GSAW2004
Ground System Architectures Workshop

Breakout Session 10A
Architecture-Centric Evolution & Evaluation (ACE2) of Software-Intensive Systems

Chair

Dr. Sergio Alvarado
The Aerospace Corporation

Committee

Daniel Dayton, Suellen Eslinger, Dr. Peter Hantos, Myron Hecht, Karen Owens, Dr. Phillip Schmidt, and L. Robert Varney
The Aerospace Corporation

Dr. Thomas Alspaugh, John Georgas, and Scott Hendrickson
Institute for Software Research, UC Irvine

ACE2 Session Goals

- **Promote central role of software architecture during acquisition/development of software-intensive systems**
  - ❖ Improved responsiveness to changes in requirements and complexity
  - ❖ Early identification of flaws
  - ❖ Streamlined system implementation, testing, and maintenance
- **Explore how to specify and evaluate software system architectures that support software system evolution**
  - ❖ Techniques for software architecture representation
  - ❖ Tools for software architecture analysis
  - ❖ Software system architecting practices, standards, and policies

ACE2 Session Discussion Baseline

1. **Architecture as a Basis for Understandability**
   - Provide views of software system with levels of granularity appropriate for each stakeholder (acquirer, overseer, developer, tester, and operator) to have insight into system functionality

2. **Architecture as a Basis for Assessing Maintainability**
   - Link requirements to system implementation so that stakeholders can assess degree of system change and cost/schedule impact from upgrading, changing, and integrating COTS products used in implementation

3. **Architecture as a Basis for Assessing Extensibility**
   - Link requirements to system implementation so that stakeholders can assess degree of system change and cost/schedule impact from new requirements on system size, complexity, environments, services, and interoperability

4. **Architecture as a Basis for Assessing Executabilty**
   - Support development of executable models so that stakeholders can assess impact of new requirements on system performance and reliability

ACE2 Session Agenda

- **First Segment (13:00 – 15:00)**
  - Lt. Col. Laura Pope, Air Force Space and Missile Systems Center
  - Dr. Joel Sercel, MILSATCOM Joint Program Office
  - Dr. Linda Northrop, Software Engineering Institute
  - Dr. Peter Hantos, The Aerospace Corporation
  - Discussion and formulation of findings

- **Second Segment (15:15 – 17:00)**
  - Capt. Bryan Berg, Air Force Space and Missile Systems Center
  - Peter Shames, Jet Propulsion Laboratory
  - Jim Boegman, Raytheon
  - Dr. Allen Nikora, Jet Propulsion Laboratory; Myron Hecht and Douglas Buettner, The Aerospace Corporation
  - Discussion and formulation of findings

Lt. Col. Laura Pope:
ACE2 Opening Statement

- **Why does the Government care about migrating to an architecture-centric evolution and evaluation of software-intensive systems?**
  - Understandability
    - CONOPS not fully mature at the start of software design
  - Extensibility/Executability
    - Changing interfaces, new requirements, changing CONOPS
  - Maintainability
    - Architecture design does not adequately consider O&M costs
- **Create & document architecture-centric views of software-intensive system up-front**
  - Fully coordinate them with all system stakeholders
  - Keep them current.

Dr. Joel Sercel:
Architecture as a Tool for Managing Change

- **Architecture is a set of constraints on designs**
  - Effective constraints define effective architecture
  - C4ISR useful but not necessary nor sufficient
- **Architecture necessary for managing change**
  - Defined early in the product development life cycle
  - Maintained as collaborative product of software IPT

Dr. Linda Northrop:
Architecture Business Cycle Ensuring Product Qualities

- **The architecture must be descriptive and prescriptive**
- **Quality attribute requirements drive the software architecture**
  - Examples: Understandability, Maintainability, Extensibility, Executability …
  - SEI has methodology/tools for defining quality attributes
- **Architecture-centric activities drive software system life cycle**
  - Explicit focus on quality attributes
  - Directly involve stakeholders

Dr. Peter Hantos:

Software Reviews Since Acquisition Reform – Architecture-Driven Considerations

- **Architecture-driven considerations essential in carrying out reviews**
  - ❖ MIL-STD-1521B is inadequate as the basis for design reviews
- **Object-oriented methodologies critical in planning reviews**
  - ❖ Configuration Item concept not supportive of development practices
- **In-process reviews must track allocated Technical Performance Measurements**

Capt. Bryan Berg:
COBRA Architecture

- **COBRA=COTS-Based Real-Time Architecture**
  - Based on COTS to minimize cost and maximize functionality
  - COTS chosen for “best in class”
  - Architectural decisions based on system risk

- **Lessons Learned**
  - **Maintenance:** Upgrades limited because of hardware/software compatibility
  - **Extensibility:** Older products cannot be upgraded cost effectively
  - **Way Ahead:**Use standardized interfaces to avoid compatibility problems

Peter Shames:
Reference Architecture for Space Data Systems (RASDS)

- **RASDS provides architectural view of end-to-end data systems**
- **Understandability**
  - ❖ Provides insight into functionality and relationship among elements so that complexity may be managed
- **Maintainability**
  - ❖ Supports allocation of functionality, design trades, deployment trades, and analysis of impact of requirements changes
- **Extensibility**
  - ❖ Provides the means to describe and reason about system and component size, complexity, performance, and operating environments
- **Executability**
  - ❖ It is possible to model system behavior at a coarse level of granularity

Jim Boegman:
Raytheon (NPOESS) Perspective on Software Architecture

- **Spectrum from architecture to implementation**
  - ❖ Requirements describe the spectrum
- **Software architecture**
  - ❖ Understandability
    - – Different views enable comprehension at appropriate level of detail
  - ❖ Maintainability/extensibility/executability
    - – Architecture alone not enough to assess cost/schedule impacts

Dr. Allen Nikora, Myron Hecht, & Douglas Buettner:
Software Reliability Measurement

- **Reliability-centric process**
  - Software reliability important to determine software release schedule
  - Reliability estimated from testing results
- **Architecture not a good predictor of software reliability**
- **Reliability a good indicator of good architecture**

ACE2 Session Summary

- **Central role of software architecture in understandability**
  - Define, create, document, and keep current architecture-centric views
  - Directly involve all stakeholders
- **Standards needed to support reviews**
- **Open question: How to specify architecture to address maintainability, extensibility & executability**
  - Develop up-front stakeholder agreement on views and requirements on illities
  - Define domain-specific reference architectures
  - Use architecture in conjunction with other tools/models (e.g., reliability models)
- **GSAW should continue supporting ACE2 discussion**
  - Gain insight why/how to develop descriptive and prescriptive architectures