Improving Software Engineering on NASA Projects

May 11, 2010
Tim Crumbley & John C. Kelly
Office of Chief Engineer
Objective: “...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA” (plan approved 2002)

Sponsor: NASA Office of Chief Engineer
## FY 2010 Software Improvement Initiative Plans

### Policy & Procedural Requirements

**Ongoing**
- NPD 7120.4* - completed
- NPR 7150.2A, SW Engineering Requirements update - completed
- OCE Survey* (10 Centers + HQ)

**New for 2010**
- SW Engr. Handbook (Electronic)
- Center Compliance with new NPR 7150.2A (Phase 1)
- OSMA’s update to NASA Safety and Assurance standards
  - Representative to help develop new Programmable Logic Devices Policy/Std/HB*

**Crosscutting**
- Center SW Improvement Plans
- Training (including NPR 7150.2A Classroom & NASA SATERN)
- NASA Engineering Network*, Software.nasa.gov
- Software Inventory, SIMS Tool, Analysis & Suggestions for projects to receive IV&V
- SWG F2Fs, Leads Meeting, & Telecons
- Communications / Exchanges (CMMI Steering Group, v1.3 CCB, TIMs, etc.)
- Interface to Systems Engineering Working Group

### Processes

**Ongoing**
- CMMI Appraisals
- NASA & Center Process Asset Libraries (PALs)
- SW Measurement

**New for 2010**
- Center processes updated for consistency with new NPR 7150.2A

### Technology

**Ongoing**
- Tool Shed
- Sys & SW Journal
- Reviewers and Rep. to OSMA’s SW Assurance Research Program (SARP)*

**New for 2010**
- Update SW Technology Strategy for 2011 and beyond
- Interface to SW Architecture Review Board effort (NESC)*
- Interface to Multi-Core Flight computing*
- Interface to SW Engineering Research Center (SERC)
- Interface to NASA Aviation Safety Program*

*Software Engineering portions or contributions*
1. Software Requirements
2. Internal NASA-wide requirements (NPD, NPR, & Standards)
3. Training & Skill Development
4. Complex Electronics, FPGA, PLD, etc. (blurring of H/W – S/W boundary)
5. Insight/Oversight of Contractor SW development
6. Tools
7. Empowerment of program/project SW personnel
8. Metrics/Measurement
9. COTS - Impacts of maintaining COTS and technologies for long-term systems and missions
10. Cost estimation - need a standard approach
Top Software Issues from NASA Centers 2010

1. Internal NASA-wide NASA Procedural Requirements & Standards (including Software Classification) (2)
2. Cost Estimation (10)
3. Software Workforce Level (New)
4. Systems Engineering / Software Engineering Interfaces (New)
5. Small Project Implementations (cross cutting) (New)
6. Empowerment of Software Engineering Personnel (7)
7. Software Requirements (1)
8. Complex Electronics (4)
9. Training & Skill Development (3)
10. Insufficient attention to Software on Contracts (New)
Objective:
“…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA” (plan approved 2002)

Sponsor: NASA Office of Chief Engineer
Training to ensure project engineers have necessary skills

Local tools to support projects
### Completed CMMI Appraisals

<table>
<thead>
<tr>
<th>Center</th>
<th>Rating</th>
<th>Date</th>
<th># Projects</th>
<th>Type</th>
<th>Project size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LaRC- ASDC</td>
<td>PP(CL3), CM(CL1)</td>
<td>17-Nov-06</td>
<td>1</td>
<td>Data Center Support</td>
<td>85</td>
</tr>
<tr>
<td>MSFC</td>
<td>ML3</td>
<td>24-Apr-07</td>
<td>3</td>
<td>Development</td>
<td>57,4,2 = 63</td>
</tr>
<tr>
<td>JPL</td>
<td>ML3</td>
<td>27-Sep-07</td>
<td>7</td>
<td>Dev &amp; Maintenance</td>
<td>1,8,11,17,16,1,30 = 84</td>
</tr>
<tr>
<td>GSFC</td>
<td>ML2 + RSKM(2)</td>
<td>16-May-08</td>
<td>4</td>
<td>Dev &amp; Maintenance</td>
<td>25,2,18,8 = 53</td>
</tr>
<tr>
<td>LaRC- FSSB</td>
<td>ML2 + CL3</td>
<td>3-Oct-08</td>
<td>3</td>
<td>Services</td>
<td>1,1,3 = 5</td>
</tr>
<tr>
<td>LaRC- SDAB</td>
<td>PP(CL3), REQM(CL3), CM(CL3), MA(CL3)</td>
<td>13-Mar-09</td>
<td>4</td>
<td>Development</td>
<td>1,5,10,5 = 21</td>
</tr>
<tr>
<td>JSC</td>
<td>ML2</td>
<td>2-Apr-09</td>
<td>4</td>
<td>Development</td>
<td>7,6,45,2 = 60</td>
</tr>
<tr>
<td>KSC</td>
<td>ML2</td>
<td>18-Sep-09</td>
<td>1</td>
<td>Development</td>
<td>225</td>
</tr>
</tbody>
</table>

