Downloaded from http://www.everyspec.com

National Aeronautics and
Space Administration

NOT MEASUREMENT SENSITIVE

NASA-STD-8739.8 w/Change 1
July 28, 2004

SOFTWARE ASSURANCE STANDARD

NASA TECHNICAL STANDARD

REPLACES NASA-STD-2201-93 DATED NOVEMBER 1992

Downloaded from http://www.everyspec.com

Record of Changes

Change No.	Date	Title or Brief Description	Entered By
1	5 May 2005	Administrative changes to the Preface; Paragraphs 1.1, 1.4, 1.5, 2.1.1, 2.2.2, 3, 5.1.2.3, 5.4.1.1; 5.6.2, 5.8.1.2, 6.7.1.a, 7.3.2, 7.3.3, 7.5, 7.5.1; Table 1; Appendix A; Appendix C to reflect NASA Transformation changes, reflect release of NPR 7150.2, NASA Software Engineering Requirements and to make minor editorial changes. Note: Some paragraphs have changed pages as a result of these changes. Only pages where content has changed are identified by change indications.	WBHIII

Downloaded from http://www.everyspec.com

PREFACE

Effective Date: July 28, 2004

This document has been issued to make available to software engineers, managers, assurance engineers, and safety practitioners a standard for assessing software systems for software's contribution to safety and quality. It describes the processes and procedures for analyzing and applying appropriate software assurance techniques and methods to software. Software assurance engineers, software managers, and engineers are the primary focus. The audience also includes Safety and Mission Assurance Directors and Program/Project Managers to inform them of the requirements levied on the software assurance process.

This document:

- Provides a software life cycle perspective for the minimum required software assurance procedures that contribute to quality software.
- Provides the acquirer and provider requirements for software assurance and software engineering activities to obtain the most cost effective, best quality, and safest products.
- Provides basic procedures for establishing, operating, and maintaining a software assurance program whether in house or contracted.
- Provides specific requirements for the umbrella process of software assurance and its disciplines of software quality, software reliability, software safety, software verification and validation, and independent verification and validation.

This standard is approved for use by NASA Headquarters and all NASA Centers and is intended to provide a common framework for consistent practices across NASA programs. This document is to be applied to all software developed by, or for, NASA and to the incorporation of open source, commercial off-the-shelf (COTS), Government off-the-shelf (GOTS), or modified off-the-shelf (MOTS) software in a NASA system. This document applies to new contracts and subcontracts for developing software for use in NASA systems and should be referenced therein. Procuring NASA Mission Directorate Programs or Centers need to review this document and, working with your appropriate Safety and Mission Assurance Directorate's software assurance contact, make a conscious, documented decision as to how best to apply this document to current contracts and on-going projects.

Downloaded from http://www.everyspec.com

Questions concerning the application of this publication to specific procurements or requests should be referred to both the NASA Mission Directorate Program or Center and the NASA Headquarters Safety and Mission Assurance Office.

This standard cancels **NASA-STD-2201-93**, Software Assurance Standard, of November 10, 1992.

/s/
Bryan O'Connor
Chief Safety and Mission Assurance Officer

Downloaded from http://www.everyspec.com

CONTENTS

PARAGRAPH		PAGE
PREFACE		iii
CONTENTS		v
1.	SCOPE	1
1.1	Scope	1
1.2	Purpose	1
1.3	Applicability	2
1.4	Tailoring	2
1.5	Organization of the Standard	2
2.	APPLICABLE DOCUMENTS	4
2.1	Applicable Documents	4
2.2	Reference Documents	4
3.	DEFINITIONS AND ACRONYMS	6
3.1	Definitions	6
3.2	Acronyms	9
4.	SOFTWARE ASSURANCE OVERVIEW	11
5.	ACQUIRER SOFTWARE ASSURANCE	14
5.1	Initialization, Pre-Award	14
5.2	Post RFP, Pre-Award	17
5.3	Post-Award, Pre-Implementation	17
5.4	Contract Implementation, Development	18
5.5	Acceptance	19
5.6	Operation	19
5.7	Maintenance	20
5.8	Retirement	20
6.	PROVIDER SOFTWARE ASSURANCE	21
6.1	Software Assurance Program	21
6.2	Software Assurance Management	21
6.3	Software Assurance Plan	22
6.4	Software Assurance Plan Change Procedures	22
6.5	Software Assurance Approval Authority	22
6.6	Software Assurance Records	23
6.7	Software Assurance Status Reporting	23
6.8	Training	23
6.9	Subcontractor Controls	24

Downloaded from http://www.everyspec.com

PARAGRAPH		PAGE
7.	SOFTWARE ASSURANCE DISCIPLINES	25
7.1	Software Quality	25
7.2	Software Safety	27
7.3	Software Reliability	27
7.4	Software V&V	28
7.5	IV&V	28
APPENDIX A:	THE SOFTWARE ASSURANCE CLASSIFICATION ASSESSMENT	30
A-1.	Software Safety Litmus Test	32
A-2.	Determination of Software Class	34
A-3.	Software Classification Scoring Process	37
A-4.	Determination of Software Assurance Level of Effort	44
A-5.	Software Assurance Classification Report Template	46
APPENDIX B:	ACQUIRER SOFTWARE ASSURANCE PLAN TEMPLATE OUTLINE	48
APPENDIX C:	REQUIREMENTS COMPLIANCE MATRIX	51

FIGURES AND TABLES

Figure A-1.	Software Assurance Classification Assessment Process	31
Table 1.	Example of Tailoring for Software Assurance Requirements	16
Table A-1.	Definitions for Hazard Severity	33
Table A-2.	Software Classification Score Sheet	38
Table A-3.	Additional Software Assurance Criteria	45
Table A-4.	Software Assurance Classification Report Template	47

Downloaded from http://www.everyspec.com

1. SCOPE

1.1 Scope

This standard specifies the software assurance requirements for software developed or acquired[1] and maintained by the National Aeronautics and Space Administration (NASA) and for open source software, Government off-the-shelf (GOTS) software, modified off-the-shelf (MOTS) software, and commercial off-the-shelf (COTS) software when included in a NASA system. This Standard applies to use of new and existing (e.g., reuse, legacy, heritage) software products and components.

The NASA Software Assurance Standard (hereinafter referred to as the "Standard") supports NPD 2820.1, NASA Software Policies, and NPR 7150.2 NASA Software Engineering Requirements. This Standard is compatible with all software life cycle models (e.g., waterfall, spiral, evolutionary, incremental, package-based), and addresses all software life cycle processes, including acquisition, supply, development, operation, and maintenance.

This Standard specifies the requirements for software assurance for use by NASA projects, programs, facilities, and activities. It provides a consistent, uniform basis for defining the requirements for software assurance programs to be applied and maintained throughout the life of that software, that is, from project conception, through operations and maintenance, until the software is retired.

In this Standard the words *assure* and *ensure* have the following usages:

- Assure is used when software assurance practitioners make certain that the specified software assurance, management, and engineering activities have been performed by others.

- Ensure is used when software assurance practitioners themselves perform the specified software activities.

1.2 Purpose

The purpose of this Standard is to:

- Establish a common framework, including generic quality procedures, for the software assurance process in support of all life cycle processes, regardless of who performs them.

- Establish and support the cooperation of various groups who are conducting different aspects of the total software assurance process.

- Support and utilize the independent reporting structure required for NASA safety, reliability, and quality processes.

- Define software assurance activities and tasks to meet the objectives of software assurance.

[1] Software acquired from any source; e.g., contractor, university, company.

Downloaded from http://www.everyspec.com

1.3 Applicability

This Standard applies to all software assurance activities during the entire software life cycle of the software developed or acquired for NASA, either internally or externally, including the incorporation of open source, COTS, GOTS, or MOTS into NASA systems. Legacy and reuse software products are also covered with a focus on how they fit into the new systems. The requirements of this Standard are applicable whenever NASA is either the acquirer or provider, and to the extent specified in the contract or other agreement such as Memorandum of Agreement/Understanding.

1.4 Tailoring

Program and project managers, working with software assurance personnel, use the NPR 7150.2, NASA Software Engineering Requirements, and Appendix A, the Software Assurance Classification Assessment, to identify the appropriate software class and the level of software assurance effort to apply. Tailoring the implementation of software assurance requirements is acceptable commensurate with the program/project classification as well as size, complexity, criticality, and risk. Additional tailoring guidance will be provided in the Software Assurance Guidebook. Waivers and/or deviations to the specific requirements in this standard will be via the governing Independent Technical Authority (ITA) processes. A waiver/deviation package will be prepared by a software assurance expert and approved according to NPR 8715.3, NASA Safety Manual.

A compliance matrix listing all of the requirements in this Standard along with the personnel roles and responsibilities required for each requirement is available in Appendix C. This matrix can be used by the program, project, or facility as a checklist to ensure coverage of all requirements in the Standard as tailored.

1.5 Organization of the Standard

Section 2 of this Standard contains the list of documents applicable (directly related) to this Standard as well as the reference documents (information only). Section 3 provides definitions and acronyms used in this Standard. Section 4 provides overview information regarding software assurance and its related disciplines as they apply within NASA and to its contractors. Section 4 does not contain requirements. Sections 5, 6, and 7 contain the requirements of this Standard specified in NPR 7150.2, NASA Software Engineering Requirements. Paragraphs are numbered in these sections to enable recognition of, and to trace conformance to, the requirements. Section 5 addresses the requirements of the acquirer from program/project initiation through retirement of the software. Section 6 addresses the infrastructure and software assurance requirements for the provider. Section 7 identifies requirements specific to software assurance disciplines that may apply to both the acquirer and the provider.

Appendix A provides the Software Assurance Classification Assessment. This should be used in conjunction with the software class criteria in NPR 7150.2 for determining the corresponding software assurance classification and activities. Appendix A identifies the process for performing a

Downloaded from http://www.everyspec.com

software assurance classification assessment and determining if software is safety-critical. The tables in Appendix A are also to be used to help classify and rank software for possible application of Independent Verification and Validation. Appendix B provides a template for a Software Assurance Plan for the acquirer. Appendix C provides a requirements compliance matrix for the Standard.

Downloaded from http://www.everyspec.com

2. APPLICABLE DOCUMENTS

2.1 Applicable Documents

Documents cited in this Standard are listed in this section.

2.1.1 Government documents

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NPD 2810.1	NASA Information Security Policy
NPD 2820.1	NASA Software Policies
NPR 1441.1	NASA Records Retention Schedules
NPR 7120.5	NASA Program and Project Management Processes and Requirements
NPR 7150.2	NASA Software Engineering Requirements
NASA-STD-8719.13	Software Safety Standard

2.1.2 Non-government documents

IEEE 730-2002	IEEE Standard for Software Quality Assurance Plans

2.2 Reference Documents

Reference documents listed in this section are for information only.

