MathWorks®

Accelerating the pace of engineering and science

NASA NPR 7150.2 Compliant Flight Software Development Workflow

REQUIREMENTS

Req. Baseline

NPR 7150.2 – 4.1 Software Requirements
4.1.1 The requirements phase is one of the most important phases of software engineering. Studies show that the top problems in the software industry are due to poor requirements elicitation, inadequate requirements specification, and inadequate management of changes to requirements. Requirements provide the foundation for the entire life-cycle, as well as for the software product. Requirements also provide a basis for planning, estimating, and monitoring. Requirements are based on customer, user, and other stakeholder needs and design and development constraints. The development of requirements includes elicitation, analysis, documentation, verification, and validation. Ongoing customer validation of the requirements to ensure the end products meet customer needs is an important part of the life-cycle process. This can be accomplished via rapid prototyping and customer-involved reviews of iterative and final software requirements.

Design, Source Code and Test Case Traceability

NPR 7150.2 – 4.5 Software Testing
4.5.4 The project manager shall verify the requirement to the implementation of each software requirement. [SWE-067]
4.5.8 The project manager shall update software test plan(s) and software test procedure(s) to be consistent with software requirements. [SWE-071]
4.5.9 The project manager shall provide and maintain bidirectional traceability from the software test procedures to the software requirements. [SWE-072]

Simulink Requirements

NPR 7150.2 – 4.1 Software Requirements
4.1.2.1 The project manager shall establish, capture, record, approve, and maintain software requirements, including the software quality requirements, as part of the technical specification. [SWE-050].
4.1.2.3 The project manager shall perform, record, and maintain bidirectional traceability between the software requirement and the higher-level requirement. [SWE-052]
4.1.3.1 The project manager shall track and manage changes to the software requirements. [SWE-053]

NPR 7150.2 – 4.3 Software Design
4.3.4 The project manager shall perform, record, and maintain bidirectional traceability between the following: [SWE-059]
a. Software requirements and software architecture.
b. Software architecture and software design.
c. Software requirements and software design.

Test Case Traceability

Design constraints (Equivalence classes, Boundary Values, Derived Requirements)

Simulation Cases Results

Coverage Metrics

Simulink Coverage

Simulink Design Verifier

Design Error Detection and Property Proving

Design Traceability

Automatic Test Case Generation

Requirement Based Functional Test Cases + Test Cases identified using Formal Methods

NPR 7150.2 – 4.5 Software Testing
4.5.7 The project manager shall use validated and accredited software models, simulations, and analysis tools required to perform qualification of flight software or flight equipment. [SWE-070]

NPR 7150.2 – 4.3 Software Design
4.3.1 Software design is the process of defining the software architecture, components, modules, interfaces, and data for a software system to satisfy specified requirements. The software architecture is the fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution. The software architectural design is concerned with creating a strong overall structure for software entities that fulfill allocated system and software-level requirements. Typical views captured in an architectural design include the decomposition of the software subsystem into design entities, computer software configuration items, definitions of external and internal interfaces, dependency relationships among entities and system resources, and finite state machines. The design should be further refined into lower-level entities that permit the implementation by coding in a programming language. Typical attributes that are documented for lower-level entities include: identifier, type, purpose, function, constraints, subordinates, dependencies, interface, resources, processing, and data. Rigorous specification languages, graphical representations, and related tools have been developed to support the evaluation of critical properties at the design level. Projects are encouraged to take advantage of these improved design techniques to prevent and eliminate errors as early in the life cycle as possible.

NPR 7150.2 – 4.5 Software Testing
4.5.1 The purpose of testing is to verify the software functionality and remove defects. Testing verifies the code against the requirements and the design to ensure that the requirements are implemented. Testing also identifies problems and defects that are corrected and tracked to closure before product delivery. Testing also validates that the software operates appropriately in the intended environment. Please note for Class A software, additional software test and integration requirements exist in NPR 8705.2 beyond those listed below.

NPR 7150.2 – 4.1 Software Requirements
4.1.3.3 The project manager shall perform requirements validation to ensure that the software will perform as intended in the customer environment. [SWE-055]

Model Coverage Analysis

Testing Environment Settings

Simulink Test

Simulation in the Loop Functional Testing

MODEL

SWRS + SADD

Modelling Standards

Simulink Check

Model Conformance Checks

NPR 7150.2 – 4.4 Software Implementation
4.4.3 The project manager shall select, adhere to, and verify software coding methods, standards, and/or criteria. [SWE-061]

Doc Templates Scripts

Simulink Report Generator

- Software Development Plan (SDP)
  - Modelling and Design Standards
- SW Requirements Specification (SRS)
  - Software Logical Model
  - Req. Traceability Matrix
- SW Design Document (SDD)
  - Software Architectural Design
  - Software Behaviour
  - Internal Interface Design
- Software Verification Report (SVR)
  - Autocode input model review
  - Test Traceability Matrix
  - Model Conformance Report
  - Model Coverage Report

...

