Architecture Analysis

Boeing’s Experiences using the SEI ATAM® and QAW Processes

Presented by: Don O’Connell
Associate Technical Fellow
Components of Our Analysis Tool Set

- **SEI ATAM<sup>SM</sup>** - qualitatively assesses architectural decisions in light of quality attribute requirements and business goals.
- **SEI’s QAW** - gathers key stakeholders together to better describe, qualify and quantify desired architectural attributes for a system.
- Boeing’s Predictive Analysis, Performance Method (PAPM) - process and tool for quantitative, predictive performance analysis of mission computing systems. Focus is on resource utilization and end to end latencies.
- Boeing’s Predictive Analysis, Availability Method (PAAM)- process and tool to predict and trade availability, fault detection, fault isolation, redundancy and reconfiguration characteristics of software intensive systems.
- Boeing’s Predictive Analysis, Security Method (PASM) - method to analyze security and information assurance architectural qualitative and quantitative properties.
- Boeing’s Reused Architectural Component Method (RACM) - analyzes COTS and reuse components and technologies against their cost, risk and appropriateness for an architecture.
- The charts that follow summarize our ATAM and QAW experiences.
**ATAM\textsuperscript{SM} evaluations are conducted in four phases.**

- **Phase 0: Partnership and Preparation**
  - Duration: varies
  - Meeting: primarily phone, email

- **Phase 1: Initial Evaluation**
  - Duration: 1.5 - 2 days each for Phase 1 and Phase 2
  - Meeting: typically conducted at customer site

- **Phase 2: Complete Evaluation**
  - Duration: varies
  - Meeting: primarily phone, email

- **Phase 3: Follow-up**
  - Duration: varies
  - Meeting: primarily phone, email

- Red = talking points

- **Key Points**
  - One Key is the Go/noGo decision
  - Key to scope any additional work
  - We are there to help, not criticize
  - Try to get 2 business driver pitches, from both sides
  - Note architecture changes from phase 1
  - Scenarios are key, spend extra time gathering, homework
  - Do the report right away, else it loses value
  - Follow up with arch help

- Note architecture changes from phase 1
- Scenarios are key, spend extra time gathering, homework
- Do the report right away, else it loses value
- Follow up with arch help

---

**Copyright © 2006 Boeing. All rights reserved.**
ATAMSM Steps

Boeing Technology | Phantom Works

<table>
<thead>
<tr>
<th>Phase 1</th>
<th>Phase 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>•1. Present the ATAMSM</td>
<td></td>
</tr>
<tr>
<td>•2. Present business drivers</td>
<td></td>
</tr>
<tr>
<td>•3. Present architecture</td>
<td></td>
</tr>
<tr>
<td>•4. Identify architectural approaches</td>
<td></td>
</tr>
<tr>
<td>•5. Generate quality attribute utility tree</td>
<td></td>
</tr>
<tr>
<td>•6. Analyze architectural approaches</td>
<td></td>
</tr>
<tr>
<td>•7. Brainstorm and prioritize scenarios</td>
<td></td>
</tr>
<tr>
<td>•8. Analyze architectural approaches</td>
<td></td>
</tr>
<tr>
<td>•9. Present results</td>
<td></td>
</tr>
</tbody>
</table>

- Business drivers are key, and usually hard to get a good job
- Usually too much information, stop scenario questions, focus on tactics and patterns. Also, tactic/quality table.
- Often this is already done for us
- Difficult to do a good job each time, perhaps modify technique
- Put architecture on stage, nothing else.
- Do a good job here. Key is look for those few necessary scenarios that are not addressed by arch (gold nuggets). These will yield good value of ATAM.
- Close out dead topics quick. Spend lots of hours analyzing. Delegate writeups.
- What to do with ppt after presented?
Conceptual Flow of ATAM℠

- Make a big one of these charts, put it on the wall and refer to it often.

