7 Processes that Enable NASA Software Engineering Technologies

Value-Added Process Engineering

February 2011
Agenda

- Agency Process Requirements
- 7 Software Engineering Processes
  - Purpose, Benefits, and Experiences
- Honorable Mention
NASA’s Software Engineering Requirements

- Software engineering is a core capability and key enabling technology for NASA's missions and supporting infrastructure

- NASA Software Engineering Requirements (NPR 7150.2A)
  - Provide a minimal set of requirements established by the Agency for software
    - Applies to all software created by or for NASA – during all phases
    - For use by both the contractor and in-house communities
  - Support NASA programs/projects to accomplish planned goals (e.g., mission success, safety, schedule, and budget) while satisfying specified requirements
  - Are implemented through Center-specific process definition documents
NPR 7150.2A CMMI Requirement

[SWE-032] The project shall ensure that software is acquired, developed and maintained by an organization with a non-expired Capability Maturity Model Integration® for Development (CMMI-DEV) rating as measured by a Software Engineering Institute (SEI) authorized lead appraiser as follows:

- For Class A software: CMMI-DEV Maturity Level 3 Rating or higher for software, or CMMI-DEV Capability Level 3 Rating or higher in all CMMI-DEV Maturity Level 2 and Maturity Level 3 process areas for software.

- For Class B software: CMMI-DEV Maturity Level 2 Rating or higher for software, or CMMI-DEV Capability Level 2 Rating or higher for all Maturity Level 2 process areas.

- For Class C software: The required CMMI-DEV Maturity Level for Class C software will be defined per Center or project requirements.
<table>
<thead>
<tr>
<th>Year</th>
<th>CMM Level 2</th>
<th>CMM Level 3</th>
<th>CMMI Level 2</th>
<th>CMMI Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>MSFC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>JSC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2003</td>
<td>LaRC/ARC</td>
<td>MSFC/ARC*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>GRC</td>
<td>JSC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>JPL/JSC</td>
<td></td>
<td>MSFC</td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td></td>
<td></td>
<td></td>
<td>GSFC</td>
</tr>
<tr>
<td>2007</td>
<td></td>
<td></td>
<td></td>
<td>MSFC/JPL</td>
</tr>
<tr>
<td>2008</td>
<td></td>
<td>LaRC (FSSB)</td>
<td>LaRC (FSSB)*</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td>JSC/KSC/LaRC</td>
<td>LaRC (SDAB)*</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
<td>MSFC (SIL)/ARC/GRC</td>
<td></td>
<td>MSFC (FSW)/JPL</td>
</tr>
</tbody>
</table>

* Partial implementation
What's your frustration?

- Lack of planning?
- Vague requirements?
- Poor Quality?

Let’s look at a few of our favorite processes!
Product integration is the assembly of software components to ensure correct product functionality.

Product integration is:
- a highly critical and non-trivial part of the development
- frequently overlooked during planning phase

Critical elements of product integration include:
- defining and implementing the integration environment
- management of interfaces
- component integration sequences
- communication between stakeholders

For software systems, integration is typically the first opportunity to observe implementation results.
# 7 Product Integration

- **Benefits**
  - Exposes and drives out defects prior to formal testing
    - Reduces costs for error correction and re-testing
    - Can reduce the length of formal testing (fewer error corrections necessary)
  - Last opportunity to create new functionality before formal test begins
  - Encourages well-defined interfaces and components for easier inspection, integration, and automation
  - Increases the probability for high quality products and timely deliveries to verification and validation activities

February 2011
# 7 Product Integration

- Who makes it happen?

- Test Team
- Design/Implementation
- Project Management
- Configuration Management
# 6 Configuration Management

- Configuration Management (CM) establishes and maintains the integrity of specified work products
  - Typically the most misunderstood and under appreciated process
- Fundamental CM involves ...
  - Identifying configuration items
  - Controlling changes to configuration items
  - Establishing a CM system that supports control objectives
  - Sustaining integrity of baseline products
  - Maintaining accurate status of configuration data
# 6 Configuration Management

**Benefits**

- Baselines provide a stable foundation for continuing evolution of specified work products

**Build Variance Detection**
- Knowing the last known good build, changes can be effectively detected and examined or rolling back to the last known good configuration can be achieved

**Effective Change Management**
- Knowing the configuration of a given CI saves time that would be spent figuring out the configuration versus being able to immediately engineer the change in the known configuration
# 6 Configuration Management

- **Benefits (cont’d)**
  - Enhanced Ability to Rebuild
    - If a CI fails or is involved in a disaster, it is far easier to rebuild if the final production build of the CI is known
  - Assists with Cost/Schedule Estimating
    - Understanding what software goes into a given CI allows for proper costing to serve as an input to planning and estimating process
# 6 Configuration Management

- Who makes it happen?

- Configuration Management
- Requirements Team
- Design/Implementation
- Test Team
- Everybody!!
# 5 Verification

- Software verification is a broad and complex software engineering discipline that ensures transitional and final work products adhere to their specified requirements.

- Growth in complexity of designs increases the importance of formal verification techniques.

