BEYOND SHALL STATEMENTS: MODERNIZING REQUIREMENTS ENGINEERING

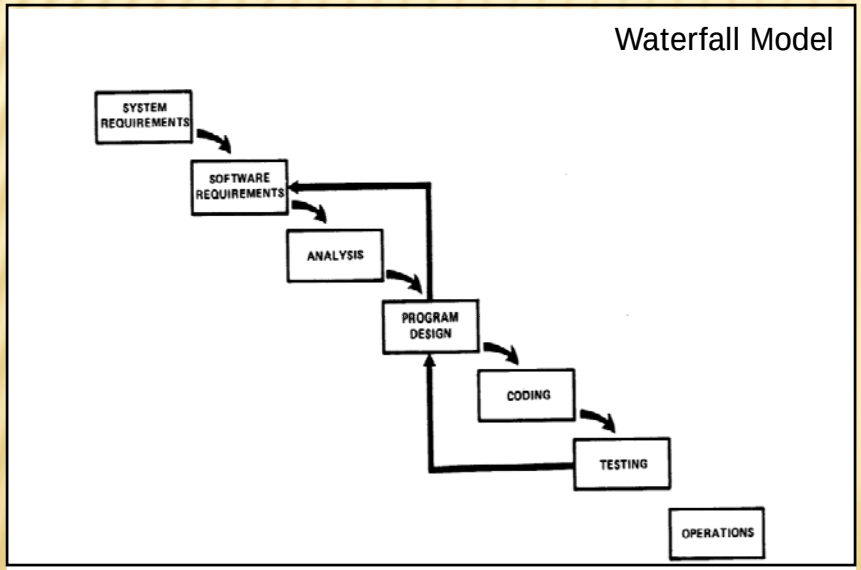
Leyna Cotran
Lockheed Martin Space Systems Company &
University of California, Irvine
Systems Engineer Staff
leyna.c.cotran@lmco.com

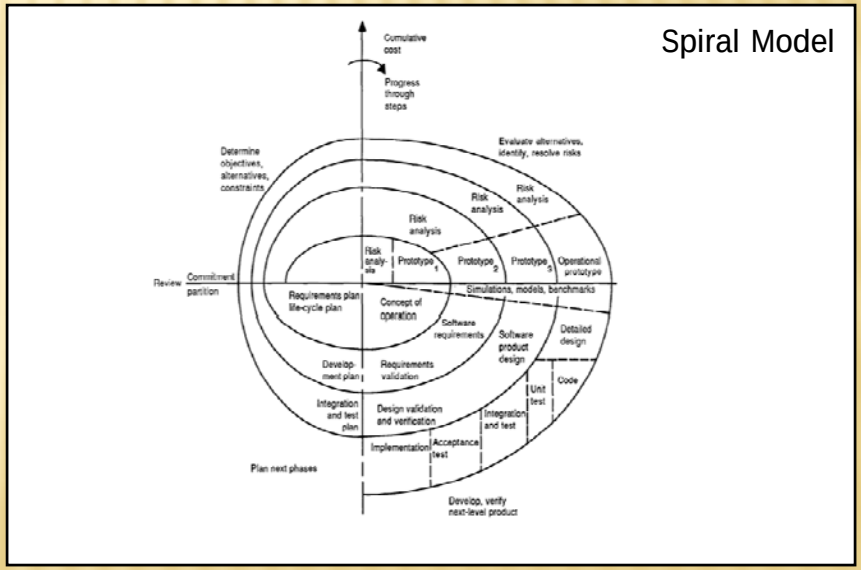
WORKSHOP MOTIVATION

- Requirements continue to be a critical issue for large systems engineering efforts.
- Requirements have not been a prominent discussion topic at GSAW.
- GSAW topics generally focus on new technologies and design techniques (i.e. service-oriented architectures and product-line development).
  - These techniques have significant impacts on system acquisition and requirements development
  - These implications have been under-investigated.
- Significant room for improvements in techniques, practices.

THE HARDEST PART

- "The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed requirements…No other part of the work cripples the resulting system if done wrong. No other part is as difficult to rectify later. Therefore, the most important function that the software builder performs for the client is the iterative extraction and refinement of the product requirements."-- F. Brooks





OBSERVATIONS OF REQUIREMENTS ENGINEERING PRACTICES

- Requirements engineering activities have generally stood alone
  - Many requirements and specs continue to be ambiguous, incomplete
    - Still focused on natural-language “shall” statements
  - Independent of design – positives and negatives here
  - Many requirements over-constrain the system
    - Clearly bound to one particular solution
    - Don’t distinguish external constraints from internal ones
- Requirements developed to meet contract deliverables as much as to provide value in the engineering process

WHY MOVE "BEYOND SHALL STATEMENTS?"