- Exercise the architecture with scenarios. Within this framework, you can do additional things (affordability, architectural proof, etc)

**Business Drivers**

**Quality Attributes**

**Scenarios**

**Architectural Approaches**

**Architectural Decisions**

**Analysis**

**Tradeoffs**

**Sensitivity Points**

**Non-Risks**

**Risks/Issues**

**Risk Themes**

- Summarize (risks + issues) = risk themes

- Issue definition = out of control of architect

Exercise the architecture with scenarios. Within this framework, you can do additional things (affordability, architectural proof, etc).
At Boeing the effort to perform an ATAM\textsuperscript{SM} has been somewhat consistent. On average:

- 4 evaluators at 70 hours each, 1 facilitator at 120 hours
- 90 hours from architecture team
- 15 stakeholders at 16 hours each

This typically varies by 10-20\% (up or down), depending on the situation
- Except, we had 40 stakeholders in one instance
- If the architecture is not well documented, 2 or 3 times more architecture team hours are required to prepare. This is probably a good thing.

Some projects are too small, or require information more rapidly, and some are too large for the standard process
- A one day ATAM\textsuperscript{SM} approach has been developed for the small project
- A large ATAM\textsuperscript{SM} approach may take 3 to 5 times as much evaluator time for direct interaction on the program
• ATAM can vary in size due to project size

<table>
<thead>
<tr>
<th>Project</th>
<th>Evaluators</th>
<th>Participants</th>
<th>hours/evaluator</th>
<th>hours/facilitator</th>
<th>Architect's hours</th>
<th>Scenarios Analyzed</th>
<th>risk themes</th>
<th>non risk themes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>14</td>
<td>60</td>
<td>100</td>
<td>132</td>
<td>12</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>10</td>
<td>70</td>
<td>100</td>
<td>92</td>
<td>15</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>15</td>
<td>80</td>
<td>120</td>
<td>92</td>
<td>17</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>13</td>
<td>70</td>
<td>100</td>
<td>132</td>
<td>29</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>100</td>
<td>60</td>
<td>9</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>5</td>
<td>22</td>
<td>70</td>
<td>120</td>
<td>80</td>
<td>20</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>5</td>
<td>21</td>
<td>70</td>
<td>120</td>
<td>40</td>
<td>24</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>H</td>
<td>4</td>
<td>14</td>
<td>70</td>
<td>90</td>
<td>100</td>
<td>21</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>average</td>
<td>4.3</td>
<td>14.7</td>
<td>61.4</td>
<td>108.6</td>
<td>89.7</td>
<td>18.0</td>
<td>3.7</td>
<td>1.3</td>
</tr>
</tbody>
</table>
## Selected ATAM℠ Risk Themes by Review

### Boeing Technology | Phantom Works

**E&IT | NCO Thrust**

<table>
<thead>
<tr>
<th>Topic/Program</th>
<th>Performance, Scalability</th>
<th>System Management, Failures</th>
<th>Security, Information Assurance</th>
<th>Product Line Planning, Lead System Integrator</th>
<th>changing or new technologies</th>
<th>Legacy, COTS Integration</th>
<th>New operational procedures</th>
<th>Unknown or new requirements</th>
<th>Software Architecture Undefined</th>
<th>Architectural Tactics</th>
<th>Usability</th>
<th>Safety</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>No performance model, tiny deployment</td>
<td>Make vs Day for system management components</td>
<td>Multi level security</td>
<td>Scope of product line</td>
<td>Infrastructure, MLS, oracle</td>
<td>Impacts to legacy by arch, infrastructure</td>
<td>Handoff, MLS, workload</td>
<td>System management, handoff</td>
<td>Requirements ill defined, scope, illdefined, missing components</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Max load scenario, tiny deployment</td>
<td>Failure modes, smart forwarding</td>
<td>Infrastructure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>System engineering gives unanticipated requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Maximum load, scalability</td>
<td>Failure detection, automatic reconfiguration</td>
<td>Modifiability, variability architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Understanding of Final Requirements</td>
<td>Complexity of using model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>fault management, fault requirements</td>
<td>undefined security architecture and requirements</td>
<td>COTS Quality Attributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>undefined components, undefined behavior</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Ramps illdefined, Design illdefined</td>
<td>Impact of new infrastructure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Process**

