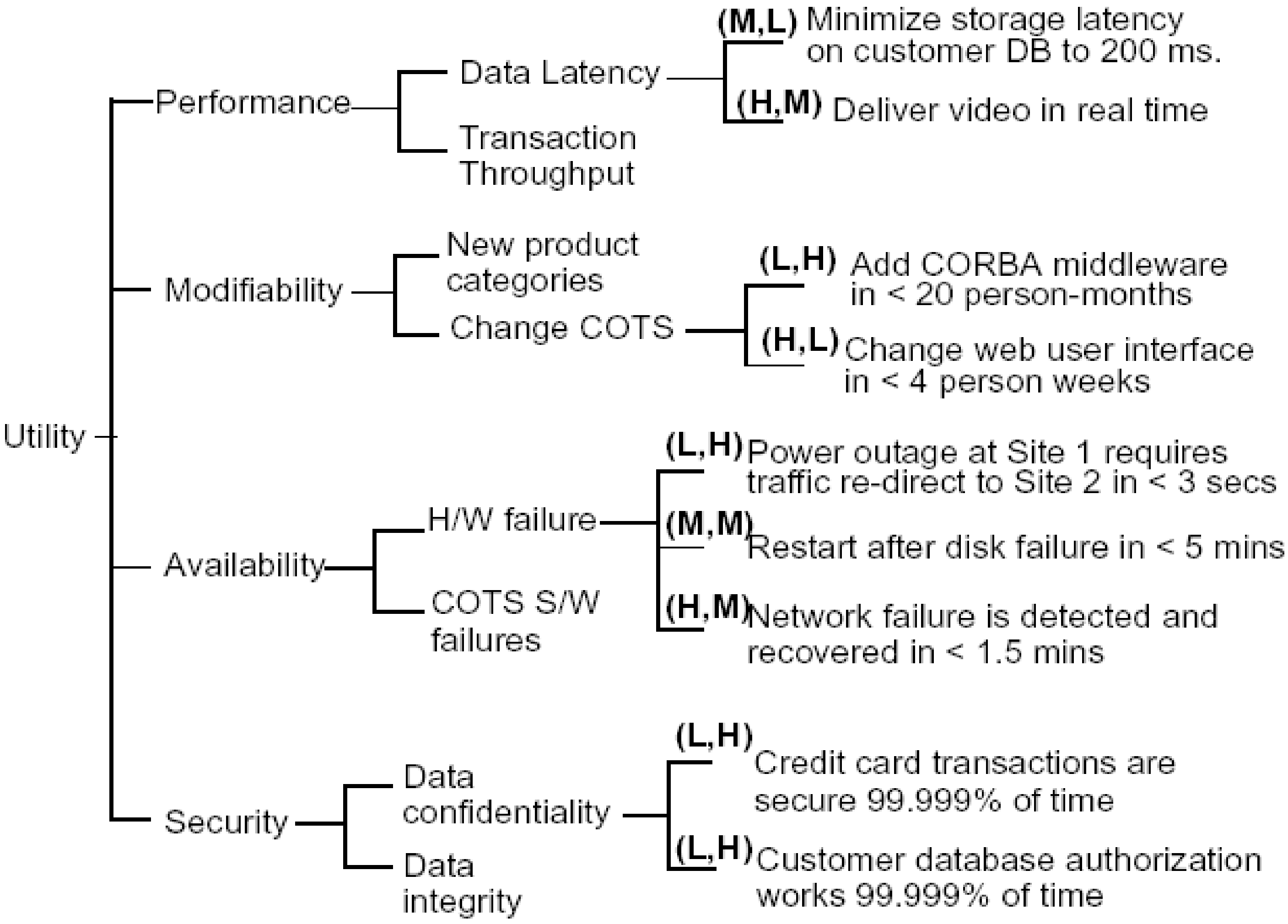
scenarios are leaves

of utility tree

Output:

a prioritization of specific quality attribute requirements.