2.2.1 Government documents

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NPD 8700.1	NASA Policy for Safety and Mission Success
NPD 8720.1	NASA Reliability and Maintainability (R&M) Program Policy
NPR 2810.1	Security of Information Technology
NPR 8000.4	Risk Management Procedural Requirements
NPR 8715.3	NASA Safety Manual

Downloaded from http://www.everyspec.com

NPR 8735.2	Management of Government Safety and Mission Assurance Surveillance Functions for NASA Contractors
NASA-STD-2202-93	Software Formal Inspections Standard
NASA-GB-8719.13	NASA Software Safety Guidebook
NASA-GB-A201	Software Assurance Guidebook, September 1989
NASA-GB-A301	Software Quality Assurance Audits Guidebook, December 1990
NASA-GB-A302	Software Formal Inspections Guidebook, August 1993

2.2.2 Non-government documents

ISO 9126-1: 2001	Software Engineering Product Quality Part 1: Quality Model
ISO/IEC 12207:1995	Software life cycle processes
ISO 9001: 2000	Quality systems –Model for quality assurance in design, development, production, installation, and servicing
ISO 90003:2000	Quality Management And Quality Assurance Standards - Part 3: Guidelines For The Application Of ANSI/ISO/ASQC 9001:1994 To The Development, Supply, Installation And Maintenance Of Computer Software
IEEE 610.12	IEEE Standard Glossary of Software Engineering Terminology
IEEE 982.1-1988	IEEE Standard Dictionary of Measures to Produce Reliable Software
IEEE 1012-1998	IEEE Standard for Software Verification and Validation
IEEE Std. 1028-1997	IEEE Standard for Software Reviews
SEI-SW-CMM	Software Engineering Institute Capability Maturity Model®
SEI-CMMI	Software Engineering Institute Capability Maturity Model Integration sm
SEI	Continuous Risk Management Guidebook

Downloaded from http://www.everyspec.com

3. DEFNITIONS AND ACRONYMS

The references for the definitions in this Standard are NASA documents and consensus standards. Additional definitions may be found in NPR 7150.2, NASA Software Engineering Requirements.

3.1 Definitions

Term	Definition
Acquirer	The entity or individual who specifies the requirements and accepts the resulting software products. The acquirer is usually NASA or an organization within the Agency but can also refer to the Prime contractor – subcontractor relationship as well.
Assessment	An objective evaluation of performed processes or products and services against their applicable process descriptions, standards, procedures, and requirements.
Audit	An examination of a work product or set of work products performed by a group independent from the developers to assess compliance with specifications, standards, contractual agreements, or other criteria. [Based on IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology]
Configuration Item	An aggregation of hardware, software, or both, that is established and baselined, with any modifications tracked and managed. [Based on IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology] Examples include requirements document, Use Case, or unit of code.
Functional Configuration Audit (FCA)	An audit conducted to verify that the development of a configuration item has been completed satisfactorily, that the item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification, and that its operational and support documents are complete and satisfactory.
Independent Verification and Validation (IV&V)	Verification and validation performed by an organization that is technically, managerially, and financially independent. IV&V, as a part of software assurance, plays a role in the overall NASA software risk mitigation strategy applied throughout the life cycle, to improve the safety and quality of software.
Insight	Surveillance mode requiring the monitoring of acquirer-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.

Downloaded from http://www.everyspec.com

Term	Definition
Oversight	Surveillance mode that is in line with the supplier's processes. The acquirer retains and exercises the right to concur or non-concur with the supplier's decisions. Non-concurrence must be resolved before the supplier can proceed. Oversight is a continuum that can range from low intensity, such as acquirer 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., software inspections).
Peer Review	A review of a software work product, following defined procedures, by peers of the producers of the product for the purpose of identifying defects and improvements. [SEI-CMM Software Engineering Institute Capability Maturity Model®]
Physical Configuration Audit (PCA)	An audit conducted to verify that one or more configuration items, as built, conform to the technical documentation that defines it. [Based on IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology]
Process	A set of interrelated activities, which transform inputs into outputs. [ISO/IEC 12207, Software life cycle processes]
Process Assurance	Activities to assure that all processes involved with the project adhere to plans and comply with the contract and/or any memorandum of agreement/understanding.
Product Assurance	Activities to assure that all required plans are documented, and that the plans, software products, and related documentation adhere to plans and comply with the contract and/or any memorandum of agreement/understanding.
Provider	The entities or individuals that design, develop, implement, test, operate, and maintain the software products. A provider may be a contractor, a university, a separate organization within NASA, or within the same organization as the acquirer. The term “provider” is equivalent to “supplier” in ISO/IEC 12207, Software life cycle processes.
Review	A process or meeting during which a software product or related documentation is presented to project personnel, customers, managers, software assurance personnel, users or user representatives, or other interested parties for comment or approval. [IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology] Reviews include, but are not limited to, requirements review, design review, code review, test readiness review. Other types may include peer review and formal review.
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 contained in hardware (e.g., firmware, programmable logic, and programmable gate arrays). This also includes COTS, GOTS, MOTS, reuse, legacy, and heritage software products and components.

Downloaded from http://www.everyspec.com

Term	Definition
Software Assurance	The planned and systematic set of activities that ensure that software life cycle processes and products conform to requirements, standards, and procedures. [IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology] For NASA this includes the disciplines of Software Quality (functions of Software Quality Engineering, Software Quality Assurance, Software Quality Control), Software Safety, Software Reliability, Software Verification and Validation, and IV&V.
Software Assurance Program Metrics	Metrics related to the activities defined in the Software Assurance Program. Examples include number of reviews/audits planned vs. reviews/audits performed, software assurance effort planned vs. software assurance effort actual, and corrective actions opened vs. corrective actions closed.
Software Assurance Record	A record that provides objective evidence of the extent of the fulfillment of the requirements for software quality, safety, reliability, verification and validation, and, when present, IV&V. This includes documentation of the software assurance activities and analyses results.
Software Life Cycle	The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and sometimes, retirement phase. [IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology]
Software Product Quality	A measure of software that combines the characteristics of low defect rates and high user satisfaction.
Software Quality	The discipline of software quality is a planned and systematic set of activities to ensure quality is built into the software. It consists of software quality assurance, software quality control, and software quality engineering. As an attribute, software quality is (1) the degree to which a system, component, or process meets specified requirements; or (2) the degree to which a system, component, or process meets customer or user needs or expectations. [IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology]
Software Quality Assurance	The function of software quality that assures that the standards, processes, and procedures are appropriate for the project and are correctly implemented.
Software Quality Control	The function of software quality that checks that the project follows its standards, processes, and procedures, and that the project produces the required internal and external (deliverable) products.

Downloaded from http://www.everyspec.com

Term	Definition
Software Quality Engineering	The function of software quality that assures that quality is built into the software by performing analyses, trade studies, and investigations on the requirements, design, code, and verification processes and results to assure that reliability, maintainability, and other quality factors are met.
Software Quality Metrics	Metrics are quantitative values that measure the quality of software or the processes used to develop the software, or some attribute of the software related to the quality (e.g., defect density).
Software Reliability	The discipline of software assurance that (1) defines the requirements for software controlled system fault/failure detection, isolation, and recovery; (2) reviews the software development processes and products for software error prevention and/or reduced functionality states; and (3) defines the process for measuring and analyzing defects and defines/derives the reliability and maintainability factors.
Software Safety	The discipline of software assurance that is a systematic approach to identifying, analyzing, tracking, mitigating, and controlling software hazards and hazardous functions (data and commands) to ensure safe operation within a system.
Surveillance	The continuous monitoring and 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. [NPR 8735.2, Management of Government Safety and Mission Assurance Surveillance Functions for NASA Contractors]
Validation	Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled. [ISO/IEC 12207, Software life cycle processes] In other words, validation ensures that "you built the right thing."
Verification	Confirmation by examination and provision of objective evidence that specified requirements have been fulfilled. [ISO/IEC 12207, Software life cycle processes] In other words, verification ensures that "you built it right."

3.2 Acronyms

CDR	Critical Design Review
CMM®	Capability Maturity Model
CMMI℠	Capability Maturity Model Integration
CMM-SW	Capability Maturity Model - Software
COTS	Commercial off-the-shelf software
GB	Guidebook
GOTS	Government off-the-shelf software
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers, Inc.
ISO	International Organization for Standardization

Downloaded from http://www.everyspec.com

ITA	Independent Technical Authority
IV&V	Independent Verification and Validation
MOA	Memorandum of Agreement
MOU	Memorandum of Understanding
MOTS	Modified off-the-shelf software
NASA	National Aeronautics and Space Administration
NPD	NASA Policy Directive
NPR	NASA Procedural Requirements
OSMA	Office of Safety and Mission Assurance
PDR	Preliminary Design Review
RFP	Request for Proposals
SEI	Software Engineering Institute
SMA	Safety and Mission Assurance
SMO	Systems Management Office
SQA	Software Quality Assurance
STD	Standard
V&V	Verification and Validation

Downloaded from http://www.everyspec.com

4. SOFTWARE ASSURANCE OVERVIEW

This section provides overview information about software assurance and its related disciplines as they apply within NASA and to its contractors. This section does not contain requirements.

NASA performs high-risk functions in the process of achieving its goals and objectives. The Program/Project Manager plans the best risk mitigation strategy for the entire project, of which software is a part. Software assurance is an umbrella risk mitigation strategy for safety and mission assurance of all of NASA's software.

The purpose of software assurance is to assure that software products are of high quality and operate safely. These include products delivered to and used within NASA, and products developed and acquired by NASA. Software assurance assists in risk mitigation by minimizing defects and preventing problems and, through its activities, enables improvement of future products and services. Software assurance is performed by various personnel at each Center in accordance with the organizational structure and governing documents for each program/project. All unresolved software assurance and risk issues are elevated to the level necessary for their resolution. Software assurance is performed by both the acquirer and provider organizations.

The Software assurance process is the planned and systematic set of activities that ensure conformance of software life cycle processes and products to requirements, standards, and procedures. Software assurance assures that the software and its related products meet their specified requirements, conform to standards and regulations, are consistent, complete, correct, safe, secure and reliable as warranted for the system and operating environment, and satisfy customer needs. Software assurance assures that all processes used to acquire, develop, assure, operate and maintain the software are appropriate, sufficient, planned, reviewed, and implemented according to plan, meet any required standards, regulations, and quality requirements. Software assurance utilizes relevant project-based measurement data to monitor each product and process for possible improvements.

Many different groups may perform different aspects of software assurance (e.g., systems engineering might perform the software safety analyses, software engineering might collect and trend defects). An entity/organization independent from the organization creating the software must either perform or assure that software assurance activities are performed correctly and to the necessary level, and that records of those activities are created, analyzed, and maintained. Many software assurance activities may be tailored and performed within the project structure, but a group independent from the project assures those activities and the results. For NASA this is the Safety and Mission Assurance (SMA) organization; for a contractor, this should be a managerially separate safety and assurance organization which should be called out in the contract. Often, one or more software assurance engineers from an SMA organization may be assigned to work with a project throughout its life cycle. While these software assurance engineers are a part of the project and participate in day-to-day activities, perform most or all of the assurance functions, and attend project meetings and reviews, they maintain a separate reporting chain through their SMA organization. This activity is much like an oversight role, that is, the software assurance engineers are closely tied in with the project and provide input on a daily basis. At other times, the independent organization, SMA, may provide only insight for the project, assuring the software

Downloaded from http://www.everyspec.com

assurance activities are performed by the project personnel and participating more by audits and at formal review intervals. In either case, there must be a close working association and joint reporting to both the project and the SMA organization.

