Aerospace Software Architecture Evaluation Framework - Introduction

Software Architecture Working Group (contact: Alan Unell)
Introduction

• Software architecture is a key part of many of our largest programs
  – *Problems with software architecture can be extremely expensive to fix, especially if caught late*
  – *Software architecture has caused trouble both cost and schedule problems for many of programs in the past*

• A software-architecture-based CSI was established in FY2010 to provide guidance on how to improve software architecture evaluations
  – *The working group developed a framework of criteria and specific questions that can be used on NSS programs*
  – *The framework is not bound to any particular methodology, process or tool set and can be tailored to different evaluation needs and development phases*  
    • Conceptual (Reference Architecture)
    • Review of contractor’s architecture in development
    • Evaluation of planned expansion
Why a framework?

- Adds rigor to the evaluation
- Ensures comprehensive coverage of the architecture
- Provides an (increasing amount) of domain-specific guidance
- Evaluation team
  - *Partitions the work*
  - *Focuses on specific architectural issues relevant to the program*
  - *Avoids inadvertently missing key issues*
  - *Avoids reinventing the wheel*
What is the framework?

• Centerpiece: 1000+ questions organized into categories, dimensions (concerns), and domains
  – Evaluators tailor the framework to suit evaluation goals since each evaluation is different
    • A subset of the questions is selected for relevance
    • These questions are refined into program-specific questions that address key requirements/performance prioritized characteristics the architecture needs to deliver
  – Tailoring is required since there are different:
    • Goals of the review
    • Program lifecycle stages and different risk factors
    • Domains: Space? Ground? Launch?
    • Sources of information about the architecture
      – Documents only? Access to architects and system experts?
• Framework provides guidance (underway) on tailoring evaluations for certain development milestones (e.g., PDR, CDR, O&M, ECP incorporation…)
• Output of the evaluation is a list of strengths, risks and potential mitigations
Evaluation Criteria

**Categories**

**Category 1**
Arch Description
- Architecture Scope
- Documentation
- Architecture Tradeoffs
- Documentation
- Use of Viewpoints and Views
- Scope of Documentation
- Consistency Across Views
- Use of Description Notations
- Organization and Traceability of Documentation

**Category 2**
Arch Scope, Priorities, Tradeoffs
- Architectural Scope
- Software Quality Attributes and Tradeoffs

**Category 3**
Arch Satisfaction of Functionality/Qualities
- Architectural Decisions and Tradeoffs Reflect Program Drivers
- Flowdown of Enterprise/System Architecture to Software Architecture
- Allocation of Requirements to Software Architecture
- Software Architecture Consistency with User Interface
- Software Architecture and Data Architecture are Integrated
- COTS/GOTS Appropriateness
- Reuse Appropriateness
- Integration of Reuse into Software Architecture
- (...many more…)

**Category 4**
Architecture Development/Evolution Methodology
- Software Architecture Process
- Personnel and Skill Mix
- Communication and Interaction
- Tools

**Dimensions**

Note: Many category 3 dimensions drill down into specific domains
Specificity of Questions

Most Generic

“Level 1” Generic Questions

Does the architecture accommodate likely areas of change and growth without significant modification?

“Level 2” Domain-Specific Questions

Is the software architecture designed to accommodate additional satellites without significant modification?

“Level 3” Program-Specific Questions

For instance, how well can the architecture accommodate going from 3 to 7 satellites in the constellation?