### Scheduled CMMI Appraisals in FY10

<table>
<thead>
<tr>
<th>SCAMPI A Center</th>
<th>SCAMPI B Center</th>
<th>SCAMPI C Center</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>Month</td>
<td>Center</td>
</tr>
<tr>
<td>MSFC</td>
<td>April</td>
<td>JPL</td>
</tr>
<tr>
<td>ARC</td>
<td>May</td>
<td></td>
</tr>
<tr>
<td>MSFC</td>
<td>June</td>
<td></td>
</tr>
<tr>
<td>GRC</td>
<td>August</td>
<td></td>
</tr>
<tr>
<td>JPL</td>
<td>September</td>
<td></td>
</tr>
</tbody>
</table>
Objective:
“…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA” (plan approved 2002)

Sponsor: NASA Office of Chief Engineer
Chartered Roles of the Software Working Group (SWG)

- Function as an advisory group
- Recommend, draft as requested, review, and promote policies, standards, & best practices
- Recommend and provide technical support for special studies
- Focus, integrate, and promote innovation and the continuous improvement of NASA's software engineering processes
- Support and help guide the establishment of software process improvement programs at each Center
- Facilitate the transfer of software technology
- Coordinate NASA representation within agency, interagency, and international boards
- Provide information to improve communication on software issues
- Ad hoc activities as needed
Objective: “…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA” (plan approved 2002)

Sponsor: NASA Office of Chief Engineer
1.3.1 Higher Agency-Level Requirements

NPD 1000.0, NASA Governance and Strategic Management Handbook.
NPD 1000.3, The NASA Organization.
NPD 1000.5, Policy for NASA Acquisition.

1.3.2 Agency-Level Software Policies and Requirements

NPD 7120.4, NASA Engineering and Program/Project Management Policy

NPD 7120.5, NASA Space Flight Program and Project Management Requirements
NPR 7120.6, Lessons Learned Process
NPR 7120.7, NASA Information Technology and Institutional Infrastructure Program and Project Management Requirements
NPR 7120.8, NASA Research and Technology Program and Project Management Requirements
NPR 7123.1, NASA Systems Engineering Processes and Requirements
NPR 7150.2, NASA Software Engineering Requirements

1.3.3 Agency-Level Multi-Center and Product Line Requirements (non-software specific)

These NPDs and NPRs elaborate, tailor, and in some cases add requirements to the ones above to address the needs of major multi-Center projects, specific product lines, and specific focus areas.

1.3.4 NASA and Industry Software Standards and Guidebooks

NASA Preferred Industry Software Standards and Guidebooks and NASA Software-Related Standards and Guidebooks are required when invoked by an NPD, NPR, Center-Level Directive, contract clause, specification, or statement of work.

1.3.5 Center-Level Directives (related to software)

Center-Level Directives are developed by NASA Centers to document their local software policies, requirements, and procedures.

1.3.6 Government In-house Development

Government in-house software development policies and procedures to provide quality products and to fulfill the requirements passed down by a project.

1.3.7 Contractor and Subcontractor Development

Contractors and subcontractors develop in-house policies and procedures to provide quality products and to fulfill the requirements passed down through a contract by a customer.
Office of Chief Engineer
Completed “Go To Architecture”

Note: NPD 7120.4D assumes NPD 2820.1 role in being the unique NASA NPD that covers software for all offices and organizations. NPD 7120.4 D is the new parent NPD for NPR 2210.1, Release of NASA Software
# NPR 7150.2A TABLE OF CONTENTS

## PREFACE
- P.1 Purpose
- P.2 Applicability and Scope
- P.3 Authority
- P.4 Applicable Documents
- P.5 Measurement/Verification
- P.6 Cancellation

## CHAPTER 1. Introduction
- 1.1 Overview
- 1.2 Organizational Capabilities and Improvement
- 1.3 Hierarchy of NASA Software-Related Documents

## CHAPTER 2. Software Management Requirements
- 2.1 Compliance with Laws, Policies, and Requirements
- 2.2 Software Life-Cycle Planning
- 2.3 Commercial, Government, Legacy/Heritage and Modified Off-The-Shelf Software
- 2.4 Software Verification and Validation
- 2.5 Project Formulation Requirements
- 2.6 Software Contract Requirements