Software assurance consists of the following disciplines:

- Software Quality
  - Software Quality Assurance
  - Software Quality Control
  - Software Quality Engineering
- Software Safety
- Software Reliability
- Software Verification and Validation (V&V)
- Independent Verification and Validation (IV&V).

Each assurance discipline brings its own perspective to the tasks; the collective effect of all these efforts provides assurance of mission safety, reliability, and quality.

Software quality consists of a planned and systematic set of activities to assure quality is built into the software. The activities include the functions of software quality assurance, software quality control, and software quality engineering, which comprise product and process assurance. These activities check that the standards, processes, and procedures are appropriate for the project; quality attributes (e.g., reliability, maintainability, testability) are built into the software; and the project correctly implements its standards, processes, and procedures.

Software safety has grown in importance as more and more NASA systems have critical functions either controlled by software or adversely impacted by a software failure. Safety and mission success can be compromised when the software fails. Software safety in conjunction with system safety works to provide a systematic approach to identifying, analyzing, tracking, mitigating and controlling software hazards and hazardous functions (data and commands) to ensure safer software operation within a system. It ensures that safety issues related to software are addressed in reviews and that specific safety analyses and tests are performed especially when there are specific software safety issues and potential hazards. Software safety assures that requirements pertaining to software's control and monitoring of the safety of the system, personnel, environment, and any safing of the system are identified and traced throughout the life cycle of the software. NASA-STD-8719.13, NASA Software Safety Standard, provides details for implementing software safety in NASA programs.

Software reliability is concerned with incorporating and measuring reliability in the products produced by each process of the life cycle. Software reliability optimizes the software through emphasis on requiring and building in software error prevention, fault detection, isolation, recovery, and/or reduced functionality states. Software reliability ensures that systems are fault tolerant when software does fail. It also includes a process for measuring and analyzing defects in the software products during development activities in order to find and address possible problem areas within the software. Measures, including those for software reliability modeling, may be found in the IEEE 982.1, IEEE Standard Dictionary of Measures to Produce Reliable Software.

Downloaded from http://www.everyspec.com

Software V&V is concerned with ensuring that software being developed or maintained satisfies functional and other requirements and that each phase of the development process yields the right products. The V&V process may include rigorous analyses and other techniques to evaluate a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. The V&V process may also apply rigorous analyses and other techniques to evaluate a system or component during or at the end of the development process to confirm that it will fulfill its intended use.

IV&V is performed by an organization that is technically, managerially, and financially independent of the development organization. For NASA, IV&V is performed and/or managed by the NASA IV&V Facility. IV&V, as a part of software assurance, plays a role in the overall NASA software risk mitigation strategy applied throughout the life cycle, to improve the safety and quality of software. IV&V, focusing on mission critical software, provides additional reviews, analyses, and in-depth evaluations of life cycle products that have the highest risk. When applied to a particular software system, IV&V works independently from the program/project, but works in coordination with the other software assurance disciplines.

Downloaded from http://www.everyspec.com

5. ACQUIRER SOFTWARE ASSURANCE

This section addresses the requirements of the acquirer from program/project initiation through retirement of the software.

5.1 Initiation, Pre-Award

The first step once a project, program or facility is conceived and initially approved is to perform an evaluation of the intended software portion of the system(s). Once the NASA project/program/facility office informs the software assurance manager of any intended systems with software, it is evaluated using the criteria in Appendix A to (1) determine the classification of the software, (2) determine the safety criticality, (3) to help determine if it will be considered for IV&V, and (4) further determine the prioritization and level of software assurance effort. This is an initial classification and ranking of the software and needs to be updated as the contract, design, and delivery of the software progresses. The results of the evaluation/assessment of the potential software for a project are coordinated with project management, recorded, maintained, and reported to the SMA Directors and Systems Management Offices (SMO).

NOTE: It is understood that software assurance and any IV&V portion of the software assurance requirements will be high level at this time. Further iterations may be needed before contract implementation.

During the Initiation phase of the program/project, the acquirer is responsible for developing the Request for Proposal (RFP) or Memorandum of Agreement/Understanding in an in-house project.

5.1.1 The acquirer shall identify a person with responsibility for software assurance, e.g., a software assurance manager.

NOTE: While the person identified with the responsibility for software assurance may have other titles, for this document, that person is referred to as the software assurance manager.

5.1.2 The software assurance manager shall perform the following tasks:

5.1.2.1 Ensure completion of the Software Assurance Classification Assessment in Appendix A, for each project, including software management agreement on the results.

5.1.2.2 Ensure that projects with safety-critical software comply with the requirements in NASA STD-8719.13 and the software assurance requirements and activities for the assessed Class of software.

5.1.2.3 Ensure that Class A and B projects, which require the most software assurance, follow all the requirements of Sections 5, 6, and 7. See Table 1 for requirements and implementation of those requirements by Software Class. While the implementation of requirements for Class B will be tailored to some degree, the actual requirements are not. Class C software may address tailoring the assurance requirements based on what is applicable for the software engineering requirements of NPR 7150.2 and according to any potential risks

Downloaded from http://www.everyspec.com

specific to the planned operational or development environment. Class D software may have the most requirements tailoring, matching the assurance activities to the less formal development activities. An experienced software assurance engineer must work closely with the project to assess the software for the project and tailor the software assurance activities accordingly. (See Table 1)

NOTE: Often Class D assurance activities consist mostly of assuring any contractual agreements meet the needs of the project/program and then performing periodic audits and surveys of the project's work to follow up. The level of software assurance effort applied to any class is commensurate with the risk, criticality, complexity, and needed reliability and quality of a project.

NOTE: If the results of the Software Assurance Classification Assessment (Appendix A) identify the software as Class E (which includes Exploratory software), then the requirements of this Standard are not mandatory.

NOTE: Class F-H software is currently the responsibility of the Chief Information Office, however, for the higher level Information Technology or business class systems, if software assurance is requested, those projects would be assured in accordance with the software engineering requirements in NPR 7120.5 they must meet.

Downloaded from http://www.everyspec.com

Table 1. Example of Tailoring for Software Assurance Requirements

Class	**A**	**B**	**C**	**D**	**E**	**F, G, H**
Effort	Full	Full	Medium	Minimal	N/A at this time	Not covered
Tailoring of Requirements	All software assurance requirements apply with no tailoring	All software assurance requirements apply – some minor tailoring to meet project objectives & mission category	Medium tailoring of software assurance requirements to meet project objectives & mission category	Major tailoring of software assurance requirements to meet project objectives & mission category	Initial Classification survey periodically to assure project remains a Class E software project	N/A unless requested
To what extent Requirements are Met*	All activities to meet these requirements will be performed.	All activities to meet the requirements will be performed, how to meet the requirements may be less rigorous.	Activities to meet the requirements may be tailored, i.e., how to meet the requirements will be less rigorous.	Activities to meet the requirements will be tailored, i.e., how to meet the requirements will be minimal.	Activities will mostly consist of software assurance reports on project classification unless otherwise contracted/agreed	Only as specified in an agreement

* How the requirements will be implemented, level of rigor to which the requirements are met.

5.1.2.4 Assure all classifications of software are compared and agreed upon with the project. As some projects may have multiple software tasks, each may need to be assessed separately. The assurance and engineering ITAs will need to settle any disagreements in classification.

5.1.2.5 Apply software assurance requirements in Section 5 for the acquirer software assurance activities, based on both the results of the Software Assurance Classification Assessment and Table 1 for guidance.

5.1.2.6 Apply software assurance requirements in Sections 6 and 7 for the provider software assurance activities for each RFP/MOU/MOA, based on both the results of the Software Assurance Classification Assessment and Table 1 for guidance.

5.1.2.7 Assure contractual statements include appropriate oversight/insight requirements, including needed deliverables (e.g., records, documents, reports).

5.1.2.8 Prepare a preliminary acquirer program/project software assurance plan documenting the planned level of software assurance effort and activities required and the necessary resources using the template provided in Appendix B.

Downloaded from http://www.everyspec.com

5.1.2.9 Verify that the RFP/MOU/MOA address software quality metrics (see definition in Section 3.1 of the Standard).

5.1.2.10 Participate in the process to identify, analyze, track, and control procurement/development risks.

5.2 Post RFP, Pre-Award

Once the RFP has been released, the acquirer receives proposals and evaluates them.

5.2.1 The software assurance manager shall perform the following tasks:

5.2.1.1 Evaluate the proposals to verify that the software assurance requirements in the RFP have been addressed.

5.2.1.2 Participate in pre-award surveys when such surveys are requested.

5.2.1.3 Participate in contract negotiation to ensure that all software engineering, software assurance, management, and development requirements have been addressed and, where appropriate, are included in any resulting contracts.

5.2.1.4 Coordinate with project management to perform an updated Software Assurance Classification Assessment with the accepted proposal information and defined software assurance development approach.

5.2.1.5 Apply the updated Software Assurance Classification Assessment results to update the software assurance requirements.

5.2.1.6 Ensure that each Software Assurance Classification Assessment Report is maintained and made available to the SMA director, SMA office, SMO, project management, and/or Center Director upon request.

5.3 Post-Award, Pre-Implementation

Once the contract is awarded, it is important to verify that both the acquirer and provider's software assurance plans are complete and baselined.

5.3.1 The software assurance manager shall perform the following tasks:

5.3.1.1 Verify that the provider's software assurance plan meets contractual requirements.

5.3.1.2 Verify that the acquirer's software assurance plan and the provider's software assurance plan are consistent, compatible, and are baselined.

5.3.1.3 Ensure that acquirer software assurance personnel are trained and qualified to accomplish their tasks.

Downloaded from http://www.everyspec.com

5.3.1.4 Assure that provider software assurance personnel are trained and qualified to accomplish their tasks.

5.4 Contract Implementation, Development

During development, the acquirer performs tasks to provide surveillance (insight/oversight) [see NPR 8735.2, Management of Government Safety and Mission Assurance Surveillance Functions for NASA Contractors] into the software development processes and products.

5.4.1 The software assurance manager shall perform the following tasks:

5.4.1.1 Provide surveillance to assure that both the acquirer and provider software assurance functions are performed according to their specific software assurance plans and the contract.

5.4.1.2 Verify that the provider has developed and maintained processes for assurance of COTS, MOTS, and GOTS software addressing both the basic acquired software and any modifications or applications written to adopt them into the intended system.

5.4.1.3 Ensure that acquirer software assurance staff performs tasks to provide insight into whether the provider is adhering to approved software assurance, management, and development plans and procedures and that these plans and procedures are effectively fulfilling their purpose. These tasks may include activities such as audits, reviews, analyses, and assessments.

5.4.1.4 Ensure that acquirer software assurance staff performs tasks to provide oversight of the provider's management, assurance, and engineering processes. Specifically, reviews, audits, and evaluations may be performed to ensure adherence to and effectiveness of approved plans and procedures.