(Defined on per-program basis by evaluation team)
Organization of Dimensions into Categories

• Four Categories:
  – Category 1: Architectural Description
  – Category 2: Architectural Scope Priorities and Trade-offs
  – Category 3: Architectural Satisfaction of Functionality/Quality
  – Category 4: Architecture Development/Evolution Methodology

• Each category has a set of associated dimensions and evaluation questions for each dimension.
  – Categories 1, 2, 4 are generic (not domain-specific).
  – Category 3 includes domain-specific questions.
Using the Framework: From Tailoring to Synthesis

**Goal Setting**
What is this evaluation for? What are its desired outcomes?

**Scoping**
Identify evaluation team. Select a subset of categories, dimensions, questions appropriate to goals.

**Partitioning**
Divide the work up among evaluators based on interest and expertise.

**Refinement**
Refine generic & domain-specific questions into program-specific questions as needed.

**Investigation**
Evaluators answer refined questions using available resources.

**Team Rollup**
Evaluation team aggregates analysis and issues from individual evaluators.

**Individual Rollup**
Evaluators generate analysis results from answers and identify issues to surface.

**Organization**
Team develops report tailored to evaluation goals, communication plan, and follow-up plans.

**Communication**
Evaluation team communicates results to customer.
Current Status

- Framework has been documented in an Aerospace-internal memorandum; we are continuing to mature the framework for a future wider release in some form.
- Have applied the framework fully to one program, partially to another
  - FY11 will see efforts to expand to other programs in different lifecycle phases
- Will be expanding the framework with
  - More questions
  - More domain-specific guidance
  - More guidance on tailoring the framework for different situations
  - Tool support
Backup Slides

Example domains, lists of dimensions, sample questions
Example Domains (Ground)

- Command & Control
- Mission Management
- Ground Mission Processing
- Ground System Management
- Ground Support Functions
Dimensions for Category 1

*Architectural Description*

- Architecture Scope Documentation
- Architecture Tradeoffs Documentation
- Use of Viewpoints and Views
- Documentation Completeness
- Internal Consistency of the Architecture
- Use of Description Notations
- Organization and Traceability of Architecture Documentation
Dimensions for Category 2

Architectural Scope Priorities and Trade-offs

• Architectural Scope
• Software Quality Attributes and Tradeoffs
Dimensions for Category 3

Architectural Satisfaction of Functionality/Qualities

- Architectural Decisions and Tradeoffs Reflect Program Drivers
- Flow-down of Enterprise/System Architecture to Software Architecture
- Allocation of Requirements to Software Architecture
- Software Architecture Consistency with User Interface
- Software Architecture and Data Architecture are Integrated
- COTS/GOTS Appropriateness
- Reuse Appropriateness
- Integration of Reuse and COTS into Software Architecture
- Openness and Platform-independence
- External Interfaces
- Modularity and Layered Architecture
- Scalability
Dimensions for Category 3 (cont.)

*Architectural Satisfaction of Functionality/Qualities*

- Flexibility
- Timeliness and Performance
- Reliability and Availability
- Security / Information Assurance
- Manageability
- Technology Readiness
- Usability
- Safety
- Extendibility/Extensibility
- Survivability
Dimensions for Category 4

Architecture Development/Evolution Methodology

- Software Architecture Process
- Personnel and Skill Mix
- Communication and Interaction
- Tools
Example Questions

“Reuse Appropriateness” Domain

• Provide the list of key reuse products. Provide pedigree for each selected reuse products. If the reuse software is a product line, provide the roadmap for the product, if there is one.

• Has there been sufficient analysis/prototyping of the reuse software to demonstrate feasibility of the reuse software for this system’s requirements?

  – What have been the selection processes (including trade studies, prototyping) employed for reuse products?

  – To what extent has reuse software been evaluated for its ability to meet requirements?

  – What analysis/prototype activities have been done to determine feasibility of the reuse software to meet this system’s needs?

  – Provide evidence and outcomes of reuse analysis/prototypes.

  – Provide evidence of the ability to integrate, modify and maintain this software.
Example Questions
“Modularity and Layered Architecture” Domain

• Is there a clear and reasonable separation of concerns (for example, application from infrastructure, user interface details from application behavior, hardware/operating system dependencies, middleware & commercial software dependencies)?

• What is the adopted layering model?
  – Are there any layer violations? Are the risks of these violations and adequate mitigations plans identified?

