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** - 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
- **Try to get 2 business driver pitches, from both sides**
- **One Key is the Go/noGo decision**
- **Key to scope any additional work**
- **We are there to help, not criticize**

**Phase 2: Complete Evaluation**
- **Duration:** varies
- **Meeting:** primarily phone, email
- **Note architecture changes from phase 1**
- **Scenarios are key, spend extra time gathering, homework**

**Phase 3: Follow-up**
- **Duration:** varies
- **Meeting:** primarily phone, email
- **Do the report right away, else it loses value**
- **Follow up with arch help**
- **Red = talking points**

- **Try to get 2 business driver pitches, from both sides**
- **One Key is the Go/noGo decision**
- **Key to scope any additional work**
- **We are there to help, not criticize**

- **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**

- **Red = talking points**
ATAMSM Steps

Boeing Technology | Phantom Works

1. Present the ATAMSM
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

Phase 2

• 2. Business drivers are key, and usually hard to get a good job
• 3. Usually too much information, stop scenario questions, focus on tactics and patterns. Also, tactic/quality table.
• 4. Often this is already done for us
• 5. Difficult to do a good job each time, perhaps modify technique
• 6. Put architecture on stage, nothing else.
• 7. 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.
• 8. Close out dead topics quick. Spend lots of hours analyzing. Delegate writeups.
• 9. What to do with ppt after presented?
• 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)

• Issue definition = out of control of architect

- Summarize (risks + issues) = risk themes
Labor Costs to perform an ATAM$^{SM}$

- At Boeing the effort to perform an ATAM$^{SM}$ 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$^{SM}$ approach has been developed for the small project
  - A large ATAM$^{SM}$ approach may take 3 to 5 times as much evaluator time for direct interaction on the program
### Boeing ATAM<sup>SM</sup> Statistics by Review

**Boeing Technology | Phantom Works**

**E&IT | NCO Thrust**

• 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>Architects' 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>
</tbody>
</table>

**Average**

4.3 | 14.7 | 61.4 | 108.6 | 89.7 | 18.0 | 3.7 | 1.3

• Reports are the programs property, not the evaluation teams.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>No performance model, tiny deployment</td>
<td>Make vs Buy 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></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Max loads, scenario, tiny deployment</td>
<td>Failure modes, smart forwarding</td>
<td></td>
<td></td>
<td>Infrastructure</td>
<td></td>
<td></td>
<td>Requirements ill defined, scope, ill defined, missing components</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Maximum loads, scalability</td>
<td>Failure detection, automatic reconfiguration</td>
<td></td>
<td></td>
<td>Modifiability, variability architecture</td>
<td></td>
<td></td>
<td>System engineering gives unanticipated requirements</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>Understand Final Requirements</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>complexity of using model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>Failed to component, fault detection</td>
<td></td>
<td></td>
<td>COTS Quality Attributes</td>
<td></td>
<td></td>
<td>Schedule and key mechanisms development</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td>Failed to component, undefined behavior</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></td>
<td></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 = ways to cure the ills*
### Scenario vs Quality Attribute Statistics

<table>
<thead>
<tr>
<th>Scenarios analyzed against quality attributes per ATAM</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, extendability)</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>1</td>
<td></td>
<td></td>
<td></td>
<td></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></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></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>3</td>
<td></td>
<td>3</td>
<td>2</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>4</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>11</td>
<td>10</td>
<td>2</td>
<td>4</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total:**

<table>
<thead>
<tr>
<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, extendability)</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>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.
<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>well planned</td>
<td></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>multi-threaded, isolation of components</td>
<td></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 ATAM$^\text{SM}$ 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, ATAM$^\text{SM}$ had a special focus on cost reduction
  • Risk identification and mitigation reduces rework
  • Cost savings suggestions also can result from the analysis.
• Sometimes, ATAM\textsuperscript{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\textsuperscript{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