5.4.1.5 Assure that both deliverable and any designated non-deliverable software development products have proper configuration management.

5.4.1.6 Assure that problem reports, discrepancies from reviews, and test anomalies are documented, addressed, analyzed, and tracked to resolution.

5.4.1.7 Assure that software products (e.g., software requirements, preliminary design, detailed design, use cases, code, models, simulators, test data, inspection results, flow diagrams) are reviewed and software quality metrics (e.g., defect metrics) are collected, analyzed, trended, and documented.

Downloaded from http://www.everyspec.com

5.5 Acceptance

The acquirer's objective is to verify that the software and all related products (e.g., requirements, design, code, documentation, and special instructions) are complete and that they meet all of the specified requirements.

5.5.1 The software assurance manager shall perform the following tasks:

5.5.1.1 Ensure that an audit (e.g., Functional Configuration Audit, Physical Configuration Audit) is performed prior to delivery to assure that all delivered products are complete, contain the proper versions, and that all discrepancies, open work, and deviations and waivers are properly documented and approved.

5.5.1.2 Ensure that any acquirer facilities (e.g., buildings, hardware) are prepared to receive and install the software.

5.5.1.3 Assure that all acceptance documentation is present, including signed certifications.

5.5.1.4 Assure that all acquisition lessons learned are recorded and entered into the NASA lessons learned database.

5.6 Operation

During operation, software assurance's objective is to ensure that software assurance practices remain in place and are used.

5.6.1 The software assurance manager shall perform the following tasks:

5.6.1.1 Ensure that software assurance processes are in place for operation of the software developed or acquired by NASA. A separate Software Assurance Plan may be necessary as a new contract may cover the operational phase.

5.6.1.2 Depending upon the operational environment and the criticality of operation, ensure that software assurance processes include a periodic audit of the operations to ensure any changes to the software or software induced operational workarounds have been reviewed and approved.

5.6.2 Software assurance staff shall perform periodic operational assessments to ensure baseline management of software requirements, design, code, and documentation and to ensure review and approval of software changes or software induced operational workarounds.

NOTE: The period for review and/or the triggers (e.g., problem reports, updates) for software assurance review should be established and documented in the operational phase of the software assurance plan.

Downloaded from http://www.everyspec.com

5.7 Maintenance

Software assurance's objective is to assure that the provider of software maintenance applies software assurance according to this Standard.

5.7.1 The software assurance manager shall perform the following tasks:

5.7.1.1 Ensure that software assurance processes are in place for software maintenance.

5.7.1.2 Assure the transfer and maintenance of any licenses, simulators, models, and test suites from the developer to NASA, or the designated maintenance contractor.

5.7.1.3 Assure that any metrics collected on the software, along with any trending and reliability data, are transferred to the maintenance organization and maintained in order to better understand and predict problem areas in the software.

5.8 Retirement

Prior to retirement of software products, planning is performed to ensure the proper disposition of software and the software assurance records and documents created over the life of the program/project.

5.8.1 The software assurance manager shall perform the following tasks:

5.8.1.1 Assure that software engineering and management prepare, approve, and execute a retirement plan.

5.8.1.2 Ensure that the retirement plan includes archival and eventual disposal of software assurance records and documents created over the life of the program/project in accordance with the requirements of NPR 1441.1, NASA Records Retention Schedules.

Downloaded from http://www.everyspec.com

6. PROVIDER SOFTWARE ASSURANCE

The provider designs, develops, implements, tests, operates, and maintains the software products. The provider has software assurance requirements relative to those software engineering activities.

6.1 Software Assurance Program

This section addresses requirements of software assurance for the actual provider of the software products.

6.1.1 The provider shall plan, document, and implement a software assurance program for software development, operation, and maintenance activities. This includes documentation of software assurance procedures, processes, tools, techniques, and methods to be used.

6.1.2 The software assurance program shall include processes for assurance of COTS, MOTS, and GOTS software addressing both the basic acquired software and any modifications or applications written to adopt them into the intended system.

6.1.3 The software assurance program shall include the disciplines of Software Quality, Software Safety, Software Reliability, and Software V&V.

6.1.4 When IV&V has been selected for a project, the provider shall coordinate with IV&V personnel to share data and information.

6.1.5 The software assurance program shall describe what metrics will be collected and reported in regards to the software assurance program activities.

6.2 Software Assurance Management

6.2.1 The provider shall identify the person responsible for directing and managing the software assurance program; e.g., a software assurance manager.

NOTE: While the person identified with the responsibility for software assurance may have other titles, for this document, that person will be referred to as the software assurance manager.

6.2.2 The software assurance manager shall establish and maintain the interfaces with project management and ensure the working relationship between software assurance personnel and that of the project.

6.2.3 The software assurance manager shall have a reporting channel to provider management that is independent of the provider's project management and software development function.

6.2.4 The software assurance manager shall conduct and document periodic reviews of the software assurance process.

Downloaded from http://www.everyspec.com

6.2.5 The software assurance manager shall conduct and document periodic reviews, audits, and assessments of the development process and products.

6.2.6 The software assurance manager shall assure that problems and risks are reported, recorded, addressed, and tracked to closure.

6.3 Software Assurance Plan

6.3.1 Each software provider shall establish and maintain a software assurance plan that addresses all software development and maintenance activities.

NOTE: For smaller projects, this plan may be incorporated in another project planning document or may be a separate document. Larger projects may have a separate plan or more than one software assurance plan.

6.3.2 The software assurance plan shall:

6.3.2.1 Conform to IEEE 730-2002, IEEE Standard for Software Quality Assurance Plans.

6.3.2.2 In addition, address how the provider will implement the requirements of Sections 6.0 and 7.0 of this Standard.

6.3.2.3 If there is any conflict between Section 6.0 or Section 7.0 of this Standard and IEEE 730-2002, IEEE Standard for Software Quality Assurance Plans, this Standard shall take precedence.

6.4 Software Assurance Plan Change Procedures

6.4.1 The provider shall submit any proposed deviations from or modification to the baselined software assurance plan to the acquirer as a formal change request.

6.4.2 Proposed changes shall be accompanied by a risk analysis, as defined in NPR 7120.5, NASA Program and Project Management Processes and Requirements, to identify the potential impact of the change.

6.5 Software Assurance Approval Authority

The software assurance manager shall have approval authority on the establishment and composition of all software baselines and any changes to the baselines before submission to the acquirer. This includes changes to software plans, procedures, verification approaches, requirements, design, and code.

Downloaded from http://www.everyspec.com

6.6 Software Assurance Records

6.6.1 Software assurance records shall be prepared, maintained, placed under configuration management, and contain the descriptions and results of software assurance activities, (e.g., audit reports, classification evaluations, milestone review, software assurance briefings, problem reporting tracking).

6.6.2 Software assurance records shall include recommended preventive measures, corrective actions, and lessons learned.

6.7 Software Assurance Status Reporting

6.7.1 The provider shall prepare software assurance status reports that include:

a. Highlights of organization and key personnel changes.
b. Assurance accomplishments and resulting software assurance program metrics for activities such as inspection and test, reviews, contractor/subcontractor surveys, audits.
c. Subcontractor assurance accomplishments, including items listed above, plus summaries of acceptance and certification reports.
d. Significant problems, their status, solutions, and remedial and preventive actions.
e. Trends in software quality metric data (e.g., defect types, location, priority/criticality).
f. Plans for upcoming software assurance activities.
g. Recommendations and lessons learned.

6.8 Training

6.8.1 Personnel managing, developing, and implementing the software assurance process shall be trained and/or experienced in software assurance.

6.8.2 Software assurance training shall be obtained and/or originated and maintained for management, engineering, and assurance personnel.

6.8.3 Software assurance personnel shall be trained in relevant software engineering design methods and languages, processes, development environments, tools, test techniques, and other software engineering and assurance methods needed to stay current with the engineering environment and products they must assure.

6.8.4 Software assurance personnel shall be trained for the environment and operational particulars of the program/project to which they are assigned. This may include on-the-job training as well as orientation and specific engineering training.

6.8.5 Records shall be maintained and readily available for review (e.g., training, testing, and certification/recertification status of personnel).

Downloaded from http://www.everyspec.com

6.9 Subcontractor Controls

6.9.1 The provider shall flow down the requirements of this Standard to any subcontractor who develops, tests, maintains, operates, or provides services for the software.

6.9.2 The provider shall assure that the subcontractors satisfy the requirements of this Standard.

Downloaded from http://www.everyspec.com

7. SOFTWARE ASSURANCE DISCIPLINES

This section contains the requirements for the disciplines of software assurance.

Software assurance consists of the following disciplines:

- Software Quality
  - Software Quality Assurance
  - Software Quality Control
  - Software Quality Engineering
- Software Safety
- Software Reliability
- Software V&V
- IV&V.

Each assurance discipline brings its own perspective to the tasks; the collective effect of all these efforts provides assurance of mission safety, reliability, and quality. Software quality focuses on the software processes and products to ensure that quality has been built into the software products. Software safety identifies safety-critical software and systematically identifies, analyzes, tracks, mitigates, and controls software hazards and hazardous functions to ensure safer software operation within a system. Software reliability ensures that the fault tolerance and redundancy functions are appropriate and implemented correctly for the system; it measures the reliability of the system through defect data and the level of fault and failure detection. Software V&V ensures that software being developed or maintained satisfies functional, performance, and other requirements and that each process of the development process yields the right products. IV&V provides an objective examination of the provider's mission critical software processes and products of the system to ensure that right system has been built and that it has been built correctly. While each discipline has a specific focus, they interact with one another and do not duplicate activities.

7.1 Software Quality

Software quality is concerned with assuring that quality is built into the software products. Software quality assures creation of complete, correct, workable, consistent, and verifiable software plans, procedures, requirements, designs, and verification methods. Software quality assures adherence to those software requirements, plans, procedures, and standards to successive products. The software quality discipline consists of product assurance and process assurance activities that are performed by the functions of software quality assurance, software quality engineering, and software quality control.

7.1.1 **Product assurance** shall be performed to assure that:

7.1.1.1 All of the required plans (e.g., configuration management, risk management, provider's assurance plan, software management plan) are documented, adhere to applicable standards and procedures, are mutually consistent, and are being executed.

Downloaded from http://www.everyspec.com

7.1.1.2 All software requirements are defined, traceable from one life cycle phase to another, and analyzed in a manner that is measurable or otherwise verifiable.

7.1.1.3 Software products and related documentation have been evaluated, according to the software assurance plan.

7.1.1.4 Project documentation, including plans, procedures, requirements, design, verification documentation, reports, schedules, and records and any changes to them are reviewed for impact to the quality of the product.

7.1.1.5 Formal and acceptance software testing are witnessed by software assurance personnel to verify satisfactory completion and outcome.

7.1.1.6 Lower level testing results and software development folders are updated, audited, and/or reviewed for completeness.

7.1.1.7 Software quality metrics are in place and are used to ensure the quality and safety of the software products being delivered. Trends in software quality metrics are reported to assist in risk mitigation.

7.1.1.8 The software development plans specify the standards and procedures for management, acquisition, engineering, and assurance activities.