## CHAPTER 3. Software Engineering (Life-Cycle) Requirements
- 3.1 Software Requirements
- 3.2 Software Design
- 3.3 Software Implementation
- 3.4 Software Testing
- 3.5 Software Operations, Maintenance, and Retirement

## CHAPTER 4. Supporting Software Life-Cycle Requirements
- 4.1 Software Configuration Management
- 4.2 Risk Management
- 4.3 Software Peer Reviews/Inspections
- 4.4 Software Measurement
- 4.5 Best Practices
- 4.6 Training

## CHAPTER 5. Software Documentation Requirements
- 5.1 Software Plans
- 5.2 Software Requirements and Product Data
- 5.3 Software Reports

## CHAPTER 6. Tailoring, Engineering Technical Authority, and Compliance Measurement
- 6.1 Tailoring of Requirements
- 6.2 Designation of Engineering Technical Authority(s)
- 6.3 Compliance

## APPENDIX A. Definitions
## APPENDIX B. Acronyms
## APPENDIX C. References
## APPENDIX D. Requirements Mapping Matrix
## APPENDIX E. Software Classifications
Software is not all the same

- flight software ↔ non-flight software
- engineering software ↔ general purpose software
- safety critical software ↔ non-safety critical software

... and it shouldn’t be treated the same!
NASA-wide Software Classification*

Class A  | Space Flight Human Rated Software Systems
Class B  | Non-Human Space Rated Software Systems
Class C  | Mission Support Software & Facilities
Class D  | Analysis and Distribution Software
Class E  | Development Support Software

Class F  | General Purpose Computing Software
         | (Multi-Center or Multi-Program/Project)
Class G  | General Purpose Computing Software
         | (Single Center or Project)
Class H  | General Purpose Desktop Software

Notes 1. “It is not uncommon for a project to contain multiple systems and subsystems having different software classes” (P.2.1)
2. Whether software is safety critical is an independent determination based on NASA-STD 8719.13

* Established by NPR 7150.2
Objective:
“...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA” (plan approved 2002)

Sponsor: NASA Office of Chief Engineer
Software Engineering DACUM

Early Career Courses:
1. Introduction to Aerospace at NASA (IAN)
2. Software Engineering 101
3. Software Requirements Development and Management
3a. Peer Reviews/ Inspections (short separate class)
4. Software Implementation
5. Software Testing

Mid-Career Courses
6. Software Engineering 201
6a. Software Maintenance
7. Software Configuration Management
8. Software Design
9. Software for Embedded Systems
10. Software Safety & Reliability

Later Career Courses
11. Software Engineering 301
12. Software and Process Metrics
13. Software Estimations
14. Software Process Improvement
15. Software Acquisition
16. Formal Methods for Software

Structured On-the-Job Learning
Informal On-the-Job Learning
Software Engineering Initiative

1. Center SEPGs
   Process Improvements
   SW Metrics

2. Software Working Group

3. SW Policy
   NASA NPD 2820.1 & NPR 7150.2

4. Training, CMMI Appraisals
   & Career Development

5. SW Technology Infusion
   & Journal

6. Engineering, Assurance,
   & Safety Collaboration
   NASA SW Process Asset Library
   NASA SW Inventory
   NASA Engineering Network (Software)

Special NASA Software Engineering Studies
   Flight Software Complexity
   Fault Management Workshop
   Programmable Logic Devices (Complex Electronics)

Objective:
“…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA” (plan approved 2002)

Sponsor: NASA Office of Chief Engineer
<table>
<thead>
<tr>
<th>Initiative</th>
<th>PI</th>
<th>Center</th>
<th>The Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Tools and Techniques for V&amp;V of IVHM Systems</td>
<td>Johann Schuman</td>
<td>ARC</td>
<td>Analysis of complex IVHM systems require methods beyond traditional testing – will also add information and perspective to the on-going work on safety case</td>
</tr>
<tr>
<td>Architectural Analysis of Dynamically Reconfigurable Systems</td>
<td>Mikael Lindvall</td>
<td>GSFC</td>
<td>A follow-on to prior work, this initiative will be adjusting the work plan to support MSL and GMSEC</td>
</tr>
<tr>
<td>Command Process Modeling &amp; Risk Analysis</td>
<td>Lelia Meshkat</td>
<td>JPL</td>
<td>The team will be developing tools and techniques to design and analyze robust command/operations process</td>
</tr>
<tr>
<td>Infusion of SDA for Automated Assessment of LADEE</td>
<td>Guille Del Carmin</td>
<td>ARC</td>
<td>Application of SDA (developed under NASA SBIR) on LADEE – collecting qualitative and quantitative information. This tool was previously tested by JSC MOD</td>
</tr>
<tr>
<td>Software Architecture Modeling and Assurance with AADL for the JPL SMAP Project</td>
<td>Katie Weiss</td>
<td>JPL</td>
<td>Application of the AADL approach of SMAP to be run in tandem with current processes</td>
</tr>
<tr>
<td>Toward Clear and Consistent Textual Requirements: An Application of Natural Language Processing Techniques</td>
<td>Allen Nikora</td>
<td>JPL</td>
<td>Developing approaches to support the automated discovery of ambiguous and inconsistent natural language requirements</td>
</tr>
</tbody>
</table>
Research & Technology Infusion Journal