• Is a layering model used consistently throughout the architecture (an example of inconsistency: some permit a component to use services at any lower layer, some permit use of services only at the next layer)?
  – Are modular design principles (high cohesion among components, weak coupling & well-defined interfaces between components) incorporated to allow software to be functionally partitioned into scalable components?
Program
Evaluation Process

• Evaluation is divided into 4 top level categories
  – Each category contains a set of evaluation dimensions; there are 35 total dimensions (aka attributes against which the architecture is evaluated)
  – Each evaluation dimension includes several evaluation questions.

• Program name’s architecture-related requirements are mapped to the evaluation dimensions

• Evaluation questions have been are tailored to meet the specific program requirements.

• The evaluation was based on the architectural documents (SAD, SSDD) and interviews

• If any dimensions were out of scope list them here

(Items in blue are to be filled in by review team)
Evaluation Ratings & Summaries
## Overall Program Architecture Evaluation Ratings

<table>
<thead>
<tr>
<th>Category</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Architectural Description</td>
<td>G</td>
</tr>
<tr>
<td>2. Architectural Scope Priorities and Trade-offs</td>
<td>G</td>
</tr>
<tr>
<td>3. Architectural Satisfaction of Functionality/Quality</td>
<td>G</td>
</tr>
<tr>
<td>4. Architecture Development/Evolution Methodology</td>
<td>Y</td>
</tr>
</tbody>
</table>

- **G** Criteria satisfied
- **Y** Criteria issues exist that cause concern
- **R** Serious issues exist that may impact execution

SBU Markings as Appropriate
## Category 1: Architectural Description

<table>
<thead>
<tr>
<th>Sec</th>
<th>Dimension</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Architecture Scope Documentation</td>
<td>G</td>
</tr>
<tr>
<td>1.2</td>
<td>Architecture Tradeoffs Documentation</td>
<td>G</td>
</tr>
<tr>
<td>1.3</td>
<td>Use of Viewpoints and Views</td>
<td>G</td>
</tr>
<tr>
<td>1.4</td>
<td>Documentation Completeness</td>
<td>G</td>
</tr>
<tr>
<td>1.5</td>
<td>Internal Consistency of the Architecture</td>
<td>G</td>
</tr>
<tr>
<td>1.6</td>
<td>Use of Description Notations</td>
<td>G</td>
</tr>
<tr>
<td>1.7</td>
<td>Organization and Traceability of Architecture Documentation</td>
<td>G</td>
</tr>
</tbody>
</table>

- **G**: Criteria satisfied
- **Y**: Criteria issues exist that cause concern
- **R**: Serious issues exist that may impact execution
# Category 2: Architectural Scope Priorities and Trade-offs

<table>
<thead>
<tr>
<th>Sec</th>
<th>Dimension</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Architectural Scope</td>
<td>G/Y</td>
</tr>
<tr>
<td>2.2</td>
<td>Software Quality Attributes and Tradeoffs</td>
<td>G</td>
</tr>
</tbody>
</table>

- **G**: Criteria satisfied
- **Y**: Criteria issues exist that cause concern
- **R**: Serious issues exist that may impact execution

SBU Markings as Appropriate
# Category 3: Satisfaction of Functionality & Quality