7.1.1.9 The software is verified (e.g., tested, analyzed, measured) for compliance with functional and performance requirements.

7.1.1.10 The status and quality of the software are presented at formal reviews.

7.1.1.11 Problems with products are reported during participation in formal and informal reviews (e.g., inspections, peer reviews, test readiness reviews, requirements reviews) along with regular reporting to project management and engineering during team meetings.

7.1.2 **Process assurance** shall be performed to assure that:

7.1.2.1 Those software life cycle processes employed for the project adhere to the applicable plans.

7.1.2.2 Problems found with implementation of the software life cycle processes, including management, engineering, and assurance, are documented, tracked, and resolved through the problem reporting and corrective action process and through discussions with the project manager.

7.1.2.3 The software engineering practices, development environment, test environment, and libraries employed for the project adhere to applicable standards and procedures.

7.1.2.4 Formal reviews and inspections are monitored and address software quality issues.

Downloaded from http://www.everyspec.com

7.1.2.5 All management, engineering, and assurance processes are audited for compliance with applicable plans.

7.1.2.6 The software quality metrics process is assessed for compliance to appropriate documentation or requirements. Trending is accomplished following the defined software quality metrics process.

7.2 Software Safety

While much of software safety depends on a good software development process and the overall software assurance activities, software safety is specifically concerned with the features of the software where failure could impact safety.

7.2.1 The requirements for NASA-STD-8719.13, NASA Software Safety Standard, shall be implemented.

7.2.2 Software safety tasks shall be coordinated between system safety program, software development, and software assurance to ensure completion of required tasks and elimination of duplicate efforts.

7.2.3 In the course of performing software assurance, any safety risks shall be communicated to the appropriate safety organization.

7.2.4 Periodic reviews and/or audits shall be conducted for compliance with the defined software safety process for acquisition, development, and assurance of safety-critical software.

7.3 Software Reliability

Software reliability is concerned with both incorporating reliability into the products starting with requirements and measuring the reliability of the products produced by each process of the life cycle. Measures, including number, type, location, and criticality of defects, and measures for software reliability modeling found in IEEE 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software, are selected as appropriate for project risk, security, safety requirements, cost, size, complexity, life span, consequence of failure, and other attributes. Software reliability requirements will address the level and manner of fault and failure detection, isolation, fault tolerance, and recovery expected to be fulfilled by the software as part of the overall system.

7.3.1 Software assurance shall assure that fault tolerance and redundancy have been specified, implemented correctly, and verified by testing.

7.3.2 Software reliability analyses and measurements, including trends and metric data, shall be included in appropriate status reports to the software assurance manager and project management. This data is to be used to trace and recommend actions on specific modules which may have less than desired reliability.

Downloaded from http://www.everyspec.com

7.3.3 Collection and classification of defects found during formal and informal reviews shall be maintained.

7.3.4 The use of software quality metrics shall be documented, monitored, analyzed and tracked during each stage of development and across development and operational phases. Examples include fault counts by severity levels, time between discovery and removal of faults, and number of faults found in a time period per lines of code or number of function points.

7.3.5 Trend analyses shall be performed on the software quality metrics and made available for lessons learned or root cause analyses.

7.4 Software V&V

Software V&V is concerned with ensuring that software being developed or maintained satisfies functional, performance, and other requirements (validation) and that each phase of the development process has been performed and yields the right products (verification). Every participant in the software life cycle process (e.g., acquirer, project management, engineering, and assurance) plays a role in some aspect of the V&V process.

7.4.1 Software assurance shall assure that software V&V activities occur according to established plans, policies, procedures, and standards.

7.4.2 Software assurance shall participate in the formal and informal reviews. Such activities include peer reviews, inspections, and milestone reviews (e.g., software requirements review, design reviews, test readiness reviews, certification readiness reviews).

7.4.3 Software assurance shall witness or review/audit results of software testing and demonstration.

7.4.4 Software assurance shall use defect data collected by the project to analyze software quality metrics.

7.4.5 Software assurance shall collect and maintain software assurance records showing the participation of software assurance staff in verification and validation efforts, such as minutes, records, artifacts, and signature on test reports.

7.4.6 Software assurance shall provide objective evidence to the project and NASA SMA of the software's readiness for operational release.

7.5 IV&V

Selection of software projects for IV&V support will be based on the results of the Software Assurance Classification Assessment. All software projects that are identified as safety-critical or software Class A will be candidates for IV&V support. All software projects that are identified as safety-critical will be the highest priority candidates for IV&V support unless a request is made from a Center IV&V liaison to the Office of Safety and Mission Assurance (OSMA) for exemption.

Downloaded from http://www.everyspec.com

Once a project is chosen for IV&V, exemption will require an assessment of the software project by the NASA OSMA and approval by the Chief Safety and Mission Assurance Officer.

Software projects that are neither Class A nor safety-critical but still request Headquarters coverage of IV&V on their projects will be considered only by special request from either the NASA OSMA, the IV&V Board of Directors, or by the IV&V Center Liaisons. Software intense projects not selected as candidates for IV&V by NASA OSMA may negotiate with the IV&V Facility to specify and fund IV&V for their projects directly.

When IV&V is either selected for a project by NASA Headquarters or chosen by the acquirers to have IV&V support on their projects, the following is performed:

7.5.1 All software projects that are identified as safety-critical or software Class A by using NPR 7150.2, Software Engineering Software Assurance Classification Assessment shall be candidates for IV&V with safety criticality as the highest criterion.

7.5.2 IV&V work shall be performed by the contractors selected and managed by the NASA IV&V Facility.

7.5.3 When the IV&V function is required, the provider shall provide all required information to NASA IV&V Facility personnel. (This requirement includes specifying on the contracts and subcontracts, IV&V's access to system and software products and personnel.)

7.5.4 The NASA IV&V Facility shall initially conduct a planning and scoping exercise to determine the specific software components to be analyzed and the tasks to be performed. The IV&V approach will be documented in an IV&V plan.

7.5.5 The IV&V team shall provide input to the appropriate software assurance personnel, as well as provide feedback to the project manager as agreed in the IV&V Plan.

Downloaded from http://www.everyspec.com

APPENDIX A. The Software Assurance Classification Assessment

The Software Assurance Classification Assessment was developed by NASA to identify and evaluate the characteristics of software in determining the software's classification and level of software assurance to be applied. This assessment is conducted for all NASA software. The process for assessing and classifying software consists of four steps, as shown in Figure A-1:

1. Ascertain if the software has safety implications for the system, property external to the system, or to the environment (Appendix A.1). If human life is a risk factor, please indicate this as well.

2. Determine the software engineering class of software (Class A-E) based on NPR 7150.2, NASA Software Engineering Requirements and copied in Appendix A.2 for convenience. If the software can be classified as Class E (which includes Exploratory), based on its intended use of the software, the projected customer base, and potential for release or software infusion (Appendix A.2) then no further software assurance determination is needed.

3. Then, using the Software Classification Score Sheet, expand upon the classification and obtain a criticality score based on project/mission characteristics, the tasks the software will perform, and any software unique characteristics (Appendix A.3). As some projects may have multiple software tasks, each may need to be assessed and classified separately. An example of the Software Classification Score Sheet is in Appendix A-3, however, the most current version should be obtained from the following URL:

http://swg.jpl.nasa.gov/assets/index.cfm?Display=Software%20Classification%20Score%20Sheet

4. Classify the software and software assurance effort based on the results from the preceding steps (Appendix A.4). An overall rating for the project software as well as any specific Classes for separate software tasks within a project is helpful in determining software assurance resources are appropriately applied.

Results from all steps are documented in a Software Assurance Classification Report and maintained as a quality record. A template for a Software Assurance Classification Report is provided in Appendix A.5.

Downloaded from http://www.everyspec.com

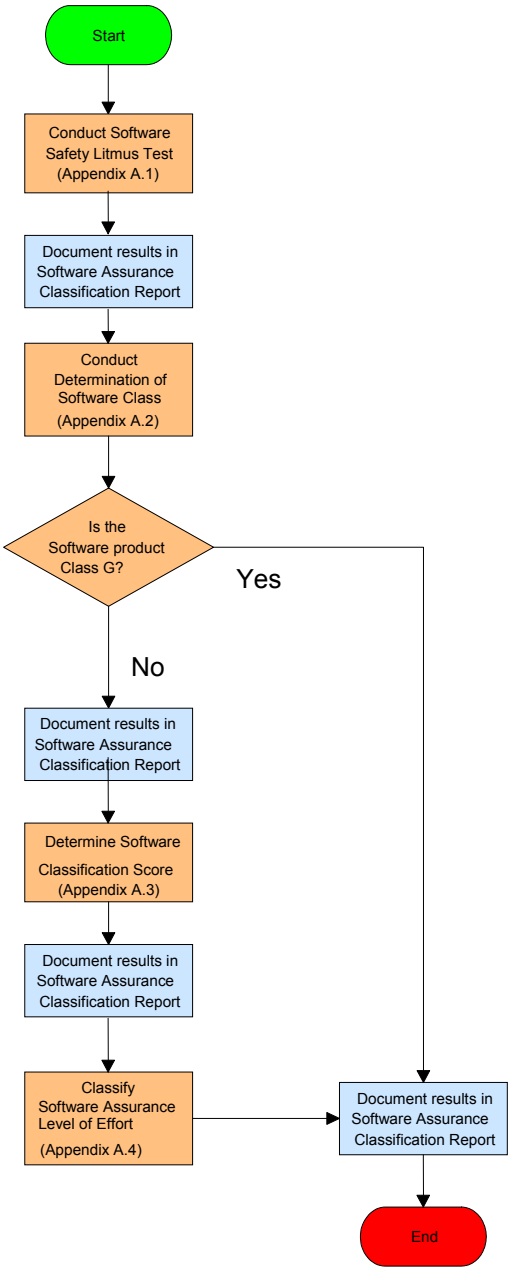

Figure A-1. Software Assurance Classification Assessment Process

Downloaded from http://www.everyspec.com

A.1 Software Safety Litmus Test

The Software Safety Litmus Test below is applied to all projects with software to determine if the software is safety-critical[1]. The software is considered safety-critical if it meets any of the following criteria:

a. Resides in a safety-critical system (as determined by a hazard analysis) AND at least one of the following apply:

(1) Causes or contributes to a hazard.
(2) Provides control or mitigation for hazards.
(3) Controls safety-critical functions.
(4) Processes safety-critical commands or data[2].
(5) Detects and reports, or takes corrective action, if the system reaches a specific hazardous state.
(6) Mitigates damage if a hazard occurs.
(7) Resides on the same system (processor) as safety-critical software[3].

b. Processes data or analyzes trends that lead directly to safety decisions (e.g., determining when to turn power off to a wind tunnel to prevent system destruction).

c. Provides full or partial verification or validation of safety-critical systems, including hardware or software subsystems.

If the software is determined to be safety-critical from the above criteria, then the project must adhere to the NASA-STD-8719.13, NASA Software Safety Standard.

Document the results of this test in the Software Assurance Classification Report.

---

[1] NASA defines safety-criticality from the definition of Hazard Severity in NPR 8715.3, NASA Safety Manual, Chapter 3, System Safety, and Appendix D Analysis Techniques. Hazard severity definitions are shown in Table A-1.