- Key concepts include ...  
  - Select verification work products  
  - Establish verification environment/procedures/criteria  
  - Perform verification
# 5 Verification

**Benefits**

- Requirements Phase – Ensure requirements are verifiable, achievable, actionable, measurable, related to identified business needs, and defined to a level of detail sufficient for system design

- Design Phase – Review/analysis using models, simulations, and prototypes
# 5 Verification

- Benefits (cont’d)
  - Implementation Phase – Analysis to help detect complexity, memory, arithmetic exception, out-of-bounds array access, and coding standard problems
  - Test Phase – verifies software as implemented. It addresses specified requirements and ONLY specified requirements
  - Peer Reviews - one of the most effective methods of verification since they improve product quality by detecting errors as early as possible
# 5 Verification

- Who makes it happen?

- Configuration Management
- Requirements Team
- Design/Implementation
- Test Team
- Project Management

Everybody!!!
# 4 Software Assurance

- Product assurance provides management and staff an objective evaluation of organizational processes and associated work products

- Key concepts include ...
  - Objectively evaluate processes/products against specified standards
  - Document non-compliance issues and provide feedback to management and staff
  - Ensure non-compliances are addressed
# 4 Software Assurance

• Benefits

• Product assurance provides insight into process implementation as compared to process definition
  • Identifies process improvement opportunities
  • Monitors process implementation effectiveness

• Ensures critical work products align with specified standards in support of customer/contract requirements

• Provides management with visibility into process effectiveness and product quality
# 4 Software Assurance

- Who makes it happen?

Requirements Team

Design/Implementation

Configuration Management

Test Team

Software Assurance
Measurement and Analysis

- Measurement and analysis defines and maintains a measurement capability that supports management information needs as they relate to mission objectives.

- Key concepts include ...
  - Identify mission objectives
  - Derive measures from mission objectives
  - Select analysis techniques
  - Define data collection, storage, and reporting mechanism
# 3 Measurement and Analysis

- **Benefits**
  - Provides quantitative determination of how well you are doing relative to mission objectives, other projects, the past, and/or the plan
  - Provides a mechanism to monitor selected aspects of a project to provide timely information for management decision making
  - Improves communication
  - Encourages appropriate behavior
  - Pinpoints opportunities for improvement
# 3 Measurement and Analysis

- Who makes it happen?

Project Leads
# 2 Requirements Management

- Requirements management documents and verifies requirements and requirements changes that meet customer expectations.

- Key concepts include ...
  - Understand operational concepts and system-level requirements.
  - Establish and manage changes to detailed software requirements.
  - Maintain bi-directional traceability.
  - Identify inconsistencies between requirements and work products.
# 2 Requirements Management

**Benefits**

- Encourages development of high-quality requirements and elicitation of requirements from customers
- Bi-directional traceability enables close evaluation to eliminate lower level requirements that do not support mission requirements
- Allows detailed requirements definition and tracking to ensure product completeness
- Enables requirements change management to ensure product lifecycle integrity
- Helps avoid requirement creep
# 2 Requirements Management

- Who makes it happen?

Design/Implementation

Requirements Team

Test Team

Customer/User
# 1 Planning & Monitoring

- Project planning defines and documents the necessary project activities so that they may be monitored to ensure deviations are recognized soon enough to take corrective actions.

- Key concepts include ...
  - Develop and maintain the project plan/schedule
  - Establish work product and task estimates
  - Define communication and monitoring methods
  - Conduct milestone/progress reviews
# 1 Planning & Monitoring

- Key concepts (cont’d)
  - Obtain commitment to the plan
  - Monitor against the plan
    - Estimates
    - Commitments
    - Risks
    - Stakeholder involvement
    - Take corrective actions when necessary
# 1 Planning & Monitoring

- **Benefits**
  - Ensures timely determination of cost/schedule impacts
  - Allows standardization and quantifying of project goals
  - Enables tracking of project schedule milestones
  - Provides insight into technical/cost risk management
  - Identifies stakeholder participation issues
  - Tracks/controls corrective actions to closure
  - Monitors management of project data
# 1 Planning & Monitoring

- Who makes it happen?

Project Management

Project Leads
Honorable Mention

• Keys to keep all these processes working well...
  • Sponsorship (management support)
    • Stress the importance of maintaining good processes
    • Ensure that adequate resources are available to support processes
  • Standardization of processes
    • Have a library of process assets -- process descriptions, tools, templates, lessons learned
    • Use a measurement repository to capture organizational “norms”, improve cost estimation and gauge success of improvements
    • Develop tailoring guidelines to make processes reasonable for all types of projects
CMMI Benefits at NASA

- Reduces risk of software failure increasing mission safety
- More predictable software cost estimates and delivery schedules
- Smarter buyer of contracted software
- More defects found and removed earlier
- Reduces duplication of efforts between projects
- Increases ability to meet the challenges of evolving software technology
- Software development planning improved across the Agency
- NASA’s contractor community has heard the word that the bar has been raised with respect to software engineering and is responding appropriately
Enjoy the journey