ATAM Utility Tree



- Utility
  - Performance
    - Data Latency
      - (M,L) Minimize 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 Site 1 requires traffic re-direct to Site 2 in < 3 secs
      - (M,M) Restart after disk failure in < 5 mins
      - (H,M) Network failure is detected and recovered in < 1.5 mins
    - COTS S/W failures
  - Security
    - Data confidentiality
      - (L,H) Credit card transactions are secure 99.999% of time
      - (L,H) Customer database authorization works 99.999% of time
    - Data integrity

(Importance,Risk) L=low, M=medium, H=high

Step 5- 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.**

Step 5 – Scenario examples

- **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.**

ATAM:
6. Elicit and Analyze Architecture Styles

Evaluation Team probes architectural styles from the point of view of specific quality attributes to identify risks.

Identify the styles which 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

ATAM: Risks and Non-Risks

Risks are potentially problematic architectural decisions

Non-risks are good decisions relying on implicit assumptions.

Risk and non-risk constituents

- architectural decision
- quality attribute requirement
- rationale

Sensitivity points are candidate risks and risks are candidate tradeoff points.

Example risk: Rules for writing business logic modules in the second tier of your 3-tier style are not clearly articulated. This could result in replication of functionality thereby compromising modifiability of the third tier.

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 6: Sensitivity & Tradeoffs

- Sensitivity – A property of a component that is critical to success of system.
  - The number of simultaneous database clients will affect the number of transaction a database can process per second. This assignment is a sensitivity point for the performance
  - Keeping a backup database affects reliability
  - Power of encryption (Security) sensitive to number of bits of the key

- Tradeoff point- A property that affects more than one attribute or sensitivity point.
  - 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.
  - Keeping the backup database affects performance also so it's a trade-off between reliability and performance

Scenario: S12 (Detect and recover from HW failure of main switch.)

Attribute: Availability

Environment: normal operations

Stimulus: CPU failure

Response: 0.999999 availability of switch

Architectural decisions	**Risk**	**Sensitivity**	**Tradeoff**
Backup CPU(s)	R8	S2	
No backup Data Channel	R9	S3	T3
Watchdog		S4	
Heartbeat		S5	
Failover routing		S6	

Reasoning:

- ensures no common mode failure by using different hardware and operating system (see Risk 8)

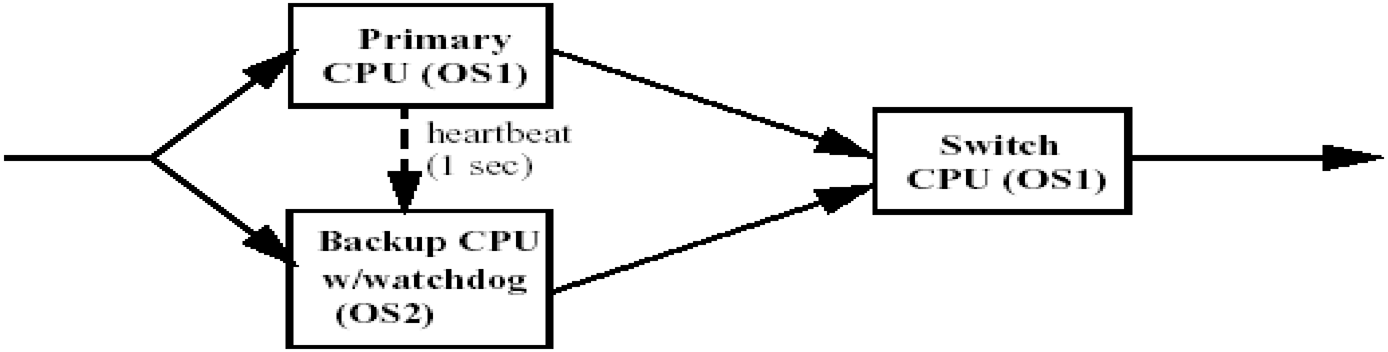
- worst-case rollover is accomplished in 4 seconds as computing state takes …

- guaranteed to detect failure with 2 seconds based on rates of heartbeat and watchdog …

- watchdog is simple and proven reliable

- availability requirement might be at risk due to lack of backup data channel … (see Risk 9)

Architecture diagram:


Primary CPU (OS1)
heartbeat (1 sec)
Backup CPU w/watchdog (OS2)
Switch CPU (OS1)