Analyzing Software Architectures

EEC 625
Lecture #15

March 20, 2006

Architecture Evaluation

Why? Economic value; you want to see what the value of an architectural choice is, before committing to it.

When? As early as possible, but continuous evaluation is also needed.

Cost. Be cognizant of this!

Benefits of Architecture Evaluation

- Financial
- Forced preparation for review
- Captured rationale
- Early detection of problems
- Validation of requirements
- Improved architectures

Preconditions for a Successful Evaluation

- Clearly articulated goals and requirements
- Controlled scope
- Cost-effectiveness
- Key personnel availability
- Competent evaluation team
- Managed expectations

ATAM

Architecture Tradeoff Analysis Method

Looks at how well an architecture satisfies particular quality goals.

Provides insight into how quality goals trade off.

ATAM Participants

- Project decision makers
- Architecture stakeholders
- Evaluation team
  - Team leader
  - Evaluation leader
  - Scenario scribe
  - Proceedings scribe
  - Timekeeper
  - Process observer
  - Process enforcer
  - Questioner

ATAM Outputs

- Concise presentation of the architecture
- Articulation of business goals
- Quality requirements in terms of a collection of scenarios
- Mapping of architectural decisions to quality requirements
- Set of identified sensitivity and tradeoff points
- Set of risks and non-risks
- Set of risk themes

Secondary Outputs

"Better-than-before" representations of the architecture.

Scenarios can guide evolution.

Statement of rationale.

Sense of community – intangible.

ATAM Phases and Characteristics

Phase	Activity	Participants	Duration
0	Partnership and preparation	Evaluation team leadership, key decision makers	Few weeks
1	Evaluation	Evaluation team, decision makers	1 day, followed by 2-3 week gap
2	Evaluation (contd.)	Evaluation team, decision makers & stakeholders	2 days
3	Follow-up	Evaluation team and evaluation client	1 week

ATAM Evaluation Steps – Phase 1

1. Present the ATAM – evaluation leader
2. Present business drivers – everyone
3. Present architecture – lead architect
4. Identify architectural approaches – evaluation team
5. Generate quality attribute utility tree – evaluation team and decision makers
6. Analyze architectural approaches – evaluation team

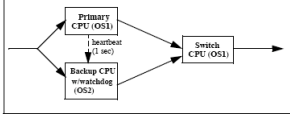
Architectural Approach Analysis

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:



ATAM Evaluation Steps – Phase 2

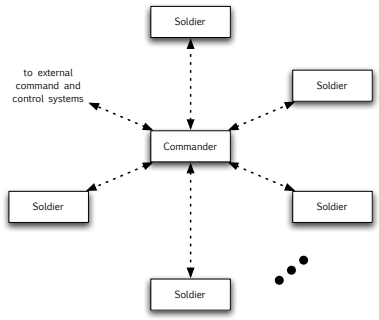
7. Brainstorm and prioritize scenarios
8. Analyze architectural approaches
9. Present results

ATAM Example: BCS

Battlefield Control System.

This system is to be used by army battalions to control the movement, strategy, and operations of troops in real time in the battlefield. This system is currently being built by a contractor, based upon government furnished requirements.

BCS: Hardware View



BCS: Identifying Architectural Approaches

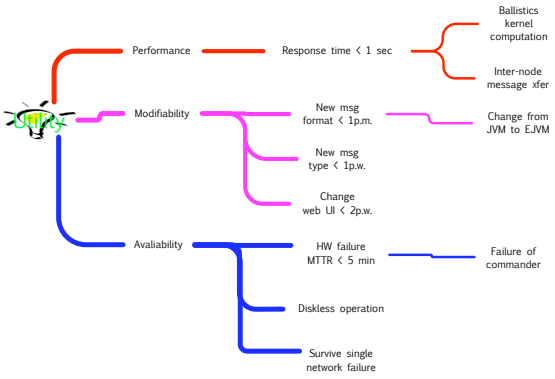
Loosely organized as client server.

For availability, a backup commander approach was described.

For modifiability, standard subsystem organizational patterns were described.

For performance, an independent communicating components approach was described.

BCS: Utility Tree



BCS: Analyzing Architectural Approaches

System quality (Q_S) is a function f of the quality of the modifiability (Q_M), the availability (Q_A), and the performance (Q_P):

$$Q_S = f(Q_M, Q_A, Q_P)$$

BCS: Avaliability

$Q_A = g$(the fraction of time that the system is working)

System is working when `Commander` is working.

When `Commander` fails, system fails. But a `Soldier` can be turned into a `Commander` – a client can be converted to a server.

Repair time is the time taken to convert a `Soldier` node into the `Commander` node.

