GSAW 2006

Mission Assurance-Driven Processes for Software-Intensive Ground Systems

Suellen Eslinger
The Aerospace Corporation

March 28, 2006

THE AEROSPACE CORPORATION

Mission Assurance: National Security Space (NSS) Definitions*

- **Mission Success**
  - ❖ The acquisition and operations of systems that meet specified requirements and user expectations in terms of end-to-end operational effectiveness, operability, and supportability

- **Mission Assurance**
  - ❖ The disciplined application of engineering, acquisition, and management principles, processes, and standards to achieve mission success

* “Mission Assurance Standards and Practices,” National Reconnaissance Office, NROD 65-7, 8 October 2004, Unclassified.

The Mission Assurance Problem for Software-Intensive Systems

- **NSS system acquisition failures attributable to software continue to mount, especially for the large, software-intensive ground systems**
  - ❖ Performance deficiencies
  - ❖ Extensive software defects
  - ❖ Large, unanticipated cost and schedule overruns
- **These failures contribute to a lack of mission success for these critical national security programs**
- **However, many of the development contractors for these space systems advertise high maturity levels**
  - ❖ Levels 3, 4 and 5 when appraised against the Capability Maturity Model® Integration^{SM} (CMMI®)

Process Improvement Using the CMMI®

- The CMMI® is a generic model, designed to be useful for process improvement in all product domains and for multiple disciplines (e.g., systems engineering, software engineering)
- Therefore, the CMMI® provides great latitude in how its expected practices can be implemented to meet its stated goals
- In Levels 4 and 5, process improvement is based on "quality and process performance objectives"*, which are driven by the organization's business objectives

* "Capability Maturity Model® Integration (CMMI®), Version 1.1 (CMMI-SE/SW/IPPD/SS, V1.1), Staged Representation, Software Engineering Institute (CMU/SEI-2002-TR-012), March 2002. For example, see pp. 27, 482, 486, 496, and 524.

CMMI® Process Areas Needed for Mission Assurance for Software-Intensive Systems

- **The principal CMMI® process areas (PAs) needed for mission assurance consist of the following Level 2 and 3 PAs:**
  - ❖ Engineering
    - – Requirements Management, Requirements Development, Technical Solution, Product Integration, Verification, Validation
  - ❖ Support
    - – Configuration Management, Product and Process Quality Assurance
  - ❖ Project Management
    - – Risk Management, Integrated Supplier Management
- **However, mission assurance goes well beyond the CMMI® expected practices in these PAs!**
  - ❖ **Processes also need to be effective in producing high quality products that will not require significant downstream rework**
- **CMMI® Level 4 and 5 process areas are not required to achieve mission success**
  - ❖ Level 4 and 5 can certainly help, but only if the organization's "quality and process-performance objectives" are targeted to mission assurance

Ensuring High Mission Assurance Processes

Question: How can you ensure that high mission assurance processes are used to develop your software-intensive ground system?

Answer: Use a robust software development standard!

- **For developers, this means establishing processes that go beyond the minimum requirements of the CMMI®**
  - ❖ Mission assurance-driven processes
  - ❖ Not just mature processes (i.e., institutionalized, predictable) based on CMMI® expected practices
- **For the government, this means making a robust software standard contractually compliant**
- **The CMMI® is NOT a standard!**

The Software Development Standard for Space Systems (SDSSS)

- **The military standard for software development (MIL-STD-498) has now been updated to include explicit requirements for high mission assurance processes**
  - ❖ Part of the SMC/NRO Mission Assurance Improvement Task Force (MAITF) effort
- **The standard is now being applied on new SMC and NRO contracts and is approved for public release**
  - ❖ R. J. Adams et al, “Software Development Standard for Space Systems,” The Aerospace Corporation, TOR-2004(3909)-3537 Revision B, 11 March 2005
- **Don’t let the title be misleading!**
  - ❖ The standard applies to ground software development as well as onboard software development
  - ❖ The standard is not space-specific—It applies to any software development effort where mission assurance is a concern

Mission Assurance-Driven Processes

- **This presentation will focus on two categories of processes critical to mission success**

1. **Testing* Activities**
   - Software unit testing
   - Software integration testing, including
     - ➢ Software unit integration testing
     - ➢ Software/hardware integration testing
     - ➢ Both within and across software items
   - Software qualification testing

2. **Quality-Enhancing Activities**
   - Peer reviews
   - Product evaluations

* The word “testing” in this presentation includes the use of all verification methods (I, A, D, T)

Software Unit Testing

SDSSS

- **Unit testing required for each software unit**
- **Exit criteria specified for unit testing, e.g.,**
  - ❖ All statements and branches
  - ❖ Error and exception handling
  - ❖ Interfaces, including boundary and limit conditions
  - ❖ Algorithms
- **Regression testing of affected unit test cases required for all changes to previously tested software**
- **Conditions specified for unit testing of reuse software**

CMMI®

- **Unit testing NOT required or even expected**
  - ❖ Subpractice 4 in the Technical Solution PA under the expected practice SP 3.1: "Implement the designs of the product components" states "Perform unit testing of the product component as appropriate"**
  - ❖ Subpractices are part of the informative material (not required or expected)

** All quotations from the CMMI® in this presentation are taken from the following reference:

"Capability Maturity Model® Integration (CMMI®), Version 1.1 (CMMI-SE/SW/IPPD/SS, V1.1), Staged Representation, Software Engineering Institute (CMU/SEI-2002-TR-012), March 2002.