NPR 7150.2 – 4.3 Software Design
4.3.2 The project manager shall develop, record, and maintain the software design. [SWE-056]
4.3.3 The project manager shall develop, record, and maintain a design based on the software architectural design that describes the lower-level units so that they can be coded, compiled, and tested. [SWE-058]

Source Code Traceability

NPR 7150.2 – 4.4 Software Implementation
4.4.7 The project manager shall perform, record, and maintain bidirectional traceability from software design to the software code. [SWE-064]

Coder Settings

NPR 7150.2 – 4.4 Software Implementation
4.4.1 Software implementation consists of implementing the requirements and design into code, data, and records. Software implementation also consists of following coding methods and standards. Unit testing is also usually a part of software implementation (unit testing can also be conducted during the testing phase).

Embedded Coder

NPR 7150.2 – 3.8 Automatic Generation of Software Source Code
3.8.1. The project manager shall define the approach to the automatic generation of software source code including: [SWE-146]
a. Validation and verification of auto-generation tools.
b. Configuration management of the auto-generation tools and associated data.
c. Identification of the allowable scope for the use of auto-generated software.
d. Verification and validation of auto-generated source code.
e. Monitoring the actual use of auto-generated source code compared to the planned use.
f. Policies and procedures for making manual changes to auto-generated source code.
g. Configuration management of the input to the auto-generation tool, the output of the auto-generation tool, and modifications made to the output of the auto-generation tools.

NPR 7150.2 – 4.4 Software Implementation
4.4.2 The project manager shall implement the software design into software code. [SWE-060]

NPR 7150.2 – 4.4 Software Implementation
4.4.8 The project manager shall validate and accredit software tool(s) required to develop or maintain software. [SWE-136]

Software Verification Test Case Traceability

Simulink Requirements

Source Code Traceability

Automatic Code Generation

NPR 7150.2 – 3.16 Software Security
3.16.6 The project manager shall ensure that the space flight software systems are assessed for possible security vulnerabilities and weaknesses. [SWE-158]

Simulink Code Inspector

Automatic Code Inspection - structure and traceability

Coding Standards

Polyspace Bug Finder

Code Conformance (MISRA,...)

Validation Objectives Settings

Polyspace Code Prover

Prove Absence of RT Errors

NPR 7150.2 – 5.3 Software Peer Reviews and Inspections
5.3.2 The project manager shall perform and report the results of software peer reviews or software inspections for: [SWE-087]
d. Software code as defined in the software and or project plans.

NPR 7150.2 – 4.4 Software Implementation
4.4.4 The project manager shall verify the software code by using the results from static analysis tool(s). [SWE-135]

Doc Templates Scripts

Simulink Report Generator

- Software Development Plan (SDP)
  - Coding standards
  - Test Traceability Matrix
  - Tool Validation Documentation
- Software Unit Integration Test Plan (SUITP)
- Software Validation Report (SVR)
  - Code Generation Report
  - Software Unit Test Report
  - Software Code Traceability Matrix
  - Code Coverage Report
  - Robustness Report
  - Independent SW Validation Report
  - Review and Inspection Reports

...

Software in the Loop (SIL) Unit Testing

SOURCE CODE

NPR 7150.2 – 4.4 Software Implementation
4.4.5 The project manager shall unit test the software code per the plans for software testing. [SWE-062]

NPR 7150.2 – 4.4 Software Implementation
4.4.6 The project manager shall provide a software version description for each software release. [SWE-063]

Testing Environment Settings

Code Coverage

Reuse of Previous Test Cases
Addition of Real-Time specific Test Cases

Simulink Test

SIL Test Cases Results

Simulink Coverage

Compiler

NPR 7150.2 – 4.5 Software Testing
4.5.2 The project manager shall establish and maintain: [SWE-065]
a. Software test plan(s).
b. Software test procedure(s).
c. Software test report(s).
4.5.5 The project manager shall evaluate test results and record the evaluation. [SWE-068]

