NASA Procedural Requirements

COMPLIANCE IS MANDATORY

NPR 7150.2A
Effective Date:
November 19, 2009
Expiration Date:
November 19, 2014

Printable Format (PDF)

Request Notification of Change (NASA Only)

Subject: NASA Software Engineering Requirements

Responsible Office: Office of the Chief Engineer

| TOC | Preface | Chapter1 | Chapter2 | Chapter3 | Chapter4 | Chapter5 | Chapter6 | AppendixA | AppendixB | AppendixC | AppendixD | AppendixE | ALL |

Appendix E. Software Classifications

The applicability of requirements in this NPR to specific systems and subsystems containing software is determined through the use of the NASA-wide definitions for software classes defined below and the designation of the software as safety-critical or non-safety-critical in conjunction with the Requirements Mapping Matrix in Appendix D. These definitions are based on 1) usage of the software with or within a NASA system, 2) criticality of the system to NASA's major programs and projects, 3) extent to which humans depend upon the system, 4) developmental and operational complexity, and 5) extent of the Agency's investment. Classes A – E cover engineering related software in decreasing order of applicable NPR 7150.2 requirements. Classes F through H cover business and IT software in decreasing order of applicable NPR 7120.2 requirements. Using the Requirements Mapping Matrix, the number of applicable requirements and their associated rigor are scaled back for lower software classes and software designated as non-safety-critical. Situations where a project contains separate systems and subsystems having different software classes are not uncommon.

For a given system or subsystem, software is expected to be uniquely defined within a single class. If more than one software class seems to apply, then assign the higher of the classes to the system/subsystem. Any potential discrepancies in classifying software within Classes A - E are to be resolved using the definitions and the five underlying factors listed in the previous paragraph. Engineering and Safety and Mission Assurance provide dual Technical Authority chains for resolving classification issues. The NASA Headquarters' Chief Engineer is the ultimate Technical Authority for software classification disputes concerning definitions in this NPR.

Class A: Human Rated Space Software Systems

Definition:

Human Space Flight Software Systems*: (ground and flight) developed and/or operated by or for NASA that are needed to perform a primary mission objective of human space flight and directly interacts with human space flight systems. Limited to software required to perform "vehicle, crew, or primary mission function," as defined by software that is:

1. Required to operate the vehicle or space asset (e.g., spacesuit, rover, or outpost), including commanding of the vehicle or asset, or

2. required to sustain a safe, habitable environment for the crew, or

3. required to achieve the primary mission objectives, or

4. directly prepares resources (e.g., data, fuel, power) that are consumed by the above functions.

* Includes software involving launch, onorbit, in space, surface operations, and entry, descent, and landing.

---

1 Current standards that address habitability and environmental health, including atmospheric composition and pressure, air and water quality and monitoring, acceleration, acoustics, vibration, radiation, thermal environment, combined environmental effects, and human factors, are documented in NASA-STD-3000, Volumes I-II, Man-Systems Integration Standards.

Examples:

Examples of Class A software (human rated space flight) include but are not limited to:

During Launch: abort modes and selection; separation control; range safety; crew interface (display and controls); crew escape; critical systems monitoring and control; guidance, navigation, and control; and communication and tracking.

On Orbit/In Space: Extra Vehicular Activity (EVA); control of electrical power; payload control (including suppression of hazardous satellite and device commands); critical systems monitoring and control; guidance, navigation, and control; life support systems; crew escape; rendezvous and docking; failure detection; isolation and recovery; communication and tracking; and mission operations.

On Ground: pre-launch and launch operations; Mission Control Center (and Launch Control Center) front end processors; spacecraft commanding; vehicle processing operations; and re-entry operations; flight dynamics simulators used for ascent abort calls; and launch and flight controller stations for manned spaceflight.

Entry, Descent and Landing (EDL): command and control; aero-surface control; power; thermal; and fault protection; and communication and tracking.

Surface Operations: planet/lunar surface EVA; and communication and tracking.

Exclusions:

Class A does not include:

1. Software which happens to fly in space but is superfluous to mission objectives (e.g., software contained in an iPod carried on board by an astronaut for personal use), or

2. software that exclusively supports aeronautics, Research and Technology, and science conducted without spaceflight applications, or

3. systems (e.g., simulators, emulators, stimulators, facilities) used to test Class A systems containing software in a development environment.

Class B: Non-Human Space Rated Software Systems or Large Scale Aeronautics Vehicles

Definition:

Space Systems*: Flight and ground software that must perform reliably to accomplish primary mission objectives, or major function(s) in Non-Human Space Rated Systems.

Limited to software that is:

1. Required to operate the vehicle or space asset (e.g., orbiter, lander, probe, flyby spacecraft, rover, launch vehicle, or primary instrument), such as commanding of the vehicle or asset, or

2. required to achieve the primary mission objectives, or

3. directly prepares resources (data, fuel, power, etc.) that are consumed by the above functions.

Airborne Vehicles: Large Scale1 aeronautic vehicles that are NASA unique in which the software:

1. Is integral to the control of an airborne vehicle, or

2. monitors and controls the cabin environment, or

3. monitors and controls the vehicle's emergency systems.

