7.18 - Documentation Guidance

This version of SWEHB is associated with NPR 7150.2B. Click for the latest version of the SWEHB based on NPR7150.2C

1. Purpose

This topic provides a set of minimum content guidance for software project plans, reports, and procedures. This guidance previously appeared in the version of this Handbook associated with NPR 7150.2A, which contained requirements for software documentation. Those requirements have been removed from the NPR and incorporated into the guidance provided in this Handbook topic.

1.1 Introduction

Guidance for each document is provided via the linked pages below. Software documentation resources and lessons learned are also reiterated on those respective pages.

The tab labels are abbreviated as follows:

- CR-PR - Software Change Request - Problem Report
- CSIP - Center Software Improvement Plan
- IDD - Interface Design Description
- Inspect - Software Inspection, Peer Reviews, Inspections
- Maint - Software Maintenance Plan
- Metrics - Software Metrics Report
- Safety - Software Safety Plan
- SAP - Software Assurance Plan
- SCMP - Software Configuration Management Plan
- SDD - Software Data Dictionary
- SDP-SMP - Software Development - Management Plan
- SRS - Software Requirements Specification
- STP - Software Test Plan
- STR - Software Test Report
- SUM - Software User Manual
- SVD - Software Version Description
- SwDD - Software Design Description
- Test - Software Test Procedures
- Train - Software Training Plan

To supplement and implement this guidance, NASA-specific documentation templates, examples, checklists, and more are available in Software Processes Across NASA (SPAN), accessible to NASA users from the SPAN tab in this Handbook. Center-specific guidance and resources, such as templates, are available in Center Process Asset Libraries (PALs).

2. References

Click here to view master references table.

- (SWEREF-025)


- (SWEREF-031)

- (SWEREF-038)


Software Engineering Improvement Plan FY12, NASA Langley Research Center (LaRC), 2012. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA-users from the SPAN tab in this Handbook.

Software Management Plan (SMP) Template, GRC-SW-TPLT-SMP, NASA Glenn Research Center (GRC), 2009. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA users from the SPAN tab in this Handbook.


Software User Manual (SUM) Template, GRC-SW-TPLT-SUM, NASA Glenn Research Center (GRC), 2011. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA-users from the SPAN tab in this Handbook.

Checklist for the Contents of Software Critical Design Review (CDR), 580-CK-008-02, Software Engineering Division, NASA Goddard Space Flight Center (GSFC), 2010. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA-users from the SPAN tab in this Handbook.

Checklist for the Contents of Software Preliminary Design Review (PDR), 580-CK-007-02, Software Engineering Division, NASA Goddard Space Flight Center (GSFC), 2010. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA-users from the SPAN tab in this Handbook.


NASA Space Flight Program and Project Management Requirements, w/Changes 1-16 NPR 7120.5E, NASA Office of the Chief Engineer, 2012. Effective Date: August 14, 2012, Expiration Date: August 14, 2019

Software Design Description Template, GRC-SW-TPLT-SDD, NASA Glenn Research Center (GRC), 2011. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA-users from the SPAN tab in this Handbook.


An Axiomatic Approach of Software Functionality Measure. Chen, Yuan-Tsong; Tanik, Murat M.; Southern Methodist University, Dallas, TX IEEE0-8186-3520-7/92, 1992. Published in: 1992 Proceedings - The Third International Workshop on Rapid System Prototyping


SOFTWARE ASSURANCE STANDARD (Baseline w/ Ch 1 of 5/5/05) Baseline w/ Ch 1 of 5/5/05

Software Requirements Specification (SRS) Template, GRC-SW-TPLT-SRS, NASA Glenn Research Center, 2011. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA users from the SPAN tab in this Handbook.


OSMA/NSC's SMA Technical Excellence Program (STEP) program.


STEP Level 2 Overview of Software Safety course, SMA-SA-WBT-230 SATERN (need user account to access SATERN courses). This NASA-specific information and resource is available in at the System for Administration, Training, and Educational Resources for NASA (SATERN), accessible to NASA-users at https://saterninfo.nasa.gov/.

