— Working Group Outbrief —

Architecture-Centric Evolution (ACE) Working Group 2008

Session 10C

Sheri Benator, Sergio Alvarado,
Phil Schmidt,
The Aerospace Corporation

Session Goals

- Sixth 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, & analysis
- Presentations & panel discussion
  - Software architecture techniques, tools, and practices for more responsive ground systems that better adapt to new capabilities and missions

Presenters/Panelists

- **Development Perspective**
  - Jeff Garland, Crystal Clear Software, Inc.
  - George K Auyeung , Lockheed Martin, IS&GS
  - Jeff Estefan, Jet Propulsion Laboratory
- **Research and Tools Perspective**
  - Dr. Peter Capell, Software Engineering Institute
  - Dr. Hadar Ziv, University of California Irvine
  - Paula Obeid, EmbeddedPlus
- **Acquisition and Oversight Perspective**
  - Bill Macaulay, The Aerospace Corporation
  - Dr. Peter Hantos, The Aerospace Corporation

Key Points

- Best of times and worst of times for software architecture
  - Major improvements in documentation, evaluation, and tools
  - Gap between requirements and architecture persists
- Identified successful architecture techniques
  - Formal process for architecture quality evaluation
  - Dynamic modeling/simulation to verify architecture
    - High level components
    - However, it is difficult to identify and define the key scenarios
  - Formal analysis of failure modes/response
  - Early day-in-the-life scenario development

Key Points

- Panel challenged to define architecture process and artifacts for software system
  - Key attributes: extensible, adaptable, and maintainable
  - Defined necessary analysis and documentation
  - Program realities (schedule, costs, immediate needs) resulted in tradeoffs
  - “Nice to haves” won’t occur without explicit requirements
- Importance of architecture governance
  - Principles, architecture board, compliance
- Software needs to be delivered as a complete package
  - Code, architecture, tests, environment…

Conclusions

- Programs would benefit from early and continuing discussion of architectural quality tradeoffs vs. programmatic constraints (cost, risk, schedule)
- For each program, stakeholders need to agree on:
  - Important architectural qualities (e.g. adaptability, extensibility, maintainability)
  - Definition, measurement, and evaluation of those qualities
  - Minimal acceptable threshold (what can't be compromised)