This definition includes software for vehicles classified as "test", "experimental", or "demonstration" which meets the above definition for Class B software. Also included are systems in a test or demonstration where the software's known and scheduled intended use is to be part of a Class A or B software system.

* Includes software involving launch, onorbit, in space, surface operations, and entry, descent, and landing.

---

1 Large-scale (life-cycle cost exceeding $250M) fully integrated technology development system — see NPR 7120.8, section 5.1.1.1.

Examples:

Examples of Class B software includes but are not limited to:

Space, Launch, Ground, EDL, and Surface Systems: propulsion systems; power systems; guidance navigation and control; fault protection; thermal systems; command and control ground systems; planetary/lunar surface operations; hazard prevention; primary instruments; science sequencing engine; simulations which create operational EDL parameters; subsystems that could cause the loss of science return from multiple instruments; flight dynamics and related data; launch and flight controller stations for non-human spaceflight.

Aeronautics Vehicles (Large Scale NASA Unique): guidance, navigation, and control; flight management systems; autopilot; propulsion systems; power systems; emergency systems (e.g., fire suppression systems, emergency egress systems; emergency oxygen supply systems, traffic/ground collision avoidance system); and cabin pressure and temperature control.

Exclusions:

Class B does not include:

1. Software that exclusively supports non-primary instruments on Non-Human Space Rated Systems (e.g., low cost non-primary university supplied instruments) or

2. systems (e.g., simulators emulators, stimulators, facilities) used in testing Class B systems containing software in a development environment.

Class C: Mission Support Software or Aeronautic Vehicles, or Major Engineering/Research Facility Software

Definition:

Space Systems:

1. Flight or ground software that is necessary for the science return from a single (non-primary) instrument, or

2. flight or ground software that is used to analyze or process mission data, or

3. other software for which a defect could adversely impact attainment of some secondary mission objectives or cause operational problems, or

4. software used for the testing of space assets, or

5. software used to verify system requirements of space assets by analysis, or

6. software for space flight operations, that is not covered by Class A or B.

Airborne Vehicles:

Systems for non-large scale aeronautic vehicles in which the software:

1. is integral to the control of an airborne vehicle, or

2. monitors and controls the cabin environment, or

3. monitors and controls the vehicle's emergency system.

Systems on an airborne vehicle (including large scale vehicles) that acquire, store, or transmit the official record copy of flight or test data.

Major Engineering/Research Facility: Systems that operate a major facility for research, development, test, or evaluation (e.g., facility controls and monitoring, systems that operate facility-owned instruments, apparatus, and data acquisition equipment).

Examples: Examples of Class C software include but are not limited to:

Space Systems: software that supports prelaunch integration and test; mission data processing and analysis; analysis software used in trend analysis and calibration of flight engineering parameters; primary/major science data collection storage and distribution systems (e.g., Distributed Active Archive Centers); simulators, emulators, stimulators, or facilities used to test Class A, B, or C software in a development, integration and test environments (development environment includes environments used from unit testing through validation testing); software used to verify system-level requirements associated with Class A, B, or C software by analysis (e.g., guidance, navigation and controls (GN&C) system performance verification by analysis); simulators used for mission training; software employed by network operations and control (which is redundant with systems used at tracking complexes); command and control of non-primary instruments; ground mission support software used for secondary mission objectives, real-time analysis, and planning (e.g., monitoring, consumables analysis, mission planning).

Aeronautics Vehicles: guidance, navigation, and control; flight management systems; autopilot; propulsion systems; power systems; emergency systems (e.g., fire suppression systems, emergency egress systems, emergency oxygen supply systems, traffic/ground collision avoidance system); cabin pressure and temperature control; in-flight telescope control software; aviation data integration systems; and automated flight planning systems.

Major Engineering/Research Facility: major Center facilities; data acquisition and control systems for wind tunnels, vacuum chambers, and rocket engine test stands; ground-based software used to operate a major facility telescope; and major aeronautic applications facilities (e.g., air traffic management systems; high fidelity motion based simulators).

Exclusions: Systems unique to a research, development, test, or evaluation activity in a Major Engineering/Research Facility or Airborne Vehicle where the system is not part of the facility or vehicle and does not impact the operation of the facility or vehicle.

Class D: Basic Science/Engineering Design and Research and Technology Software

Definition:

Basic Science/Engineering Design:

1. Ground software that performs secondary science data analysis, or

2. ground software tools that support engineering development, or

3. ground software used in testing other Class D software systems, or

4. ground software tools that support mission planning or formulation, or

5. ground software that operates a research, development, test, or evaluation laboratory (i.e., not a Major Engineering/Research Facility), or

6. ground software that provides decision support for non-mission critical situations.

Airborne Vehicle Systems:

1. Software whose anomalous behavior would cause or contribute to a failure of system function resulting in a minor failure condition for the airborne vehicle (e.g., the Software Considerations in Airborne System and Equipment Certification, DO-178B, "Class D"), or

2. software whose anomalous behavior would cause or contribute to a failure of system function with no effect on airborne vehicle operational capability or pilot workload (e.g., the Software Considerations in Airborne System and Equipment Certification, DO-178B, "Class E"), or

3. ground software tools that perform research associated with airborne vehicles or systems.

