20 Years of Architecture

Bob Schwanke
Siemens Corporation
Corporate Research
Princeton, NJ USA
A Pre-history of Software Architecture

• Information-hiding Principle
  • (“On the criteria …” Parnas, 1972)
• Hierarchical Structure
  • (“On a buzzword …” Parnas, 1974?)
• Data Encapsulation
  • (“Some conclusions from an experiment … Parnas, ???)
  • (“Modularization and hierarchy in a family of operating systems”, Habermann/Flon/Cooprider 1975)
• Separate dependency specs from code
  • (“Programming-in-the-Large …” DeRemer and Kron, 1975?)
• Module Guide
  • (“The modular structure of complex systems”, Parnas, Clements, Weiss 1984)
• Software Engineering Institute (Habermann, 1984)
Hard Problems in Modularity

Modules should decouple development tasks
- Which ones?
- How far into the future?
- Can’t decouple them all

Anticipating Change
- Marketplace
- Stakeholders
- Technology

Measuring Modularity
- Detecting modularity errors using code structure and change sets.
  - Files that change together, not due to static dependency
  - Prof. Yuanfang Cai and students, Drexel University
- Predicting future change using structure measures and change history
  - Prof. Alan MacCormack, MIT/Harvard Business School and students
  - Analyzing Siemens projects
Technology Stacks
• Specialization forces us to rely on third-party components
• A recent small project imported 15 technologies.
• Bigger project: 300 open source components, 30 distinct licenses
• Lose control over aggregated quality attributes
  • E.g. telephone switch reliability.
    • Four VoIP switch HW/SW vendors
    • Third-party server hardware
    • 18 month server market window
    • How reliable is the hardware?
Hard Problems in Architecture Description

Maintainable Architecture Descriptions
• Subsystem tree is almost enough
• Other information has diminishing return

System Architecture
• Mostly Software Architecture
• Add physical/mechanical/electrical components
• Cross-domain communication, trust, and engagement.
• Requires “real” engineering education