Software Integration Testing

SDSSS

- **Software integration testing required**
  - ❖ On target hardware, under conditions as close to operations as possible
- **Exit criteria specified for software integration testing, e.g.,**
  - ❖ Interfaces, including limits and boundary conditions
  - ❖ Integrated error and exception handling
  - ❖ End-to-end functional capabilities
  - ❖ Start up, termination, restart
  - ❖ Verification of software requirements allocated to the integrated units
  - ❖ Performance testing; stress testing
  - ❖ Fault detection, isolation and recovery
  - ❖ Resource utilization measurement
- **Regression testing of affected software integration test cases required for all changes to previously tested software**
- **Conditions specified for software integration testing of Commercial Off-the-Shelf (COTS) and reuse software**

CMMI®

- **Addressed by three expected practices in the Product Integration PA**
- **Software unit integration testing not explicitly required**
  1. SP 1.3: “Establish and maintain procedures and criteria for integration of the product components”
  2. SP 3.2: “Assemble product components according to the product integration sequence and available procedures”
  3. SP 3.3: “Evaluate assembled product components for interface compatibility”

Software Qualification Testing

SDSSS

- **Software qualification testing required**
  - ❖ On target hardware, under conditions as close to operations as possible
- **Exit criteria specified for software qualification testing, e.g.,**
  - ❖ Verification of all software requirements, software interface requirements, software specialty engineering requirements
  - ❖ Stress testing
  - ❖ Resource utilization measurement
  - ❖ Verification of all software requirements allocated to COTS and reuse (modified or unmodified) software
- **Regression testing of affected software qualification test cases required for all changes to previously tested software**
- **People responsible for software qualification testing cannot be the developers of the software unit under test**

CMMI®

- **Addressed by four expected practices in the Verification PA**
- **Software qualification testing not explicitly required**
  1. SP 1.1: "Select the work products to be verified and the verification methods that will be used for each"
  2. SP 1.3: "Establish and maintain verification procedures and criteria for the selected work products"
  3. SP 3.1: "Perform verification on the selected work products"
  4. SP 3.2: "Analyze the results of all verification activities and identify corrective action"

Peer Reviews

SDSSS

- **Peer reviews of work products required**
- **Specific requirements include**
  - ❖ Identifying type of peer review
  - ❖ Identifying mandatory key reviewers
  - ❖ Ensuring entry criteria met
  - ❖ Reviewing materials by each reviewer before the meeting
  - ❖ Identifying and documenting defects and other issues
  - ❖ Recording results of peer review, including action items
  - ❖ Ensuring exit criteria met
  - ❖ Analyzing data about preparation, conduct and results of the peer reviews

CMMI®

- **Addressed by three expected practices and one goal in the Verification PA**
- **Goal: "Peer reviews are performed on selected work products"**
- **Expected Practices:**
  - ❖ SP2.1: "Prepare for peer reviews of selected practices"
  - ❖ SP 2.2: "Conduct peer reviews on selected work products and identify issues resulting from the peer review"
  - ❖ SP 2.3: "Analyze data about preparation, conduct and results of the peer reviews"

Software Product Evaluations

SDSSS

- **In-progress and final software product evaluations required for all software products in the standard**
- **Criteria specified for each type of software product, e.g.,**
  - ❖ Adequate, accurate, consistent, complete, feasible, testable, understandable
  - ❖ Meets requirements (technical and contractual)
  - ❖ Follows Software Development Plan
- **Independence in software product evaluation required**
  - ❖ Cannot be performed by the developers of the product

CMMI®

- **Software product evaluations NOT explicitly addressed**
  - ❖ The Verification PA addresses ensuring that selected work products meet their specified products
  - ❖ However, only peer reviews are required
  - ❖ All other examples of verification methods for software are types of testing (informative material)

Conclusion

High maturity level processes are NOT the same as high mission assurance processes!