STEP Level 2 Software Configuration Management and Data Management course, SMA-SA-WBT-204, SATERN (need user account to access SATERN courses). This NASA-specific information and resource is available in at the System for Administration, Training, and Educational Resources for NASA (SATERN), accessible to NASA-users at https://saterninfo.nasa.gov/.
• (SWEREF-347)


• (SWEREF-355)


• (SWEREF-365)


• (SWEREF-370)


• (SWEREF-373)

NPR 2210.1C - Release of NASA Software - Revalidated w/change 1 NASA Office of the Chief Technologist, 2010. NPR 2210.1C, Expiration Date: August 11, 2020

• (SWEREF-381) Software Requirements Specification (SRS) Template, GRC-SW-TPLT-SRS, NASA Glenn Research Center, 2011. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA users from the SPAN tab in this Handbook.

• (SWEREF-401)

Software Test Plan (STP). Federal Aviation Administration (FAA), December, 1999. DI-IPSC-81438A,

• (SWEREF-402)

NASA Information Security Policy. NASA Office of the Chief Information Officer, May, 2015. NPD 2810.1E,

• (SWEREF-403)

Security of Information Technology (Revalidated with Change 1, dated May 19, 2011). NASA Office of the Chief Information Officer, May 2006. NPR 2810.1A,

• (SWEREF-433)


• (SWEREF-452) SED Unit Test Guideline, 580-GL-062-02, Systems Engineering Division, NASA Goddard Space Flight Center (GSFC), 2012. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA-users from the SPAN tab in this Handbook. Replaces SWEREF-081

• (SWEREF-453) Project Planning Process, 580-PC-004-05, Software Engineering Division (ISD), NASA Goddard Space Flight Center (GSFC), 2014. This NASA-specific information and resource is available in Software Processes Across NASA (SPAN), accessible to NASA-users from the SPAN tab in this Handbook. Replaces SWEREF-088

• (SWEREF-490)


• (SWEREF-502)

Quality Assurance Expertise in Special Technical Areas (e.g., optics) Public Lessons Learned Entry: 331.

• (SWEREF-503)


• (SWEREF-507)

Thrusters Fired on Launch Pad (1975) Public Lessons Learned Entry: 403.

• (SWEREF-508)

Interface Control and Verification Public Lessons Learned Entry: 569.

• (SWEREF-511)

Consider Language Differences When Conveying Requirements to Foreign Partners (1997) Public Lessons Learned Entry: 608.

• (SWEREF-513)
Mars Climate Orbiter Mishap Investigation Board - Phase I Report  Public Lessons Learned Entry: 641.

- (SWEREF-519) Pre-Flight Problem/Failure Reporting Procedures  Public Lessons Learned Entry: 733.
- (SWEREF-520) Problem Reporting and Corrective Action System  Public Lessons Learned Entry: 738.
- (SWEREF-536) International Space Station (ISS) Program/Computer Hardware-Software/Software  Public Lessons Learned Entry: 1062.
- (SWEREF-544) Lack of Education and Training in the Use and Processes of Independent Verification & Validation (IV&V) for Software Within NASA  Public Lessons Learned Entry: 1173.
- (SWEREF-552) Kennedy Space Center (KSC) Projects and Resources Online (KPRO) Software Development and Implementation  Public Lessons Learned Entry: 1384.
- (SWEREF-574) Place Flight Scripts Under Configuration Management Prior to ORT  Public Lessons Learned Entry: 2476.

2.1 Tools

Tools relative to this Topic may be found in the table below. You may wish to reference the Tools Table in this handbook for an evolving list of these and other tools in use at NASA. Note that this table should not be considered all-inclusive, nor is it an endorsement of any particular tool. Check with your Center to see what tools are available to facilitate compliance with this requirement.

<table>
<thead>
<tr>
<th>Tool name</th>
<th>Type</th>
<th>Owner/Source</th>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COTS</td>
<td>COTS</td>
<td></td>
<td><a href="https://www">https://www</a>.</td>
<td>UML business process, architecture, software and system modeling tool with teamwork support.</td>
</tr>
</tbody>
</table>
3. Lessons Learned

The NASA Lesson Learned database contains the following lessons learned related to software project documentation:

- **Computer Hardware-Software/International Space Station/Software Development (Plan for user involvement). Lesson Learned 1132:** "The lack of user involvement results in increased schedule and safety risk to the program... follow a concurrent engineering approach to building software that involves users and other key discipline specialists early in the software development process to provide a full range of perspectives and improve the understanding of requirements before code is developed." 542

- **Computer Software/Software Safety Policy Requirements/Potential Inadequacies (Cover essential requirements for the project). Lesson Learned 1021:** "NASA is committed to assuring that required program management plans and any subordinate plans such as software or safety management plans cover the essential requirements for programs where warranted by cost, size, complexity, lifespan, risk, and consequence of failure." 532