[2] If data is used to make safety decisions (either by a human or the system), then the data is safety-critical, as is all the software that acquires, processes, and transmits the data. However, data that may provide safety information but is *not* required for safety or hazard control (such as engineering telemetry) is not safety-critical.

[3] Non-safety-critical software residing with safety-critical software is a concern because it may fail in such a way as to disable or impair the functioning of the safety-critical software. Methods to separate the code, such as partitioning, can be used to limit the software defined as safety-critical. If such methods are used, then the isolation method is safety-critical, but the isolated non-critical code is not.

Downloaded from http://www.everyspec.com

Table A-1. Definitions for Hazard Severity

Hazard Severity Definitions	**Catastrophic** Loss of human life or permanent disability; loss of major system; loss of vehicle; loss of ground facility; severe environmental damage.	**Critical** Severe injury or temporary disability; major system, facility, equipment, or environmental damage.
	Moderate Minor injury; minor damage to systems, facilities, or equipment.	**Negligible** No injury or minor injury; some system stress, but no system damage.

Downloaded from http://www.everyspec.com

A.2 Determination of Software Class

Below is a summary of the Software Engineering Classes as defined in NPR 7150.2. Please refer to NPR 7150.2 for actual software classes and their current description.

Class A Human Rated Software Systems: 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 shall be 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.

Class B Non-Human Space Rated: Flight and ground software that must perform reliability 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.

Class C Mission Support Software: 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 the tracking complexes). Class C software must be developed carefully, but validation and verification effort is generally less intensive than for Class B.

Class D Analysis and Distribution Software: 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 and distribution systems.

Class E Development Support Software: 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.

Downloaded from http://www.everyspec.com

Exploratory Software (now a subset of Class E Software, is included here for consistency purposes)

The Exploratory software determination process consists of a series of questions that explore the intended use of the software and its potential for release or software infusion. The Exploratory classification is defined for software that will never be used by anyone other than the researcher/developer, and is strictly for purposes of investigating the feasibility of some aspect of research. Exploratory software is not distributed for use outside the development and usage group, either within a Center or externally, and is not used to operate equipment or facilities that are safety-critical. If any proven concept of the Exploratory software can no longer be classified as Exploratory, then it is reassessed according to this Standard.

To be classified as Exploratory software, all answers to Questions 1 through 7 must be "NO."

1. Will the software be released to others inside the Center, inside NASA, or externally to NASA? Release is defined as: to distribute a product intended for use outside of the development team for purposes other than Exploratory software development.
2. Does the software enable, analyze, or verify a product that the Center intends to release within NASA or externally to NASA?
3. Is the software intended for use in a deliverable project, facility, or larger system?
4. Will this software be maintained or expanded after it is put to practical use?
5. Does this software pose a safety risk to personnel or a facility?
6. Would lack of documentation (e.g., management/development plan, requirements, design, test plans, test reports, version description, user's manual) or configuration control of this software impair its use from the customer's perspective?
7. If this software was lost or made unusable, would it impact the Center's missions and objectives?

Classes F, G, and H are designated by the Chief Information Officer (CIO) As such, Software Assurance is only performed at this time upon request or as designated by the CIO.

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

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

Downloaded from http://www.everyspec.com

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 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.

Class H General Purpose Desktop Software 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). 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 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.

Document the results of this assessment in the Software Assurance Classification Report. If the assessment result identifies the software as Exploratory, then the Software Assurance Classification Assessment is complete.

Downloaded from http://www.everyspec.com

A.3 Software Classification Scoring Process

The Software Classification Scoring Process is a means to assess and capture the exact tasks that software will perform. The results from this process will be used to help determine the level of software assurance to apply. While Table A-2 is provided below with all the criteria and software tasks/functions identified, the user should obtain the Excel spreadsheet version from the NASA Headquarters website for Software Engineering, (URL: http://swg.jpl.nasa.gov/assets/index.cfm?Display=Software%20Classification%20Score%20Sheet)

The following steps should be used to determine the Classification Score for the project software being assessed. This Classification can be for the entire project or for subsystems of the project or program as appropriate. What software is being classified needs to be identified clearly. Comments and notes inserted into the Excel spread sheet are not only helpful but necessary when used for the Agency software listing and for help in determining if IV&V needs to be applied.

1. If a category or task does not apply, an "x" should be placed in the cell.
2. Identify the major system (mission category): e.g., Human rated, Robotic rated, Instruments, Ground Research Project, Information Systems & Data Analysis. Enter the scores provided in column I for the category identified.
3. Identify the Secondary System/Environment/Ops (project characteristics). Enter the scores provided in column J for each characteristic that the project possesses.
4. Identify the software tasks/functions (project subsystems). Enter the scores provided in column K for each subsystem. If the subsystem does not exist, then enter an "x" in the column. If the software task exists but is in a reduced functionality or is more critical, the score may be adjusted, however, a note must be attached explaining the reason for the change.
5. Provide any contextual information in the form of an engineering note, added at a component or project level. Enter engineering notes into the Excel worksheet by using the "insert comments" feature in Excel (footnotes are used in Table A-2 for illustration). Notes are used to explain special situations and to provide background information which might be helpful in determining the criticality, complexity, usage, heritage, and other details of the development or operations of the software.
6. If a component score is associated with a software component that is safety-critical, highlight the score in ***bold italics***.
7. Identify project source code information. Provide the language(s) used, the total source lines of code (SLOC) for each language, and the percent of new software for each language. Modified source code is considered new software.
8. Total the scores for the project and document it in the Software Assurance Classification Report.
9. Use the total score as an input for the Determination of Software Assurance Level of Effort in Appendix A.4.

Downloaded from http://www.everyspec.com

Table A-2. Software Classification Score Sheet

Major System	Secondary Systems/ Environment /Operations	Software Tasks/Functions	Assigned Values I	J	K	Example Robotics	Aero-nautics	Human Rated
Human Rated (Space)			50000			X	X	50000
	On Launch			3000		X	X	X
		Solid Rocket Propulsion			100	X	X	X
		Automated Control of Recoverable Launch Components			100	X	X	X
		Crew Escape			100	X	X	100
		Liquid Rocket Propulsion			90	X	X	X
		Aero-Surface Control			90	X	X	90
		Guidance, Navigation & Control			90	X	X	90
		Abort Moding & Selection			90	X	X	X
		Separation Control			90	X	X	X
		Range Safety			90	X	X	X
		Redundancy Management/Fault Detection, Isolation and Recovery			80	X	X	80
		Environmental Control and Life Support System			80	X	X	80
		Critical Systems Control			80	X	X	80
		Critical Systems Monitoring			80	X	X	80
		Crew Interface (Displays & Controls)			70	X	X	80
		Telemetry			60	X	X	60
		Communications & Tracking			60	X	X	60
Human Rated (Space)	On Orbit w/o Rendezvous			2000		X	X	2000
	On Orbit w/ Rendezvous			2500		X	X	X
	Long-duration On-Orbit			5000		X	X	5000
		Guidance, Navigation & Control			100	X	X	100
		Command & Control			100	X	X	100
		Crew Escape			100	X	X	100
		Automated Rendezvous & Docking			90	X	X	90
		Environmental Control and Life Support System			75	X	X	75

Downloaded from http://www.everyspec.com

Major System	Secondary Systems/ Environment /Operations	Software Tasks/Functions	Assigned Values I	J	K	Example Robotics	Aero-nautics	Human Rated
		Thermal Control			75	X	X	75
		Extra-Vehicular Activity			75	X	X	75
		External Control			75	X	X	75
		Electrical Power			75	X	X	75
		Critical Systems Control			75	X	X	75
		Critical Systems Monitoring			75	X	X	75
		Redundancy Management/Fault Detection, Isolation, and Recovery			60	X	X	60
		Telemetry			60	X	X	60
		Orbital Maneuvering System			60	X	X	60
		Crew Interface (Displays & Controls)			60	X	X	60
		Communications & Track			60	X	X	60
		Portable Computer System			40	X	X	40
		Robotic Systems			40	X	X	40
		Payload Control			40	X	X	40
	On Ground			1000		X	X	1000
		Pre-launch and Vehicle Processing Operations			80	X	X	80
Human Rated (Space)	On Ground	Launch Operations			100	X	X	100
		Mission Operations			90	X	X	90
		Entry Operations			100	X	X	100
	Ground Test Facilities			1000		X	X	X
		Liquid Rocket Propulsion			90	X	X	X
		Flight Simulator			90	X	X	X
		Command & Control of equipment under test			60	X	X	X
		Critical Systems Monitoring & Control of Facility			70	X	X	X
Human Rated (Aeronautics)			40000			X	40000	X
	Aerodynamic Flight			3000		X	3000	X
		Guidance, Navigation & Control			100	X	100	X
		Crew Escape			100	X	X	X
		Automated Terrain Following			100	X	X	X

Downloaded from http://www.everyspec.com

Major System	Secondary Systems/ Environment /Operations	Software Tasks/Functions	Assigned Values I	J	K	Example Robotics	Aero-nautics	Human Rated
		Traffic/Ground Collision Avoidance			100	X	100	X
		Propulsion			90	X	90	X
		Aero-Surface Control			90	X	90	X
		Range Safety			90	X	X	X
		Environmental Control and Life Support System			80	X	X	X
		Fuel Management			80	X	X	X
		Radar Control			80	X	X	X
		Automated Landing			80	X	X	X
		Redundancy Management/Fault Detection, Isolation, and Recovery			80	X	80	X
		In-Flight Refueling			80	X	X	X
Human Rated (Aeronautics)	Aerodynamic Flight	Critical Systems Control			70	X	70	X
		Critical Systems Monitoring			70	X	70	X
		Communications & Track			60	X	X	X
		Crew Interface (Displays & Controls)			60	X	60	X
		Payload Control			60	X	X	X
		Telemetry			60	X	X	X
	Ground Test Facilities			1000		X	X	X
		Flight Simulator			90	X	X	X
		Command & Control of equipment under test			60	X	X	X
		Critical Systems Monitoring & Control of Facility			70	X	X	X
Robotic Rated			28000			28000	X	X
	Spacecraft System			1000		1000	X	X
	Non-Spacecraft System			500		X	X	X
	Nuclear Power System			5000		X	X	X
		Propulsion			100	X	X	X
		Main Power Bus			100	X	X	X
	Earth Orbit			1500		1500	X	X
	Flyby			2000		X	X	X

Comment [DW1]: Telemetry for Aero project: NOTE: When doing the IV&V exercise we did not have the Ground Test Facilities entry with its software tasks/ functions as part of the scoring sheet. This number may have been increased by 1000 + 90 + 60 + 70.