BCS: Avaliability

$$Q_A = g(\lambda_C, \mu_C, \mu_B)$$

λ_C – failure rate of the `Commander`

μ_C – repair rate of the `Commander`

μ_B – repair rate of the backup. This is zero; no provision for creating additional backups

BCS Backups

Backup `Commander` is a `Soldier` who is monitoring all messages to and from `Commander` — completely synchronized.

Going to multiple backups is not trivial.

Alternate architecture proposed:

n acknowledging backups, and m passive backups.

$$Q_A = g(n, m)$$

But this impacts performance – tradeoff point.

BCS: Performance

Communication between `Commander` and `Soldiers` on 9600 baud modem.

Experiment situations:

1. Periodic data updates to `Commander`
2. Turning a `Soldier` node into a backup
3. Doubling number of missions

BCS: Switchover

Downloading mission plans:
280 Kbits / 9.6 Kbits/second @ 29.17 seconds

Updates to environmental database:
66 Kbits / 9.6 Kbits/second @ 6.88 seconds

Acquiring issued orders:
24 Soldiers * (18 Kbits/9.6 Kbits/second) = 45.0 seconds

Acquiring Soldier locations and status:
24 Soldiers * (12 Kbits/9.6 Kbits/second) = 30.0 seconds

Acquiring inventories:
24 Soldiers * (42 Kbits/9.6 Kbits/second) = 105.0 seconds

Total @ 216.05 seconds for Soldier to become backup

BCS: Performance

Backups require periodic updates from the `Commander`

These messages average 59,000 Kbits every 10 minutes.

Requirement: 99.67 bits/sec (1% of the overall comm. bandwidth)

$$Q_P = h(n, m, CO)$$

number of acknowledging backups – n
passive backups – m
other communication overhead – CO

BCS: Key ATAM Outputs

Sensitivities and Tradeoffs

Key tradeoff: communications load

The overall performance and availability of the system is highly sensitive to the latency of the (limited and shared) communications channel, as controlled by the parameters n and m.

BCS: Key ATAM Outputs

Key Architectural Risks

Communication pattern between `Commander` and backup is distinctive. So this is prone to attacks — security could be compromised.

This flaw is only revealed when examining architecture from the perspective of multiple quality attributes simultaneously.

ATAM Summary

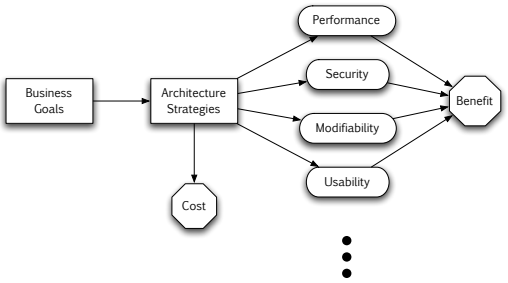
Key goal is to determine how well the architecture has been designed with respect to the quality attributes that the stakeholders have deemed important.

Tradeoffs are important.

ATAM does not deal with economics

CBAM

Cost Benefit Analysis Method



CBAM

CBAM does not make decisions for stakeholders.

Aids in elicitation and documentation of costs, benefits, and uncertainty of a "portfolio" of architectural investments.

Provides framework to calculate benefit (utility) and return on investment (ROI).

Basis for CBAM

- Consider collection of scenarios
- Assign utility to values of projected quality attribute responses
- Consider strategy that will lead to these responses
- Each strategy has a cost
- Strategy may also have side effects
- Combine the overall utility with overall cost to calculate ROI

Determining Benefit and Normalization

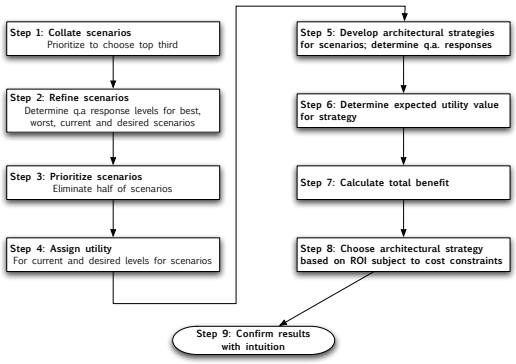
For each architectural strategy i, we can calculate a benefit B_i:

$$B_i = \sum_j (b_{i,j} \times W_j)$$

- $b_{i,j}$ — benefit accrued to strategy i due to its effect on scenario j
- W_j — weight of scenario j

$$ROI_i = \frac{B_i}{C_i}$$

CBAM Steps


Step 1: Collate scenarios
Prioritize to choose top third
Step 2: Refine scenarios
Determine q.a response levels for best, worst, current and desired scenarios
Step 3: Prioritize scenarios
Eliminate half of scenarios
Step 4: Assign utility
For current and desired levels for scenarios
Step 5: Develop architectural strategies for scenarios; determine q.a. responses
Step 6: Determine expected utility value for strategy
Step 7: Calculate total benefit
Step 8: Choose architectural strategy based on ROI subject to cost constraints
Step 9: Confirm results with intuition