



# NASA Procedural Requirements

**NPR 7150.2**

Effective Date: September 27,  
2004

Expiration Date: September  
27, 2009

**COMPLIANCE IS MANDATORY**

[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](#) | [ALL](#) |

## APPENDIX B: Definitions

<p>Class A Human Rated Software Systems</p>	<p>Applies to all space flight software subsystems (ground and flight) developed and/or operated by or for NASA to support human activity in space and that interact with NASA human space flight systems. Space flight system design and associated risks to humans are evaluated over the program's life cycle, including design, development, fabrication, processing, maintenance, launch, recovery, and final disposal. Examples of Class A software for human rated space flight include but are not limited to: guidance; navigation and control; life support systems; crew escape; automated rendezvous and docking; failure detection, isolation and recovery; and mission operations.</p>
<p>Class B Non-Human Space Rated Software Systems</p>	<p>Flight and ground software that must perform reliably in order to accomplish primary mission objectives. Examples of Class B software for non-human (robotic) spaceflight include, but are not limited to, propulsion systems; power systems; guidance navigation and control; fault protection; thermal systems; command and control ground systems; planetary surface operations; hazard prevention; primary instruments; or other subsystems that could cause the loss of science return from multiple instruments.</p>

<p>Class C Mission Support Software</p>	<p>Flight or ground software that is necessary for the science return from a single (non-critical) instrument or is used to analyze or process mission data or other software for which a defect could adversely impact attainment of some secondary mission objectives or cause operational problems for which potential work-arounds exist. Examples of Class C software include, but are not limited to, 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 and distribution systems, major Center facilities, data acquisition and control systems, aeronautic applications, or software employed by network operations and control (which is redundant with systems used at tracking complexes). Class C software must be developed carefully, but validation and verification effort is generally less intensive than for Class B.</p>
<p>Class D Analysis and Distribution Software</p>	<p>Non-space flight software. Software developed to perform science data collection, storage, and distribution; or perform engineering and hardware data analysis. A defect in Class D software may cause rework but has no direct impact on mission objectives or system safety. Examples of Class D software include, but are not limited to, software tools; analysis tools, and science data collection and distribution systems.</p>
<p>Class E Development Support Software</p>	<p>Non-space flight software. Software developed to explore a design concept; or support software or hardware development functions such as requirements management, design, test and integration, configuration management, documentation, or perform science analysis. A defect in Class E software may cause rework but has no direct impact on mission objectives or system safety. Examples of Class E software include, but are not limited to, earth science modeling, information only websites (non-business/information technology); science data analysis; and low technical readiness level research software.</p>

<p>Class F General Purpose Computing Software (Multi-Center or Multi-Program/Project)</p>	<p>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 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 Financial Management Program, such as the time and attendance system, Travel Manager, Business Warehouse, and E-Payroll.</p>
<p>Class G General Purpose Computing Software (Single Center or Project)</p>	<p>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 of Class G software include, but are not limited to, software for Center custom applications such as Headquarters' Corrective Action Tracking System and Headquarters' ODIN New User Request System.</p>
<p>Class H: General Purpose Desktop Software</p>	<p>General purpose desktop software as described for the General Purpose Infrastructure To-Be Component (Desktop Hardware &amp; Software Portfolio) of the NASA Enterprise Architecture, Volume 5 (NASA To-Be Architecture). This class includes software for Wintel, Mac, and Unix desktops as well as laptops. 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</p>

	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 of Class H software include, but are not limited to, desktop applications such as Microsoft Word, Excel, and Power Point, and Adobe Acrobat.
Contracted Software	Software created for a project by a contractor or subcontractor. Process requirements and safety analyses may be included. This is custom-made software, but not in-house.
Commercial-Off-The-Shelf (COTS) Software	Operating systems, libraries, applications, and other software purchased from a commercial vendor. Not customized for a particular project. Access to source code and documentation are often limited.
Glueware	Software created to connect the off-the-shelf software/reused software with the rest of the system. It may take the form of "adapters" that modify interfaces or add missing functionality, "firewalls" that isolate the off-the-shelf software, or "wrappers" that check inputs and outputs to the off-the-shelf software and may modify either to prevent failures.
Government Off-The-Shelf (GOTS) Software	This refers to government-created software, usually from another project. The software was not created by the current developers (see software reuse). Usually, source code is included and all documentation, including test and analysis results, is available. That is, the government is responsible for the GOTS software to be incorporated into another system. (Definition from source document: NASA-GB-8719.13, NASA Software Safety Guidebook.)
Heritage	See legacy. See software reuse.
Insight	Surveillance mode requiring the monitoring of customer-identified metrics and contracted milestones. Insight is a continuum that can range from low intensity, such as reviewing quarterly reports, to high intensity, such as performing surveys and reviews. (Definitions from source document: NPR 8735.2, Management of Government Safety and Mission Assurance Surveillance Functions for NASA Contracts.)