Downloaded from http://www.everyspec.com

Major System	Secondary Systems/ Environment /Operations	Software Tasks/Functions	Assigned Values I	J	K	Example Robotics	Aero-nautics	Human Rated
	Planetary Body Orbit			3000		X	X	X
	Controlled Reentry			3000		X	X	X
	Autonomous Flight/Landing			3000		3000		
		Impactor			1000	X	X	X
		Coordination (intra-vehicle modes)			500	X	X	X
Robotic Rated	Autonomous Flight/Landing	Fault Protection			100	100[1]	X	X
		Augmented Autonomy			100	100[2]	X	X
		Non-Nuclear Propulsion			80	80[3]	X	X
		Hazard Prevention			70	X	X	X
		Guidance, Navigation & Control			70	70[4]	X	X
		Command & Control			60	60	X	X
		Power			50	50	X	X
		Cryogenics			50	X	X	X
		Sample Return Capsule			50	X	X	X
		Thermal			40	40	X	X
		Number of Instruments - 1 to 3			50	50	X	X
		Number of Instruments > 3			100	X	X	X
	Surface Operations (includes EDL)			3000		X	X	X
		Augmented Autonomy			100	X	X	X
		Propulsion			80	X	X	X
		Guidance, Navigation & Control			70	X	X	X
		Command & Control			60	X	X	X
		Power			50	X	X	X

[1] Minimum onboard FDC. Majority of decisions to switch to redundant on board hardware made on the ground.
[2] Full use of existing flight software with flight heritage.
[3] Propulsion not needed to meet minimum requirements. Required for the final orbit shift to meet 25-year lifetime orbit requirement.
[4] Reuse of existing flight software with small modifications. Somewhat added risk because of the criticality of the system.

Downloaded from http://www.everyspec.com

Major System	Secondary Systems/ Environment /Operations	Software Tasks/Functions	Assigned Values I	J	K	Example Robotics	Aero-nautics	Human Rated
		Thermal			40	X	X	X
		Fault Protection			100	X	X	X
		Number of Instruments - 1 to 3			50	X	X	X
		Number of Instruments > 3			100	X	X	X
	Ground System			1000		1000	X	X
		Command & Control			60	60[5]	X	X
Instruments			20000			X	X	X
	On Orbit			2000		X	X	X
	Non-orbiting			1000		X	X	X
		Command & Control			70	X	X	X
		Power			50	X	X	X
		Thermal			40	X	X	X
		Fault Protection			50	X	X	X
		Cryogenics			40	X	X	X
		On-board data compression/processing			50	X	X	X
		Science sequence engine			60	X	X	X
		Mechanisms			30	X	X	X
	Ground System			1000		X	X	X
		Command & Control			60	X	X	X
Ground Research Project			25000			X	X	X
	Propulsion Hardware Testing			1000		X	X	X
		Solid Rocket Propulsion			100	X	X	X
		Liquid Rocket Propulsion			90	X	X	X
		Command & Control			60	X	X	X
		Critical Systems Monitoring & Control			70	X	X	X
	Other Test Setups			800		X	X	X
		Command & Control			60	X	X	X
		Critical Systems Monitoring & Control			70	X	X	X

[5] Reuse of existing software with small updates to the database.

Downloaded from http://www.everyspec.com

Major System	Secondary Systems/ Environment /Operations	Software Tasks/Functions	Assigned Values I	J	K	Example Robotics	Aero-nautics	Human Rated
Information Systems & Data Analysis			10000			X	X	X
	Major (Agency-wide) Business Applications			8000				
	Minor (Center or Mission specific) Business Applications			2000				
	Ground Data Processing of Science Data			1000				
	Experimental Modeling of Systems			1000				
						35110	43660	60505
	Language (s) List all that apply					C, Assem-bly	C++	TBD
		Source Lines of Code				201330	7000	TBD
		% new				2.43%	80%	TBD

Downloaded from http://www.everyspec.com

A.4 Determination of Software Assurance Level of Effort

The final part of the Software Assurance Classification Assessment consists of applying the results of the previous assessment steps to determine the Software Class and criticality along with a few additional criteria of the project's software to determine the level of effort or prioritization for applying software assurance. The Additional Software Assurance Criteria in Table A-3 are used to help augment the Software Class for determination of software assurance level of effort and prioritization of software assurance resources. For each program/project, assess the potential risks of the project's software using the Software Assurance Effort/Prioritization Criteria on the left column of Table A-3. If the program/project's software acquisition and development meets the criteria identified on the left, then the corresponding software assurance level of effort on the right side will be assigned. When the software acquired or developed meets the classification criteria for more than one level of software assurance effort, then the highest level of software assurance will be applied. Safety critical software of any software class needs the most software assurance effort.

Downloaded from http://www.everyspec.com

Table A-3. Additional Software Assurance Criteria

	Level of Software Assurance Effort					
Software Assurance Effort/Prioritization Criteria	**Full/ High**	**Full/ Medium**	**Medium/ Medium**	**Minimum /Low**	**N/A**	**N/A or upon request**
Software Classes (from NPR 7150.2)						
Class A Human Flight	X					
Class B Non-human Flight		X				
Class C			X			
Class D				X		
Class E					X	
Class F, G, and H						X
Software Safety Criticality	X					
Potential for:						
Catastrophic Mission Failure[1]	X					
Partial Mission Failure[2]		X				
Potential for waste of resource investment: [3]						
Greater than 200 work-years on software	X					
Greater than 100 work-years on software		X				
Greater than 20 work-years on software			X			
Less than 20 work-years on software				X		
Potential for impact to equipment, facility, or environment: [4]						
Greater than $100M	X					
Greater than $20M		X				
Greater than $2M			X			
Less than $2M				X		
Software Classification Score from Table A-2. (K = 1000)	≥ 45K	≥ 30K	≥ 15K	≥ 5K		>5K

1. **Catastrophic mission failure**: Loss of vehicle or total inability to meet remaining mission objectives caused by software.

2. **Partial mission failure**: Inability to meet one or more mission objectives caused by software.

3. **Potential for waste of resource investment**: This is a measure or projection of the effort (in work-years: civil service, contractor, and other) invested in the software. The measure of effort includes all software life cycle phases (e.g., planning, design, maintenance). This shows the level of effort that could potentially be wasted if the software does not meet requirements.

4. **Potential for impact to equipment, facility, or environment:** This is a measure of the cost (in dollars) of the physical resources that are placed at risk of damage, destruction, or loss due to a software failure. Potential collateral damage is to be included. This is exclusive of mission failure.

Note: Potentials listed above can apply to both test and operational scenarios where software is a controlling factor.

Downloaded from http://www.everyspec.com

A.5 Software Assurance Classification Report Template

Table A-4 presents a template that may be used for the Software Assurance Classification Report.

Downloaded from http://www.everyspec.com

Table A-4. Software Assurance Classification Report Template

Software Assurance Classification Report								
1. Project Name				2. Date				
3. Project Manager								
4. Software Assurance Manager								
Software Assurance Classification Criteria								
5a. Software Safety Litmus Test Is the Software Safety-Critical? Is Human Life a Risk Factor?	Yes ☐ ☐			No ☐ ☐				
5b. Determination for Class E, F, G, or H Software If F or G, is SA being performed? OSMA Involvement?	Yes ☐ ☐ ☐			No ☐ ☐ ☐				
5c. Software Classification Score	Score:							
5d. Software Class	A ☐	B ☐	C ☐	D ☐	E ☐	F ☐	G ☐	H ☐
5e. Software Assurance Effort/Priority	Full/ High ☐	Full/ Medium -High ☐	Medium / Medium ☐	Minimal/ Low ☐	None ☐			
6. Comments								
7. Date	Signature of Software Assurance Manager							
8. Date	Signature of Project Manager							

Downloaded from http://www.everyspec.com

APPENDIX B. Acquirer Software Assurance Plan Template Outline

This template is for use in the development of an Acquirer Software Assurance Plan. The purpose of this plan is to document the software assurance activities to be performed by the acquirer as outlined in the NASA Software Assurance Standard.

1) Introduction

1.1) Purpose

Describe the purpose and objectives for this plan.

1.2) Scope

Describe the scope for this plan. Include the contract name (if contract exists), project name, and list of software items.

1.3) Document Organization

Briefly describe the contents of each major section within this document and the contents of each appendix.

2) Reference Documents

Provide a complete list of documents referenced elsewhere in the text of this document. Include policies, standards, and similar documents used in the development of this plan. Include dates and version numbers for each document.

3) Abbreviations and Acronyms

Provide an alphabetized list of the definitions for abbreviations and acronyms used in this document.

4) Organization and Management

Provide a description of the software assurance organizational structure, including the relationship to project management and the provider(s). Identify delegated organizations performing software assurance activities.

5) Software Assurance Program

5.1) Contract Award Activities

Provide a description of the software assurance planning activities leading up to contract award (if a contract exists).

Downloaded from http://www.everyspec.com

5.1.1) **Initialization, Pre-Award**

5.1.2) **Post RFP, Pre-Award**

5.1.3) **Post-Award, Pre-Development**

5.2) **Implementation Activities by Discipline**

Provide a description of the software assurance activities for each of the software assurance disciplines throughout the life cycle.

5.2.1) **Software Quality**

5.3.1.1) Product Assurance

5.3.1.2) Process Assurance

5.2.2) **Software Safety**

5.2.3) **Software Reliability**

5.2.4) **Software Verification and Validation**

5.2.5) **Independent Verification and Validation**

6) **Documentation**

Identify the documentation governing the development, acceptance, operation, maintenance, and retirement of the software.

7) **Problem Reporting and Corrective Action**

Provide a description of the practices and procedures for reporting, tracking, and resolving problems or issues.

8) **Risk Management**

Provide a description of the methods and procedures employed to identify, assess, monitor, and control areas of risk arising during the software assurance activities.

9) **Software Assurance Program Metrics**

Provide a description of the software assurance program metrics to be developed and maintained.

Downloaded from http://www.everyspec.com

10) **Software Assurance Records**

Provide a description of the software assurance documentation to be retained and the methods and facilities to assemble, file, safeguard, and maintain this documentation, including the retention period.

11) **Training**

Provide a description of the training activities necessary to meet the needs for implementing this plan.

12) **Glossary**

This section contains the glossary of terms that are unique to this plan.

13) **Document change procedure and history**

Provide a description of the procedures for modifying this plan and maintaining a history of the changes, including a history of all modifications.

