GSAW2007
Ground System Architectures Workshop

Working Group 4A
Architecture-Centric Evolution (ACE) of Software-Intensive Systems

Chairs
Dr. Sergio Alvarado
Sheri Benator
Dr. Phillip Schmidt
The Aerospace Corporation

ACE Working Group Goals

- **Fifth of a GSAW series**
  - ❖ Promote the central role of software architectures during the acquisition & development of software-intensive systems
- **Forum for software-intensive system experts, users, developers & researchers**
  - ❖ Collaborate and elucidate high-level recommendations for improving software architectures representation, development & design
- **Presentations & panel discussion**
  - ❖ Methodologies, Tools, Techniques, & Practices for Analyzing Architectures for Software-Intensive Systems

ACE Invited Panelists

- **Acquisition and Oversight Perspective**
  - Peter Capell, Software Engineering Institute
  - Dr. Peter Hantos, The Aerospace Corporation
- **Development Perspective**
  - Richard Anthony, General Dynamics C4 Systems
  - Sean Kelly, Lockheed Martin, IS&GS
- **Research and Tools Perspective**
  - Dr. Hans-Peter Hoffman, Telelogic
  - Dr. Azad Madni, Intelligent Systems Technology, Inc.
  - Dr. Kathryn Weiss, Jet Propulsion Laboratory
- **Moderator**
  - Dr. Sergio Alvarado and Sheri Benator, The Aerospace Corporation

Key Points - 1

- **Architecture Representation**
  - UML provides a common stakeholder language
    - Best practices are needed for its use in domain-specific areas
    - Communication with meta-models is an important area of research
  - Strong software architecture-centric perspective is still new for satellite systems
    - JPL is developing architecture-centric guidance and tools for coherent architectural design
  - Key to managing complex, large-scale SW systems is to distinguish between buildtime (logical components) and runtime (deployed components) views
  - Front-end conceptual analysis is needed to understand how to select, extend, and apply tools and modeling languages
    - Use views and modeling that apply to the problem at hand
  - System architecture approach was provided using SysML and leading directly into software architecture
    - Telelogic's Harmony is a tool-independent model driven process
  - Although SysML is being applied by some organizations, it has not been fully adopted by hardware engineers
  - Tools are evolving to better support architecture needs
    - UML tool vendors working on supporting model transformation capabilities

Key Points - 2

- **Architecture Analysis**
  - Front-end analysis needed to define quality attributes and follow-on assessment needed to determine how well they are being met
    - Architecture Tradeoff Analysis elicits, prioritizes, trades-off quality requirements
    - QUASAR assesses the quality attributes of system and subsystem architectures
  - Quality assessments of system/subsystem architectures not currently written into development contracts, but implemented as best practices
  - Architectural complexity should be analyzed
    - There is a difference between problem complexity (which cannot be removed) and solution complexity (which can be reduced)
- **Organizational considerations**
  - In large programs with prime and many subs at CMMI level 5 it is unlikely that a single melded methodology can be achieved
    - Need practices to integrate/interact with disparate methods, products, tools
    - Focus on integrating products of disparate methodologies
    - Yet on one presented multi-organizational program, common process and architecture methodology with modifications where warranted led to success
  - Need for system engineering and software engineering to work together in addressing cross-cutting architecture concerns
    - Sub-contract the problem ownership and coordinate via Integrated Product Teams

Conclusions

- **Organizations need to define their software development and analysis practices within the context of**
  - Problem complexity
  - Multi-organizational teams
  - Quality assessment techniques
  - Multiple and evolving architecture methodologies, modeling languages, tools, and standards