- **Kennedy Space Center (KSC) Projects and Resources Online (KPROM) Software Development and Implementation (Project team planning). Lesson Learned 1384:** "When planning and selecting team resources for a project, consider how the resources can work together and support each other, along with the skills required. This can be a factor in meeting or delaying software project milestones if an alternative resource has not been endorsed by the team members." 552

- **Place Flight Scripts Under Configuration Management Prior to ORT (Project attention to configuration control). Lesson Number 2476:** "Project attention to the configuration control of flight scripts is likely to prevent the generation of unnecessary software iterations, improve the rigor of mission system engineering processes, and ensure consistency in the test and operations environments." 574

- **MPL Uplink Loss Timer Software/Test Errors (1998) (Plan to test against full range of parameters). Lesson Number 0939:** "Unit and integration testing should, at a minimum, test against the full operational range of parameters. When changes are made to database parameters that affect logic decisions, the logic should be re-tested." 530

- **Deep Space 2 Telecom Hardware-Software Interaction (1999) (Plan to test as you fly). Lesson Number 1197:** "To fully validate performance, test integrated software and test integrated hardware over the flight operational temperature range." 545

- **International Space Station (ISS) Program/Computer Hardware-Software/Software (Plan realistic but flexible schedules). Lesson Number 1062:** "NASA should realistically reevaluate the achievable ... software development and test schedule and be willing to delay ... deployment if necessary rather than potentially sacrificing safety." 638

- **Thrusters Fired on Launch Pad (1975) (Plan for safe exercise of command sequences). Lesson Number 0403:** "When command sequences are stored on the spacecraft and intended to be exercised only in the event of abnormal spacecraft activity, the consequences should be considered of their being issued during the system test or the pre-launch phases." 507

- **Quality Assurance Access to Critical Areas, Management (Quality Assurance Access). Lesson Number 0332:** "If program or engineering management exercises the authority to deny QA access to critical areas, QA oversight will be severely [compromised]." [Note that, while this lesson does not specifically call out software, the lesson remains relevant regarding assurance personnel access to information and areas required to perform software assurance activities.] 550

- **Quality Assurance Expertise in Special Technical Areas (e.g., optics) (Quality Assurance Expertise). Lesson Number 0331:** "If quality assurance personnel responsible for oversight of the quality of highly technical state-of-the-art development do not have a degree of expertise in that technical area, the likelihood of discovering QA problems decreases significantly." [Note that, while this lesson does not specifically call out software, the lesson remains relevant regarding assurance personnel access to information and areas required to perform software assurance activities.]

- **International Space Station (ISS) Program/Computer Hardware-Software/Software (Schedules). Lesson Number 1062:** "The ISS software development schedule is almost impossibly tight. If something else does not cause a further delay in ISS deployment, software development may very well do so. The decision this year to add integrated testing of some modules ... is a very positive step for safety. However, there is no room in the schedule for required changes that may be discovered during this testing." 636

- **Lack of Education and Training in the Use and Processes of Independent Verification & Validation (IV&V) for Software Within NASA (2001). Lesson Number 1173:** "While NASA has made major changes to emphasize the need to utilize IV&V on safety critical projects, the technology is not well understood by program managers and other relevant NASA personnel." 544

- **Erroneous Onboard Status Reporting Disabled IMAGE's Radio. Lesson Number 1799:** "The loss of the IMAGE satellite was attributed to a Single Event Upset-induced "instant trip" of the Solid State Power Controller (SSPC) that supplies power to the single-string Transponder. The circuit breaker was not reset because this hybrid device incorrectly reported the circuit breaker as closed, and ground could not command a reset because the satellite's single telemetry receiver had been disabled by the SSPC. The SSPC's problematic state reporting characteristic was an intentional design feature that was not reflected in any part documentation, and three similar "instant trips" on other NASA satellites had not been reported in the GIDEP system. Consider hardwiring receiver power to the power bus, or build redundancy into the power switching or into the operational status sensing. Ensure that GIDEP reports or NASA Alerts are written and routed to mission operations (as well as to hardware developers), and that flight software responds to command loss with a set of timed spacecraft-level fault responses." 568

- **Probable Scenario for Mars Polar Lander Mission Loss (1998) (Affects of an incomplete software requirements specification). Lesson Number 0938:** "All known hardware operational characteristics, including transients and spurious signals, must be reflected in the software requirements documents and verified by test." 529