- PAPM
- PAAM
- PASM
- RACM
- RACM
- OAW
- ArchDoc
- PRO-

*Process = ways to cure the ills*
### Scenario vs Quality Attribute Statistics

**Scenarios analyzed against quality attributes per ATAM**

<table>
<thead>
<tr>
<th>Scenarios analyzed</th>
<th>performance (scalability)</th>
<th>security (information Assurance)</th>
<th>Affordability (schedule, cost)</th>
<th>availability (reliability, robustness)</th>
<th>interoperability</th>
<th>modifiability (expandability)</th>
<th>reconfigurability</th>
<th>usability (capability)</th>
<th>become LSI, product line</th>
<th>testability</th>
<th>Safety</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>3</td>
<td></td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>3</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>total</td>
<td>35</td>
<td>12</td>
<td>32</td>
<td>26</td>
<td>9</td>
<td>34</td>
<td>11</td>
<td>24</td>
<td>4</td>
<td>4</td>
<td>15</td>
</tr>
</tbody>
</table>

*Scenarios typically focus on modifiability, performance and affordability.*

*Little focus on testability, probably a mistake that it ends up like this.*
# Selected Non-Risk Themes by Review

<table>
<thead>
<tr>
<th>Topic, Program</th>
<th>COTS, Reuse Component Planning</th>
<th>modifiability, extensibility, configurability tactics</th>
<th>Performance, scalability tactics</th>
<th>Reporting Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>well planned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>XML based design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td>multi-threaded, isolation of components</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>input files for configurability</td>
<td></td>
<td>well designed reporting</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>numerous modifiability tactics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>COTS database usage</td>
<td>Extensible System Assets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td>layered Arch</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Most programs have good modifiability tactics in place
- More mature programs have more non-risk themes
• Improved Architectural Driver, tactics, views documentation
  • Architecture team typically has to write down what was in their head

• Improved stakeholder awareness of architectural definition and architecture teams awareness of stakeholders key desires
  • All about taking time to communicate, in an ATAMSM framework

• Focused programs on architectural quality attributes and tactics used to address them.
  • Forced the program to look at their system in this light (versus functional architecture)
  • Tactics versus quality attributes table gives a great top down view. Shows gaps.

• Sometimes, ATAMSM had a special focus on cost reduction
  • Risk identification and mitigation reduces rework
  • Cost savings suggestions also can result from the analysis.
Sometimes, ATAM\(^{SM}\) had a special focus on communication
- System engineering and software engineering needed to know more about what each other were doing

Each time, ATAM\(^{SM}\) focused programs on key risks and issues
- Risk identification and program acceptance happens first, then mitigation. These are 2 distinct steps.
- Risk mitigation strategies, especially if it is our expertise
- Assist architecture teams with risk mitigation (performance, availability, infrastructure)

Specific Risks identified and since mitigated
- Performance risks identified, then mitigated
- Security architecture risks identified, then mitigated
- Infrastructural risks identified, then mitigated
- Others
Conceptual Flow of a QAW

- Make a big one, put it on the wall and refer to it.

- Insure that participation of heavyweight stakeholders occurs. This really does help shape the architecture, sends the architecture in a particular direction.

- Prioritization is important here. Especially, to lower priority of unnecessary, hard scenarios.

- A scoring of Quality Attributes can optionally be done. Result, priorities of quality attributes.
QAW Benefits, Result Observations

We have performed 2 QAWs, and used 2 others for ATAM\textsuperscript{SM} reviews
  - We had good success on 1 of our 2 QAWs
    - For one the customer stated that all quality attributes were in the requirements specification. He was wrong, but unwavering.
  - Scenarios from QAWs fed architecture definition and ATAM in other cases. This proved useful.

Improved stakeholder awareness of architectural definition and architecture teams awareness of stakeholders key desires
  - All in one room, many for the first time. This supplies the need to document outcome, quickly, effectively
  - There has been a problem finishing the document, because there are so many unknowns. We do not know how to resolve this.
  - Helped focus programs on architectural quality attributes and tactics used to address them.
  - Got both customer and contractor input into the process
  - Produced scenarios and questions
    - Seed for future ATAM
    - Architecture design should accommodate many of these scenarios
    - There should be no big surprise scenarios if done well