<table>
<thead>
<tr>
<th>Sec</th>
<th>Dimension</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Architectural Decisions &amp; Tradeoffs Reflect Program Drivers</td>
<td>G</td>
</tr>
<tr>
<td>3.2</td>
<td>Flow-down of Enterprise/System Architecture to SW Archit.</td>
<td>G</td>
</tr>
<tr>
<td>3.3</td>
<td>Allocation of Requirements to Software Architecture</td>
<td>G</td>
</tr>
<tr>
<td>3.4</td>
<td>Software Architecture Consistency with User Interface</td>
<td>G</td>
</tr>
<tr>
<td>3.5</td>
<td>Software Architecture and Data Architecture are Integrated</td>
<td>G</td>
</tr>
<tr>
<td>3.6</td>
<td>COTS/GOTS Appropriateness</td>
<td>G</td>
</tr>
<tr>
<td>3.7</td>
<td>Reuse Appropriateness</td>
<td>G</td>
</tr>
<tr>
<td>3.8</td>
<td>Integration of Reuse into Software Architecture</td>
<td>G</td>
</tr>
<tr>
<td>3.9</td>
<td>Openness and Platform-independence</td>
<td>G</td>
</tr>
<tr>
<td>3.10</td>
<td>External Interfaces</td>
<td>G</td>
</tr>
<tr>
<td>3.11</td>
<td>Modularity and Layered Architecture</td>
<td>G</td>
</tr>
<tr>
<td>3.12</td>
<td>Scalability</td>
<td>G</td>
</tr>
</tbody>
</table>

- **G**: Criteria satisfied
- **Y**: Criteria issues exist that cause concern
- **R**: Serious issues exist that may impact execution
<table>
<thead>
<tr>
<th>Sec</th>
<th>Dimension</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.13</td>
<td>Flexibility</td>
<td>G</td>
</tr>
<tr>
<td>3.14</td>
<td>Timeliness and Performance</td>
<td>G</td>
</tr>
<tr>
<td>3.15</td>
<td>Reliability and Availability</td>
<td>G</td>
</tr>
<tr>
<td>3.16</td>
<td>Security / Information Assurance</td>
<td>G</td>
</tr>
<tr>
<td>3.17</td>
<td>Manageability</td>
<td>G</td>
</tr>
<tr>
<td>3.18</td>
<td>Technology Readiness</td>
<td>G</td>
</tr>
<tr>
<td>3.19</td>
<td>Usability</td>
<td>G</td>
</tr>
<tr>
<td>3.20</td>
<td>Safety</td>
<td>G</td>
</tr>
<tr>
<td>3.21</td>
<td>Extendibility/Extensibility</td>
<td>G</td>
</tr>
<tr>
<td>3.22</td>
<td>Survivability</td>
<td>G</td>
</tr>
<tr>
<td>3.23</td>
<td>Testability</td>
<td>G</td>
</tr>
<tr>
<td>3.24</td>
<td>Enterprise Interoperability</td>
<td>G</td>
</tr>
</tbody>
</table>

G: Criteria satisfied

Criteria issues exist that cause concern

Serious issues exist that may impact execution
## Category 4: Architectural Scope Priorities and Trade-offs

<table>
<thead>
<tr>
<th>Sec</th>
<th>Dimension</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Software Architecture Process</td>
<td>G</td>
</tr>
<tr>
<td>4.2</td>
<td>Personnel and Skill Mix</td>
<td>N/A</td>
</tr>
<tr>
<td>4.3</td>
<td>Communication and Interaction</td>
<td>G</td>
</tr>
<tr>
<td>4.4</td>
<td>Tools</td>
<td>R</td>
</tr>
</tbody>
</table>

G: Criteria satisfied

G: Criteria issues exist that cause concern

R: Serious issues exist that may impact execution

SBU Markings as Appropriate
Overall Strengths

• Insert Strengths
Overall Concerns and Mitigations

- Insert Concerns and Mitigations
Evaluation Findings

Category 1, Architectural Description
Dimension 1.1 Architecture Scope Documentation

• Definition
  – *This dimension addresses whether the scope of the architecture is properly documented. Whether the scope of the architecture itself is appropriate or not is covered in a related list called “architectural scope.”*

• Summary of the Program Approach

• Overall Evaluation

• Strengths/Concerns
  – *Strengths*
  
  – *Concerns*
  
  – *Mitigations*
Evaluation Findings

Category 2, Architectural Scope Priorities and Trade-offs
Evaluation Findings

Category 3, Architectural Satisfaction of Functionality & Quality
Evaluation Findings

Category 4, Architecture Development/Evolution Methodology