Legacy	<p>These are usually software products (architecture, code, requirements) written specifically for one project and then, without prior planning during its initial development, found to be useful on other projects.</p> <p>See software reuse.</p>
Mission Critical	<p>Item or function that must retain its operational capability to assure mission success. (Definition from source document: NPR 8715.3, NASA Safety Manual.)</p>
Modified Off-The-Shelf (MOTS) Software	<p>When COTS, legacy, reuse, or heritage software is changed to a certain degree, usually more than 10%, then it is considered "modified." The changes can include all or part of the software products and may involve additions, deletions, and specific alterations. An argument can be made that any alterations to the code and/or design of an off-the-shelf software component constitutes "modification;" however, the common usage allows for some percentage of change before the off-the-shelf software is declared to be MOTS software. This may include the changes to the application shell and/or glueware to add or protect against certain features and not to the off-the-shelf software system code directly.</p> <p>See off-the-shelf.</p>
Off-The-Shelf Software	<p>Software not developed in-house or by a contractor for the specific project now underway. The software is general purpose or developed for a different purpose from the current project.</p>
Oversight	<p>Surveillance mode that is in line with the supplier's processes. The customer retains and exercises the right to concur or nonconcur with the supplier's decisions. Nonconcurrency must be resolved before the supplier can proceed. Oversight is a continuum that can range from low intensity, such as customer concurrence in reviews (e.g., PDR, CDR), to high intensity oversight, in which the customer has day-to-day involvement in the supplier's decision-making process (e.g., hardware inspections). (Definition from source document: NPR 8735.2, Management of Government Safety and Mission Assurance Surveillance Functions for NASA Contracts.)</p>

Process Asset Library	A collection of process asset holdings that can be used by an organization or project. (Definition from source document: CMMI® for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing.)
Program	The term "program" is as defined in NPR 7120.5, NASA Program and Project Management Processes and Requirements.
Project	The term "project" is as defined in NPR 7120.5, NASA Program and Project Management Processes and Requirements.
Reuse	See software reuse.
Risk Management	An organized, systematic decision-making process that efficiently identifies, analyzes, plans, tracks, controls, communicates, and documents risk to increase the likelihood of achieving program/project goals. (Definition from source document: NPR 8735.2, Management of Government Safety and Mission Assurance Surveillance Functions for NASA Contracts.)
Safety Critical Function	The term "safety critical function" is as defined in NPR 8715.3, NASA Safety Manual.
Safety Critical	The term "safety critical" is as defined in NPR 8715.3, NASA Safety Manual.
Software	Computer programs, procedures, rules, and associated documentation and data pertaining to the development and operation of a computer system. Software includes programs and operational data . This also includes COTS, GOTS, MOTS, reuse, auto code generated, firmware, and open source software components.
Software Engineering	The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software: that is, the application of engineering to software (Definition from source document: IEEE 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.)

Software Reuse	A software product developed for one use but having other uses or one developed specifically to be usable on multiple projects or in multiple roles on one project. Examples include, but are not limited to, COTS products, acquirer-furnished software products, software products in reuse libraries, and pre-existing developer software products. Each use may include all or part of the software product and may involve its modification. This term can be applied to any software product (such as, requirements and architectures), not just to software code itself. Often this is software previously written by an in-house development team and used on a different project. GOTS software would come under this category if it is supplied from one government project to another government project. (Definition from source document: NASA-GB-8719.13, NASA Software Safety Guidebook.)
Surveillance	The continual monitoring and verification of status of an entity and analysis of records to ensure that specified requirements are being met. Note: Surveillance can be performed in an insight, oversight, or a combined mode as determined by NASA using a risk-based decision process. (Definition from source document: NPR 8735.2, Management of Government Safety and Mission Assurance Surveillance Functions for NASA Contracts.)
System	The combination of elements that function together to produce the capability required to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose. (Definition from source document: NPR 7120.5, NASA Program and Project Management Processes and Requirements.)
Validation	Proof that the product accomplishes the intended purpose. May be determined by a combination of test, analysis, and demonstration. (Definition from source document: NPR 7120.5, NASA Program and Project Management Processes and Requirements.) Note: Software validation also includes software peer review and inspection.

Verification	Proof of compliance with specifications. May be determined by a combination of test, analysis, demonstration, and inspection. (Definitions from source document: NPR 7120.5, NASA Program and Project Management Processes and Requirements.)
Wrappers	See glueware.

| [TOC](#) | [Preface](#) | [Chapter1](#) | [Chapter2](#) | [Chapter3](#) | [Chapter4](#) | [Chapter5](#) | [Chapter6](#) |  
[AppendixA](#) | [AppendixB](#) | [AppendixC](#) | [AppendixD](#) | [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>

---