# Software Assurance and Cybersecurity

**William L Scherlis**
**Professor of Computer Science**
**Director, Institute for Software Research**

***Nuclear Regulatory Commission***
***17 December 2015***

## Assurance

***The aim of any testing scheme is to ensure that the customer gets substantially the software that he ordered and it must provide the customer with convincing evidence that this is so.***

**— NATO Software Engineering report 1968**

# Software assurance challenges

- Technical Difficulties
  - Higher levels of capability and increasing complexity of systems
  - Active resiliency and robustness
  - Systems composed from diversely-sourced general-purpose components
  - Diversity to reduce common cause failure
  - Distributed, interconnected, and concurrent designs
  - Compromised operating environments and continuous attack
- Difficulties in Evaluation
  - Gaps in evaluation practices
    - Reliance on unstructured informal documents
    - Inability to formally link faults, errors, failures, hazards
    - Reverse engineering and evaluations after the fact
    - Components not designed to support effective/efficient evaluation
    - Reliance on heuristic analysis tools and probabilistic models
    - Over-reliance on process compliance
  - Business structures
    - Opacity and technical data concerns
    - Compliance focus
    - Supply chains: Multi-sourcing, COTS, open source

# Addressing these challenges – project experience

- DoD OSD Systems Engineering Research Center task on System Assurance (*ongoing*)
  - Baselining evaluation standards
  - Meta-criteria development
  - Advancement of technical practice
  - Framework for evidence-based standards

- NSA Science of Security Lablet (*ongoing*)
  - Focus on five identified Hard Problems, including composition and secure systems engineering (#1), resilient design (#4), and human users (#5)
  - Development of research community
  - Advancement of scientific methodologies in security research

- NASA High Dependability Computing Program (*completed*)
  - Triangle model for testbed studies and rapid transition, partnering
    (1) Development of technical models, analyses, and tools
    (2) Advancement of measurement and evaluation capability
    (3) Mission partners and artifacts

# Opportunities for evidence-based assurance

- Emerging Technical Enablers
  - Models and analyses for designs, quality attributes, and code
  - Formal models for requirements supporting precise analysis
  - Argumentation structures for hazards and safety analysis
  - Modern data-intensive software engineering
  - Potential for rich dependency models to link evidence

- Prospects for *Safety-First Assurance Practice*
  - Assemble evidence during development and operation from executable artifacts, models, analyses, and other sources
  - Organize and analyze evidence using explicit dependency models
  - Support decision analysis and argument structures
  - Use analytic approaches to quality, defense-in-depth, and diversity
  - Build causal linkage models that link requirements, code, and operations
  - Develop sound principles and constraints for dedication evaluation

# Resources

- National Research Council study on Defense Software
  - http://www.nap.edu/catalog/12979/critical-code-software-producibility-for-defense
- National Research Council study on Dependable Systems (D Jackson)
  - http://www.nap.edu/catalog/11923/software-for-dependable-systems-sufficient-evidence
- Defense Science Board study on Foreign Software in the Supply Chain
  - http://www.acq.osd.mil/dsb/reports/ADA486949.pdf
- Software Engineering Institute post regarding software assurance
  - https://insights.sei.cmu.edu/sei_blog/2013/02/looking-ahead-the-sei-technical-strategic-plan-part-2.html
- NITRD Workshop on Designed-In Security: Practices and Research Needs
  - http://cps-vo.org/node/12673