- **Consider Language Differences When Conveying Requirements to Foreign Partners (1997) (Diagrams may be useful in requirements specifications). Lesson Number 0608:** "It is especially important when working with foreign partners to document requirements in terms that describe the intent very clearly; include graphics where possible." 511

- **Develop and Test the Launch Procedure Early (1997). Lesson Number 0609:** The Abstract states: "During the terminal countdown for the first attempted launch of Cassini, spacecraft telemetry channels indicated a false alarm condition that delayed verification of spacecraft readiness for launch, and contributed to a delay in the launch day. The anomaly was traced to erroneous telemetry documentation. Develop and release the launch procedure early enough for comprehensive testing before launch. Rigorously test and verify all telemetry channels and their alarms and ensure documentation such as telemetry definitions is kept up to date." 565

- **Mars Climate Orbiter Mishap Investigation Board - Phase I Report. Lesson Number 0641:** "The MCO Mishap Investigation Board (MIB)... determined that the root cause for the loss of the MCO spacecraft was the failure to use metric units in the coding of a ground software file." The
data in the ... file was required to be in metric units per existing software interface documentation, and the trajectory modelers assumed the data was provided in metric units per the requirements. In fact, the angular momentum (impulse) data was in English units rather than metric units. The failure to properly communicate what was in the interface documentation is a warning that the effective implementation of the IDD requirement includes adequate communication of its contents, not just the writing and recording.

- **Interface Control and Verification. Lesson Number 0569:** Problems occurred during the Mars Pathfinder spacecraft integration and test due to out-of-date or incomplete interface documentation. (While this lesson involves a hardware-related problem, it illustrates the need for accuracy in interface documentation.) Investigation showed that the main wiring harness was built in accordance with documentation that had not been updated after design changes were made. In part, this was due to independently prepared Mechanical Interface Control Drawings by the Government and the contractor. The MICDs should have had periodic verification for accuracy and compatibility.

- **Problem Reporting and Corrective Action System. Lesson Number 0738:** Impact of Non-Practice states: “Hardware/software problems that require further investigation may not be identified and tracked. Development of corrective action and need for improvement will not be highlighted to engineering. Opportunities for early elimination of the causes of failures and valuable trending data can be overlooked.” Practice states: “A closed-loop Problem (or Failure) Reporting and Corrective Action System (PRACAS or FRACAS) is implemented to obtain feedback about the operation of ground support equipment used for the manned spaceflight program.”

- **Pre-Flight Problem/Failure Reporting Procedures. Lesson Number 0733:** Impact of Non-Practice states: “Without ... formal reporting procedures, problems/failures, particularly minor glitches, may be overlooked or not considered serious enough to investigate or report to Project Management. This could result in recurrence of the problem/failure during the mission and result in significant degradation in performance.”

- **Probable Scenario for Mars Polar Lander Mission Loss (1998) (Importance of including known hardware characteristics). Lesson Number 0938:** “1. Project test policy and procedures should specify actions to be taken when a failure occurs during test. When tests are aborted, or known to have had flawed procedures, they must be rerun after the test deficiencies are corrected. When test article hardware or software is changed, the test should be rerun unless there is a clear rationale for omitting the rerun. 2. All known hardware operational characteristics, including transients and spurious signals, must be reflected in the software requirements documents and verified by test.”

- **Take CM Measures to Control the Renaming and Reuse of Old Command Files. Lesson Number 1481:** The Mars Odyssey mission ran into a version control issue when they discovered an improperly named file call script. It was determined that the team had taken an old Mars Global surveyor file to reuse. The file was renamed, but its code creation time captured in the header was not changed. This caused the system to label the file as an old file. As a result, the operations team had to manually specify the correct file to use, until subsequent code fixes were implemented.

- **Computer Hardware-Software/Software Development Tools/Maintenance. Lesson Number 1128:** “NASA concurs with the finding that no program-wide plan exists addressing the maintenance of COTS software development tools. A programmatic action has been assigned to develop the usage requirements for COTS/modified off-the-shelf software including the associated development tools. These guidelines will document maintenance and selection guidelines to be used by all of the applicable program elements.”

- **International Space Station (ISS) Program/Computer Hardware-Software/International Partner Source Code (Maintenance Agreements.) Lesson Number 1153:** The Recommendation states to “Solidify long-term source code maintenance and incident investigation agreements for all software being developed by the International Partners as quickly as possible, and develop contingency plans for all operations that cannot be adequately placed under NASA’s control.”