The Role of Software Architecture in Managing Complexity

Daniel Dvorak
JPL / Caltech
Daniel.L.Dvorak@jpl.nasa.gov
The Message

• **Software is a system’s “complexity sponge”**
  – Software most easily accommodates new functions
  – Software enables hardware standardization
  – Software accommodates evolving understanding

• **Software complexity is a growing problem**

• **Complexity is an issue of understanding**

• **The key to understanding is in patterns, which are used extensively in software**
  • Hierarchies • Abstractions • Structure

• **Architecture is the job of choosing and exploiting good patterns**

• **To address complexity, our software architectures must improve**

Source: Bob Rasmussen, JPL
Span of Software Architecture Concerns

- **Architecture should address quality attributes**
  - performance, availability, maintainability, modifiability, security, testability, operability, etc.
  - Watch for unsubstantiated or ambiguous rqmts

- **Architecture should address analyzability and implementability**

- **Architecture should address principles of design**
  - Identify and follow control architecture principles
  - Leverage appropriate architectural patterns

- **Architecture should address verifiability**
  - Can simplify or complicate verification

- **Architecture should address operability**
  - Inadequate architecture complicates operations
  - Operational workarounds raises risk
Benefits of Architecture Reviews

• “Architecture reviews tend to increase quality, control cost, and decrease budget risk.”

• “In our experience, the average [architecture] review pays back at least twelve times its cost.”
  – [Daniel Starr and Gus Zimmerman, *STQE Magazine*, July/August 2002]

• Beneficial side effects:
  – Team members gain a better understanding of their project
  – Cross-organizational learning is enhanced
  – Architecture reviews …
    • get management attention without personal retribution
    • assist organizational change