- **Mandating use of a robust software development standard will help ensure that high mission assurance processes are applied to your software-intensive ground system development**
  - ❖ SDSSS is an update of MIL-STD-498 that includes mission assurance-related requirements
- **Incentivizing mission success and the use of high mission assurance processes will help**
  - ❖ Align the development contractor’s business objectives with the government’s business objectives
  - ❖ Ensure the development contractor’s process improvement efforts are targeted toward mission success

Backup Charts

SDSSS Exit Criteria For Unit Testing

- **Unit test cases shall fully cover correct execution of all:**
  - ❖ Statements and branches
  - ❖ Error and exception handling
  - ❖ Software unit interfaces, including limits and boundary conditions
  - ❖ Start up, termination and restart (where applicable)
  - ❖ Algorithms
- **Reuse software shall be unit tested for all:**
  - ❖ Modified units
  - ❖ Units where the track record indicates potential problems, even if the units have not been modified
  - ❖ Units performing a critical function
- **Regression testing of affected software unit test cases shall be performed after any modification to previously tested software**

SDSSS Exit Criteria For Software Unit Integration Testing

- **Unit integration test cases shall cover:**
  - Correct execution of all
    - Interfaces between software units, including limit and boundary conditions
    - Integrated error and exception handling across the software units under test
    - End-to-end functional capabilities through the software units under test
    - Start up, termination and restart (where applicable)
  - Verification of all software requirements allocated to the units under test
  - Performance testing, including operational input and output data rates and timing and accuracy requirements
  - Stress testing, including worst-case scenario(s)
  - Fault detection, isolation, and recovery handling (e.g., fault tolerance, fail over, data capture and reporting)
  - Resource utilization measurement (e.g., CPU, memory, storage, bandwidth)

SDSSS Exit Criteria For Software Unit Integration Testing (Cont.)

- Wherever possible, software unit integration testing shall be performed on the target hardware in a configuration as close as possible to the operational configuration
- All COTS and reuse software, including both modified and unmodified reuse, shall undergo software unit integration testing
- Regression testing of affected software unit integration test cases shall be performed after any modification to previously tested software

SDSSS Exit Criteria For SW/HW Integration Testing

- **SW/HW integration test cases shall fully address:**
  - ❖ Correct execution of all
    - – Software-to-software and software-to-hardware interfaces among the hardware and software items under test, including limit and boundary conditions
    - – Integrated error and exception handling across the hardware and software items under test
    - – End-to-end functional capabilities through the software units under test
    - – Start up, termination and restart (where applicable)
  - ❖ Verification of all software and higher level requirements allocated to the software and hardware items under test
  - ❖ Performance testing, including operational input and output data rates and timing and accuracy requirements
  - ❖ Stress testing, including worst-case scenario(s)
  - ❖ Fault detection, isolation, and recovery handling (e.g., fault tolerance, fail over, data capture and reporting)
  - ❖ Resource utilization measurement (e.g. CPU, memory, storage, bandwidth)

SDSSS Exit Criteria For SW/HW Integration Testing (Cont.)

- Wherever possible, SW/HW integration testing shall be performed using target hardware that is as close as possible to the operational target hardware and is in a configuration as close as possible to the operational configuration
- All COTS and reuse software, including both modified and unmodified reuse, shall undergo SW/HW integration testing
- Regression testing of affected SW/HW integration test cases shall be performed after any modification to previously tested software

SDSSS Exit Criteria For Software Qualification Testing

- **Software qualification test cases shall fully address:**
  - ❖ Verification of all software requirements under conditions as close as possible to those that the software will encounter in the operational environment
    - – e.g., operational data constants, operational input and output data rates, operational scenarios, target hardware configurations
  - ❖ Verification of all software interface requirements, using the actual interfaces wherever possible or high fidelity simulation of the interfaces where not possible
  - ❖ Verification of all software specialty engineering requirements
    - – e.g., supportability, testability, dependability/reliability/ maintainability/availability, safety, security, and human systems integration, as applicable
    - – Including, in particular, verification of software reliability requirements and fault detection, isolation, and recovery requirements
  - ❖ Stress testing, including worst-case scenario(s)
  - ❖ Resource utilization measurement (e.g., CPU, memory, storage, bandwidth)

SDSSS Exit Criteria For Software Qualification Testing (Cont.)

- All software requirements shall be verified by software qualification testing whether they are satisfied by COTS, reuse (modified or unmodified) or newly developed software
- Regression testing of affected software qualification test cases shall be performed after any modification to previously tested software

Acronyms and Abbreviations - 1

CMMI®	Capability Maturity Model® Integration^{SM}
CMU	Carnegie Mellon University
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
GSAW	Ground System Architecture Workshop
HW	Hardware
I, A, D, T	Inspection, Analysis, Demonstration and Test
IPPD	Integrated Product and Process Development
MAITF	Mission Assurance Improvement Task Force
MIL	Military
NRO	National Reconnaissance Office
NSS	National Security Space
PA	Process Area

Acronyms and Abbreviations - 2

SDSSS	Software Development Standard for Space Systems
SE	Systems Engineering
SEI	Software Engineering Institute
SM	Service Mark
SMC	Space and Missile Systems Center
SP	Specific Practice
SS	Supplier Sourcing
STD	Standard
SW	Software
TOR	Technical Operating Report
TR	Technical Report
U. S.	United States

Use of Trademarks, Service Marks and Trade Names

Use of any trademarks in this material is not intended in any way to infringe on the rights of the trademark holder. All trademarks, service marks, and trade names are the property of their respective owners.