Logistics

- Sponsored by the Software Engineering Institute (SEI)
- Washington D.C. (next 3 years)
- September 18 - 21
- Series of half- and quarter-day conferences.
- 820 Attendees
  - Heavy emphasis on DoD and its major contractors; limited private sector involvement.
Conference Tracks

- Technology Transition
- Network Security
- Architecture Tradeoff Analysis and Product Line Practice
- COTS-based Systems and Dependable Systems Upgrade
- Process Improvement
Technology Transition

- Accelerating Software Technology Adoption (ASTA) Initiative:
  
  “Help producers and adopters of software engineering technology solve technology adoption problems by offering systematic, workable, and efficient strategies, methods, and tools to ensure that technologies are rapidly and effectively deployed and adopted.”
Technology Transition

- **Software Engineering Measurement and Analysis (SEMA) Initiative:**
  
  “Helps organizations use measurement and analysis practices and techniques to make data-driven decisions, as well as collecting and disseminating data on the costs and benefits of specific software engineering practices.”
Technology Transition

- Conferences included:
  - Better than a Better Mousetrap: Planning for Transition
  - The Important of Knowledge Networks in Technology Maturation and Transition
  - Requirements for Technology Change Management: Results of an Ongoing Study
  - Using Integrated Product Teams to Improve Technology Introduction.
Network Security

✦ Survivable Systems Initiative:

“Concentrates on ensuring that appropriate technology and practices are used to prevent successful attacks on networked systems and to limit the damage caused by successful attacks.”
Network Security

Conferences included:

- Concepts and Trends in Information Security
- Executive Perspective on Information Technology Security
- Information Security Risk Assessments: A New Approach
- State of the Practice of Intrusion Detection Technologies
Architecture Tradeoff Analysis and Product Line Practice

- **Architecture Tradeoff Analysis Initiative:**
  
  “Focuses on proven techniques to predict the impact of software architecture decisions on selected product quality attributes such as performance, robustness, and modification capabilities.”
Architecture Tradeoff Analysis and Product Line Practice

Product Line Practice Initiative:

“Selects, refines, and establishes technical practices of demonstrated effectiveness for identifying and exploiting commonalities that exist across software systems in particular domains.”
Architecture Tradeoff Analysis and Product Line Practice

- Conferences included:
  - An Architectural Approach to the Economic Modeling of Software
  - Architecture Tradeoff and Analysis (ATA) Technology and Transition
  - Attribute-Based Architectural Styles (ABAs)
  - Introduction to the Architecture-Based Design Method
  - Software Architecture Documentation in Practice
Architecture Tradeoff Analysis and Product Line Practice

- Conferences included:
  - Product Line Practice Technology and Transition
  - Software Product Lines in Practice
  - Building and Communicating a Product Line Business Case
  - Mining Assets for Product Lines
  - Building Reusable Testing Assets for a Product Line
COTS-Based Systems and Dependable System Upgrade

- COTS-Based Systems Initiative:
  “Establishes principles and practices for constructing and evolving systems using previously built and commercially available components.”
COTS-Based Systems and Dependable System Upgrade

- Dependable System Upgrade Initiative:
  “Addresses the challenges of achieving predictable, cost-effective, and reliable systems through techniques in fault tolerance, fault avoidance, verification, designing for upgrade and evolution, and developing a critical mass of techniques for the quality engineering of real-time systems.”
COTS-Based Systems and Dependable System Upgrade

- Conferences included:
  - COTS: What Have We Learned So Far?
  - COTS-Based Systems: Program Manager Keys to Success
  - Competency and Contingency for COTS-Based Systems
  - Picking the Right Product
  - A Process for COTS-Based System Development
COTS-Based Systems and Dependable System Upgrade

Conferences included:

- Engineering for Quality Using Model-Based Verification Techniques
- Adaptive Resource Management and Real-Time System Engineering
- Experiences with Model-Based Verification of Complex Systems
- Issues in Real-Time Systems Engineering
- Improving Predictability in Embedded Real-Time Systems
Process Improvement

- Capability Maturity Model Integration (CMMI) Initiative:
  “Developing an integrated Capability Maturity Model (CMM) framework and product suite that provide industry and government with support for process and product improvement across disciplines.”
Process Improvement

- Team Software Process (TSP) Initiative:
  “Builds on the foundations of the CMM and Personal Software Process (PSP) methods to guide organizations in forming and managing integrated product teams that consistently meet aggressive schedules and planned costs to deliver virtually defect-free software.”
Process Improvement

- Essentials of a Successful Process Deployment Process
- Organizational Performance Management
- Living at Level 3 of the People Capability Maturity Model
- It Wasn’t Our Fault: Lessons Learned in Systems Acquisition
- Honeywell PSP Deployment Strategy
Process Improvement

Conferences included:

- Enabling Statistical Process Control and Six-Sigma Software
- Implementing Six-Sigma in Software
- Introducing the Team Software Process
- Integrating the PSP into Your Organization
- Managing Risk with the Team Software Process
- Managing Software Projects with Metrics
- SPC as a Method of Continual Improvements
Observations

- There’s work being done at the SEI along with its supporters in multiple areas of software engineering.
- Many of the ideas are still theoretical and haven’t been exercised in the private sector.
- There is enough pressure in the public sector that organizations are actively pursuing some of these ideas such as PSP, TSP, and the CMM.
Observations

- Many of the processes are trying to quantify things that we haven’t been able to quantify.
- Most of the focus is on the software development process, rather than the tools and technologies.
- Emphasis is on the *engineering* component rather than on the *software* component.