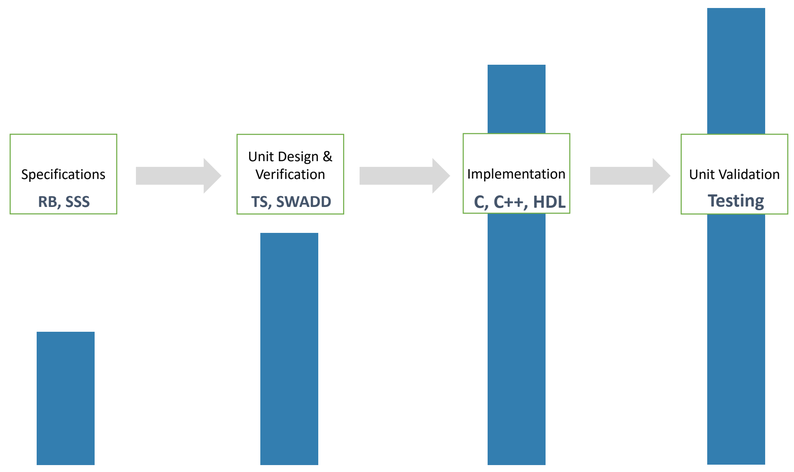
NPR 7150.2 – 4.5 Software Operations, Maintenance, and Retirement
4.6.3 The project manager shall complete and deliver the software product to the customer with appropriate records, including as-built records, to support the operations and maintenance phase of the software's life cycle. [SWE-077]

NPR 7150.2 – 4.5 Software Testing
4.5.10 The project manager shall validate the software system on the targeted platform or high-fidelity simulation. [SWE-073]

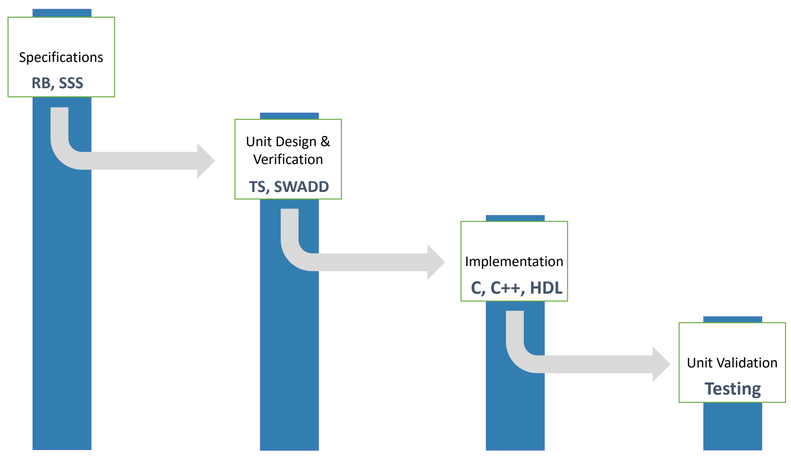
EXECUTABLE OBJECT CODE

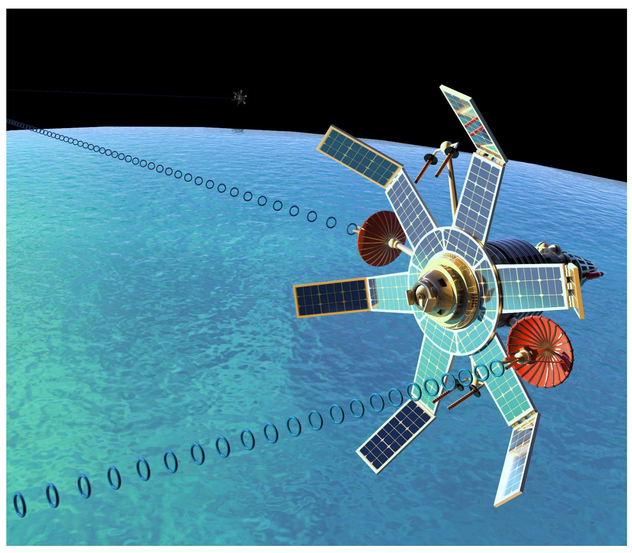
Processor and Hardware in the Loop (PIL and HIL) Unit Testing

Effort Distribution in Traditional Development Workflows



Effort Distribution in Model-Based Design Workflows





MATLAB® & SIMULINK®

Prepared by: Ossi Saarela, MathWorks
Ossi.Saarela@MathWorks.com
+1-508-647-1618
March 2018