Working Group 9B
Architecture-Centric Evolution, Evaluation & Elaboration (ACE3) of Software-Intensive Systems

Chairs
Dr. Sergio Alvarado
Dr. Scott Turner
The Aerospace Corporation

Dr. Hadar Ziv
Institute for Software Research, UC Irvine
ACE3 Session Goals

- **Address stakeholder needs in evolution, evaluation, and elaboration of architectures in software system lifecycle**
  - Presentations from members of government agencies, contractors, academia, and federally funded research and development centers

- **Promote central role of software architecture during acquisition/development of software-intensive systems**
  - Forum for elucidating high-level recommendations for improving architecture practices, representation techniques, and analysis tools
ACE3 Session Discussion Baseline

1. Elaboration
   - Architecture-based management of “requirements-creep” risk
   - Architecture constructs/tools for seamless requirement-to-implementation trace

2. Evolution
   - Architecture constructs/tools for supporting system evolution requirements
     - Maintainability
       » Upgrades, changes & integration of COTS products for system implementation
     - Extensibility
       » Increased system size, complexity, environments, services & interoperability
     - Executability
       » System performance and reliability

3. Evaluation
   - Challenges to architecture evaluation within software system acquisition
   - Architecture constructs/tools required for software system evaluation
ACE3 Presentations

• **Acquisition Perspective**
  - Frank Sisti, Air Force Space and Missile Systems Center
  - Maj. Mark Tuttle, Air Force Space and Missile Systems Center

• **Overseeing Perspective**
  - Dr. Charles Hammons, Software Engineering Institute
  - Dr. Peter Hantos, The Aerospace Corporation
  - Dr. Phillip Schmidt, The Aerospace Corporation

• **Development Perspective**
  - George Haley, Product Line Manager, Northrop Grumman
  - Jeff Garland, “Large-Scale Software Architecture Book Coauthor,” CrystalClear Software
  - Ted Faison, “Component-Based Development Book Author,” Faison Computing

• **Research Perspective**
  - Dr. Hadar Ziv, Institute for Software Research, University of California, Irvine

• **Moderators**
  - Dr. Sergio Alvarado, The Aerospace Corporation
  - Dr. Scott Turner, The Aerospace Corporation
Elaboration

• Architecture must be understandable to all stakeholders
  ✓ Software needs explicit representation in the program office (Sisti)
  ✓ Customer (government) needs only high-level architecture with key features (Sisti)

• Architecture must be elaborated in larger lifecycle context (Hantos, Ziv, Tuttle)
  ✓ Make stakeholders explicit in architecture (Ziv)

• Key UML diagrams for high-level architectures for large-scale systems (Garland)
  ✓ Context
  ✓ Component
  ✓ Component Interaction
  ✓ Layered Subsystem
  ✓ Deployment
Evolution

• Evolution more important now because of changing environment (Hammons)
  - Changing threats, rapid technological development, political environment, fluid requirements, longer service life
  - Each system serves as the seed for the next generation

• Support for system evolution must start in the architecture (Tuttle, Hammons)
  - The groundwork for evolution must be laid before the need for evolution
  - System evolution is often driven by risk reduction (Tuttle)

• Component decoupling in architecture enables continuous system evolution (Faison)
  - Decoupling enabled by standards, defined APIs, “Plug and Play”, event-based architectures, layered systems, common messaging model, and similar design elements (Faison, Hammons, Garland)
Evaluation

• Our ability to evaluate lags behind our ability to create (Hantos, Haley, Schmidt)
  ❖ Traditional metrics and evaluation processes don’t apply well to architectures (Haley, Hantos)
  ❖ Work to develop new approaches is still underway (Ziv)
  ❖ Need tools (e.g., temperature charts) to succinctly communicate evaluation to all stakeholders (Tuttle)

• Focus on bottom-line criteria for evaluation of architectures (Haley)
  ❖ Utility, Development Cost/Schedule/Risk, O&M Cost

• Architecture evaluation is a key tool for managing complexity (Schmidt) and risk (Tuttle)
  ❖ Space systems are typically very complex, distributed (Schmidt, Garland)
  ❖ Automated evaluation can identify issues otherwise lost in the complexity (Schmidt)
  ❖ We must produce architectures that can be evaluated (Schmidt, Sisti)