- The English language *is* hard to understand
  - Poor sentence structure is too common
  - And / or / shall not / if / refer-to / should
- Multiple ways a requirement can be interpreted
  - Limitations on analyzability
- Too detailed or not enough detail
- Sentence structure can be too complex or too simple
- Shall statements are not enough:
  - More contextual information is always needed
- Shall statements fall into a recycle-mode: going from one document to the next

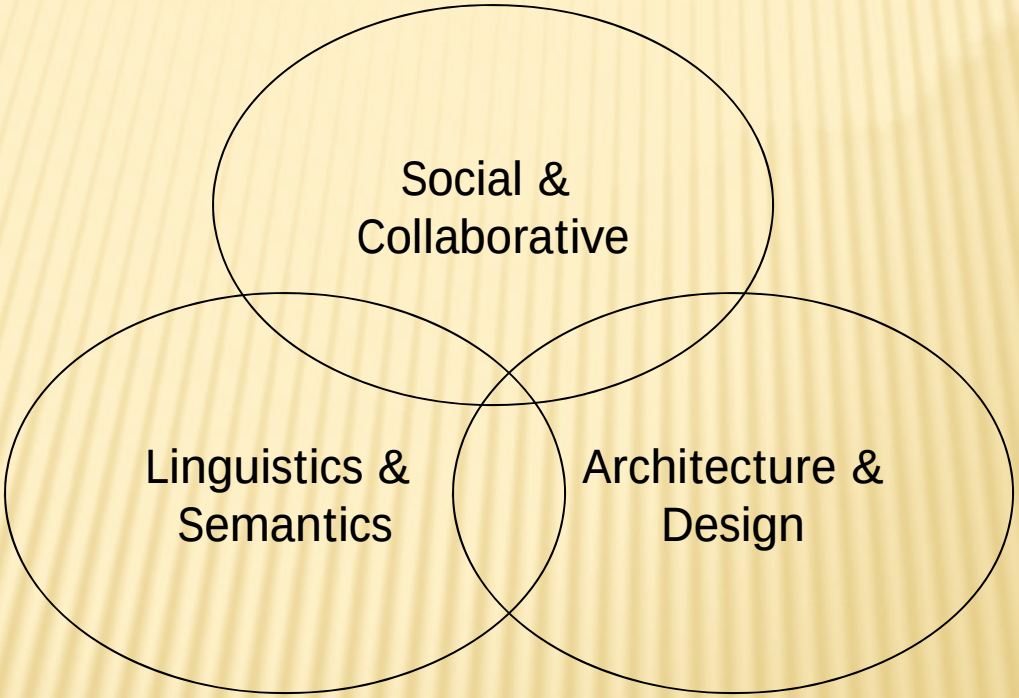
RE REALITIES

- Design and requirements activities are tightly coupled
  - Co-evolution of architecture and requirements
- Key pieces of information dispersed in the system
  - Inconsistent information
  - Informally maintained or mismanaged
  - Different representations
- Each stakeholder has a unique way of capturing information
  - Architects: styles, layouts, dataflows
  - Requirement Engineers: models, standalone statements, notes
  - Developers / Designers: source code, libraries, packages, classes
  - Testers: scripts, suites
    - ...how do we cross-cut through this critical information?

RELATIONSHIP TO ARCHITECTURE

- Traditional systems engineering teaches us to keep requirements and design separated
  - But often, new systems are incremental improvements to existing architectures, or compositions of existing architectures
  - Architectures can provide a frame of reference
    - a vocabulary
    - a basis for describing properties
    - a basis for analysis
- Topic for discussion: to what extent should we adapt our RE practices to better leverage architectures?

RE PERSPECTIVES



AGENDA FOR TODAY

- "Nano-keynote" address from Professor Richard N. Taylor
- Panel Discussion 1
  - Dr. Richard Taylor, University of California, Irvine
  - Dale Robinson, Raytheon Space and Airborne Systems
  - Emil White, Lockheed Martin Space Systems Company
- Break at 3p
- Panel Discussion 2
  - Dr. Ban Al-Ani, University of California, Irvine
  - Jorge Seidel, Aerospace Corporation
  - Andrea Richards, Raytheon Space and Airborne System