- Encourage and stimulate advanced technology and research work that is relevant to NASA’s missions
- Promote positive communication between the research and practicing professionals
- Supports the maturation of software and systems engineering related to aerospace applications
- Joint editorship between academic and government researchers
- Started in April 2005
Objective:
“…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA” (plan approved 2002)

Sponsor: NASA Office of Chief Engineer
Approach on requirements related to the developmental aspects of safety critical software
2010 (NASA STD 8719.13 and STD 8739.8 updates)

**NPR 7150.2.A, SW Engineering**

Minimum SW Engineer Requirements base on SW Classifications A – H and software safety criticality

Generic Engineering Design Requirement for safety critical software systems

**NASA SW Assurance and Safety Standards**

Requirements for identifying and applying SW Assurance methods

Requirements to implement a systematic approach for software safety*

Set of SW safety requirements (and level of direction) beyond those found in NPR 7150.2 A

**Specific Program and Project Requirements (w/Human Spaceflight track record)**

Include program/project specific SW safety requirements as well as the implementation of NPR 7150.2A, Sec. 2.2.12

**Solution:** Harmonize Assurance & Safety Standards with NPR 7150.2A to resolve confusion over redundant aspects of documents

*Note: The inclusion of some safety requirements in NPR 7150.2A does not relieve projects from complying with NASA STD 8719.13 and STD 8739.8

* Safety identification, assurance, risk & hazards analysis, FMEA,.... remain in NASA STD 8719.13.*
The NASA PAL has two goals:

Encourage sharing of software engineering assets within the agency.

Provide a repository of software engineering needs.

The NASA PAL contains pointers to over 650 assets

Effective communication of templates, guides, checklist, and procedures between Centers

Encourage Centers to continue to share their software related assets across the Agency
Software Tool Shed
Focus Area/Concept/Objective

- Deploy “in-house”, open source and COTS software tools (static analysis) on NASA mission software.
- Demonstrate the feasibility of an Application Service Provider model that provides missions with software tools and expertise.
- Taking a general approach to ASP; Language independent but a focus on Java with some C/C++ capability.
- Recognizing the need for “bug” reporting strategies for developers and managers.
- Building a relationship between NASA software engineering research, applied software engineering and mission software development (infusion and requirements).
Objective:
“…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA” (plan approved 2002)

Sponsor: NASA Office of Chief Engineer
NASA Study

Flight Software Complexity
Scope, Findings, Observations

- Challenging requirements raise downstream complexity (unavoidable)
- Lack of requirements rationale permit unnecessary requirements

- Engineering trade studies not done: a missed opportunity
- Architectural thinking/review needed at level of systems and software

- Inadequate software architecture and lack of design patterns
- Coding guidelines help reduce defects and improve static analysis
- Descopes often shift complexity to operations

- Growth in testing complexity seen at all centers
- More software components and interactions to test
- COTS software is a mixed blessing

- Shortsighted FSW decisions make operations unnecessarily complex
- Numerous “operational workarounds” raise risk of command errors
Programmable Logic Devices (Complex Electronics) NESC Problem Description

- Non descript discipline terms ("firmware", "software" & "hardware") have been used to describe a complicated device, which creates confusion
  - *Is an FPGA/ASIC containing a microprocessor function and associated code a hardware or software system?*
- No known single NASA-wide set of procedures, policy and/or guidelines exists for the design, development, test, and evaluation (DDT&E) of FPGA/ASICs for space flight applications.
- Historically, the application design’s operational speed and complexity has increased concurrently with the size of the circuitry decreasing
  - The single integrated circuit gives the appearance of minimal complexity
  - Past experience has uncovered undesirable features existing in designs

- This situation has all the ingredients of a pending accident
  - Complex design with critical functions + Difficultly in thoroughly testing all combinational logic modes + Varying DDT&E process + “It is only a chip” paradigm
Software Engineering Initiative

• Reduces risk of software failure - Increases mission safety
• More predictable software cost estimates and delivery schedules
• Smarter buyer of contracted out software
• More defects found and removed earlier
• Reduces duplication of efforts between projects
• Increases ability to meet the challenges of evolving software technology