Major Engineering/Research Facility Related: research software that executes in a Major Engineering/Research Facility but is independent of the operation of the facility.

Examples:

Examples of Class D software includes but are not limited to:

Basic Science and Engineering Design: engineering design and modeling tools (e.g., Computer-Aided Design and Computer-Aided Manufacturing (CAD/CAM), thermal/structural analysis tools); project assurance databases (e.g., problem reporting, analysis, and corrective action system, requirements management databases); propulsion integrated design tools; integrated build management systems; inventory management tools; probabilistic engineering analysis tools; test stand data analysis tools; test stand engineering support tools; experimental flight displays evaluated in a flight simulator; and tools used to develop design reference missions to support early mission planning.

Airborne Vehicles: software tools for designing advanced human-automation systems; experimental synthetic-vision display; and cloud-aerosol Light Detection and Ranging (LIDAR) installed on an aeronautics vehicle.

Exclusions:

Class D does not include:

1. Software that can impact primary or secondary mission objectives or cause loss of data that is generated by space systems, or

2. software which operates a Major Engineering/Research Facility, or

3. software which operates an airborne vehicle, or

4. space flight software.

Class E: Small Light Weight Design Concept and Research and Technology Software

Definition:

1. Software developed to explore a design concept or hypothesis, but not used to make decisions for an operational Class A, B, or C system or to-be built Class A, B, or C system, or

2. software used to perform minor desktop analysis of science or experimental data.

Examples: Examples of Class E software include but are not limited to:

parametric models to estimate performance or other attributes of design concepts; software to explore correlations between data sets; line of code counters; file format converters; and document template builders.

Exclusions:

Class E does not include:

1. Space flight systems, or

2. software developed by or for NASA to directly support an operational system (e.g., human rated space system, robotics spacecraft, space instrument, airborne vehicle, major engineering/research facility, mission support facility, primary/major science data collection storage and distribution systems), or

3. software developed by or for NASA to be flight qualified to support an operational system, or

4. software that directly affects primary or secondary mission objectives, or

5. software that can adversely affect the integrity of engineering/scientific artifacts, or

6. software used in technical decisions concerning operational systems, or

7. software that has an impact on operational vehicles.

Business and IT Infrastructure Software

Class F: General Purpose Computing Software (Multi-Center or Multi- Program/Project)

Definition: General purpose computing software used in support of the Agency, multiple Centers, or multiple programs/projects, as described for the General Purpose Infrastructure To-Be Component of the NASA Enterprise Architecture, Volume 5 (To-Be Architecture), and for the following portfolios: voice, wide area network, local area network, video, data Centers, application services, messaging and collaboration, and public Web. A defect in Class F software is likely to affect the productivity of multiple users across several geographic locations and may possibly affect mission objectives or system safety. Mission objectives can be cost, schedule, or technical objectives for any work that the Agency performs.

Examples: Examples of Class F software include but are not limited to: software in support of the NASA-wide area network; the NASA Web portal; and applications supporting the Agency's Integrated Enterprise Management Program, such as the time and attendance system, Travel Manager, Business Warehouse, and E-Payroll.

Class G: General Purpose Computing Software (Single Center or Project)

Definition: General purpose computing software used in support of a single Center or

project, as described for locally deployed portions of the General Purpose Infrastructure To-Be Component of the NASA Enterprise Architecture, Volume 5 (To-Be Architecture) and for the following portfolios: voice, local area network, video, data Centers, application services, messaging and collaboration, and public Web. A defect in Class G software is likely to affect the productivity of multiple users in a single geographic location or workgroup but is unlikely to affect mission objectives or system safety.

Examples: Examples of Class G software include but are not limited to: software for Center custom applications such as Headquarters' Corrective Action Tracking System and Headquarters' User Request Systems.

Class H: General Purpose Desktop Software

Definition: General purpose desktop software as described for the General Purpose Infrastructure To-Be Component (Desktop Hardware and Software Portfolio) of the NASA Enterprise Architecture, Volume 5 (NASA To-Be Architecture). A defect in Class H software may affect the productivity of a single user or small group of users but generally will not affect mission objectives or system safety. However, a defect in desktop IT security-related software, e.g., anti-virus software, may lead to loss of functionality and productivity across multiple users and systems.

Examples: Examples of Class H software include but are not limited to: desktop applications such as word processing applications, spreadsheet applications, and presentation applications.

| TOC | Preface | Chapter1 | Chapter2 | Chapter3 | Chapter4 | Chapter5 | Chapter6 | AppendixA | AppendixB | AppendixC | AppendixD | AppendixE | ALL |

| NODIS Library | Program Formulation(7000s) | Search |

DISTRIBUTION:
NODIS

This Document Is Uncontrolled When Printed.
Check the NASA Online Directives Information System (NODIS) Library
to Verify that this is the correct version before use: http://nodis3.gsfc.nasa.gov