Downloaded from http://www.everyspec.com

APPENDIX C. Requirements Compliance Matrix

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
Scope	1	*No requirements*	N/A				
Applicable Documents	2	*No requirements*	N/A				
Definitions And Acronyms	3	*No requirements*	N/A				
Software Assurance Overview	4	*No requirements*	N/A				
Acquirer Software Assurance	5	*Not a requirement*	N/A				
Initialization, Pre-Award	5.1	*Not a requirement*	N/A				
	5.1.1	Identify software assurance manager	Acquirer SMA Mgr				
	5.1.2	The software assurance manager shall perform the following tasks:					
	5.1.2.1	Perform Classification Assessment	Acquirer SA Mgr				
	5.1.2.2	Ensure safety critical software projects comply with NASA Software Safety Standard NASA-STD-8719.13B	Acquirer SA Mgr				
	5.1.2.3	Ensure tailoring of software assurance requirements based on software classification	Acquirer SA Mgr				
	5.1.2.4	Assure project agreement with classification	Acquirer SA Mgr				
	5.1.2.5	Apply acquirer software assurance requirements	Acquirer SA Mgr				
	5.1.2.6	Apply software assurance requirements are applied on provider in accordance with SA classifications	Acquirer SA Mgr				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	5.1.2.7	Assure contract contains oversight/insight and deliverable requirements	Acquirer SA Mgr				
	5.1.2.8	Prepare preliminary acquirer software assurance plan	Acquirer SA Mgr				
	5.1.2.9	Verify that the RFP/MOU/MOA addresses software quality metrics	Acquirer SA Mgr				
	5.1.2.10	Identify, analyze, track, and control procurement/development risks	Acquirer SA Mgr				
Post RFP, Pre-Award	5.2	*Not a requirement*	N/A				
	5.2.1	The software assurance manager shall perform the following tasks:					
	5.2.1.1	Evaluate proposals	Acquirer SA Mgr				
	5.2.1.2	Participate in pre-award surveys when such surveys are requested.	Acquirer SA Mgr				
	5.2.1.3	Participate in contract negotiations	Acquirer SA Mgr				
	5.2.1.4	Perform an updated Software Assurance Classification Assessment	Acquirer SA Mgr				
	5.2.1.5	Update software assurance requirements based on Assessment results	Acquirer SA Mgr				
	5.2.1.6	Maintain Assessment results	Acquirer SA Mgr				
Post-Award, Pre-Development	5.3	*Not a requirement*	N/A				
	5.3.1	The software assurance manager shall perform the following tasks:					
	5.3.1.1	Verify provider's software assurance plan meets contractual requirements.	Acquirer SA Mgr				
	5.3.1.2	Verify acquirer's and provider's software assurance plans are consistent, compatible, and are baselined	Acquirer SA Mgr				
	5.3.1.3	Ensure acquirer software assurance personnel are trained and qualified	Acquirer SA Mgr				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	5.3.1.4	Assure provider software assurance personnel are trained and qualified	Acquirer SA Mgr				
Contract Implementation, Development	5.4	*Not a requirement*	N/A				
	5.4.1	The software assurance manager shall perform the following tasks:					
	5.4.1.1	Assure both acquirer and provider software assurance organizations perform according to their plans	Acquirer SA Mgr				
	5.4.1.2	Verify provider has developed and maintained processes for assurance of COTS, MOTS, and GOTS software	Acquirer SA Mgr				
	5.4.1.3	Ensure insight performed over provider	Acquirer SA Mgr				
	5.4.1.4	Ensure oversight performed over provider	Acquirer SA Mgr				
	5.4.1.5	Assure proper software configuration management	Acquirer SA Mgr				
	5.4.1.6	Assure software problems, discrepancies, anomalies are documented and tracked to resolution	Acquirer SA Mgr				
	5.4.1.7	Assure software products are reviewed and assure that software quality metrics are collected and analyzed	Acquirer SA Mgr				
Acceptance	5.5	*Not a requirement*	N/A				
	5.5.1	The software assurance manager shall perform the following tasks:					
	5.5.1.1	Ensure an acceptance audit is performed prior to delivery	Acquirer SA Mgr				
	5.5.1.2	Ensure that any acquirer facilities are prepared to receive and install the software	Acquirer SA Mgr				
	5.5.1.3	Assure all acceptance documentation is complete	Acquirer SA Mgr				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	5.5.1.4	Assure acquisition lessons learned are recorded and entered into the NASA lessons learned database	Acquirer SA Mgr				
Operation	5.6	*Not a requirement*	N/A				
	5.6.1	The software assurance manager shall perform the following tasks:					
	5.6.1.1	Ensure software assurance processes are in place for operation of the software developed or acquired by NASA	Acquirer SA Mgr				
	5.6.1.2	Ensure software assurance processes include a periodic audit of the operational software	Acquirer SA Mgr				
	5.6.2	Ensure software configuration management of operational software	Acquirer SA Mgr				
Maintenance	5.7	*Not a requirement*	N/A				
	5.7.1	The software assurance manager shall perform the following tasks:					
	5.7.1.1	Ensure software assurance processes are in place for software maintenance.	Acquirer SA Mgr				
	5.7.1.2	Assure transfer and maintenance of any licenses, simulators, models, and test suites	Acquirer SA Mgr				
	5.7.1.3	Assure that any software metrics are transferred to the maintenance organization and maintained	Acquirer SA Mgr				
Retirement	5.8	*Not a requirement*	N/A				
	5.8.1	The software assurance manager shall perform the following tasks:					
	5.8.1.1	Assure that software engineering and management prepare, approve, and execute a retirement plan.	Acquirer SA Mgr				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	5.8.1.2	Ensure that the retirement plan includes proper archival and disposal of software assurance records and documents	Acquirer SA Mgr				
Provider Software Assurance	6	*Not a requirement*	N/A				
Software Assurance Program	6.1	*Not a requirement*	N/A				
	6.1.1	Plan, document, and implement software assurance program	Provider SA Mgr				
	6.1.2	Include software assurance processes for COTS, MOTS, and GOTS software	Provider SA Mgr				
	6.1.3	Include all software assurance disciplines	Provider SA Mgr				
	6.1.4	Coordinate with IV&V	Provider SA Mgr				
	6.1.5	Describe SA metrics collection and reporting	Provider SA Mgr				
Software Assurance Management	6.2	*Not a requirement*	N/A				
	6.2.1	Identify provider software assurance manager	Provider Mgmt				
	6.2.2	Establish and maintain interface between software assurance and project	Provider Mgmt Provider SA Mgr				
	6.2.3	Establish an independent reporting channel to provider management	Provider Mgmt Provider SA Mgr				
	6.2.4	Conduct and document periodic reviews of provider software assurance process	Provider Mgmt Provider SA Mgr				
	6.2.5	Conduct and document periodic reviews, audits, and assessments of the development process and products	Provider SA Mgr				
	6.2.6	Assure software problems and risks are documented and tracked to resolution	Provider SA Mgr				
Software Assurance Plan	6.3	*Not a requirement*	N/A				
	6.3.1	Establish and maintain a software assurance plan	Provider Mgmt Provider SA Mgr				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	6.3.2	The software assurance plan shall:					
	6.3.2.1	Conform plan to IEEE 730-2002	Provider Mgmt, Provider SA Mgr				
	6.3.2.2	Implement requirements of provider software assurance and software assurance disciplines sections into plan	Provider Mgmt, Provider SA Mgr				
	6.3.2.3	Give precedence of software assurance Standard sections over IEEE 730-2002	Provider Mgmt, Provider SA Mgr				
Software Assurance Plan Change Procedures	6.4	*Not a requirement*	N/A				
	6.4.1	Submit plan deviations or changes formally to acquirer	Provider SA Mgr				
	6.4.2	Perform and submit risk analysis of deviations or changes to plan	Provider SA Mgr				
Software Assurance Approval Authority	6.5	Have approval authority on the establishment and composition of all software baselines and any changes to the baselines	Provider SA Mgr				
Software Assurance Records	6.6	*Not a requirement*					
	6.6.1	Prepare, maintain, and manage configuration of software assurance records	Provider SA Mgr				
	6.6.2	Include recommended preventive measures, corrective actions, and lessons learned in software assurance records	Provider SA Mgr				
Software Assurance Status Reporting	6.7	*Not a requirement*	N/A				
	6.7.1	Prepare software assurance status reports	Provider SA Mgr				
Training	6.8	*Not a requirement*	N/A				
	6.8.1	Ensure that software assurance personnel are trained and/or experienced	Provider SA Mgr				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	6.8.2	Obtain software assurance training for management, engineering, and software assurance personnel	Provider SA Mgr				
	6.8.3	Ensure software assurance personnel training is current with assurance and development methods	Provider SA Mgr				
	6.8.4	Ensure that software assurance personnel are trained for their assigned environment	Provider SA Mgr				
	6.8.5	Ensure training records are available and maintained	Provider SA Mgr				
Subcontractor Controls	6.9	*Not a requirement*	N/A				
	6.9.1	Flow down the requirements of this Standard to all subcontractors	Provider SA Mgr				
	6.9.2	Assure that the subcontractors satisfy the flowed down requirements	Provider SA Mgr				
Disciplines	7	*Not a requirement*	N/A				
Software Quality - Product Assurance	7.1	*Not a requirement*	N/A				
	7.1.1	Product assurance shall be performed to assure that:					
	7.1.1.1	All of the required plans are documented, adhere to applicable standards and procedures, are mutually consistent, and are being executed.	Acquirer and Provider SA Engr				
	7.1.1.2	All software requirements are defined, traceable from one life cycle phase to another, and analyzed	Acquirer and Provider SA Engr				
	7.1.1.3	Evaluate software products and related documentation	Acquirer and Provider SA Engr				
	7.1.1.4	Project documentation and any changes to them have been reviewed for impact to the quality of the product	Acquirer and Provider SA Engr				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	7.1.1.5	Witness formal and acceptance-level software testing	Acquirer and Provider SA Engr				
	7.1.1.6	Update, audit, and/or review lower level testing results and development folders	Acquirer and Provider SA Engr				
	7.1.1.7	Software quality metrics are in place and are used to ensure the quality and safety of the software products.	Acquirer and Provider SA Engr				
	7.1.1.8	Standards and procedures for management, acquisition, engineering, and assurance activities are specified	Acquirer and Provider SA Engr				
	7.1.1.9	Verify software is compliant with functional and performance requirements	Acquirer and Provider SA Engr				
	7.1.1.10	Present the status and quality of the software at formal reviews	Acquirer and Provider SA Engr/SA Mgr				
	7.1.1.11	Report problems with software products at formal and informal reviews	Acquirer and Provider SA Engr/SA Mgr				
Software Quality - Process Assurance	7.1.2	Process assurance shall be performed to assure that:					
	7.1.2.1	Those software life cycle processes employed for the project adhere to the applicable plans.	Acquirer and Provider SA Engr				
	7.1.2.2	Document, track, and resolve problems found with the implementation of software life cycle processes	Acquirer and Provider SA Engr/SA Mgr				
	7.1.2.3	The software engineering practices, development environment, test environment, and libraries employed for the project adhere to applicable standards and procedures.	Acquirer and Provider SA Engr				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	7.1.2.4	Formal reviews and inspections are monitored and address software quality issues.	Acquirer and Provider SA Engr				
	7.1.2.5	Audit all management, engineering, and assurance processes for compliance with applicable plans.	Acquirer and Provider SA Engr				
	7.1.2.6	Assess the software quality metrics process for compliance to appropriate documentation or requirements.	Acquirer and Provider SA Engr				
Software Safety	7.2	*Not a requirement*	N/A				
	7.2.1	Implement the requirements for NASA-STD-8719.13, NASA Software Safety Standard	Acquirer and Provider				
	7.2.2	Coordinate software safety tasks between system safety personnel and software safety personnel	Acquirer and Provider SA Mgr				
	7.2.3	Communicate any safety risks to the appropriate safety organization	Acquirer and Provider SA Mgr				
	7.2.4	Conduct periodic reviews and/or audits for compliance with the defined software safety process	Acquirer and Provider SA Mgr				
Software Reliability	7.3	*Not a requirement*	N/A				
	7.3.1	Assure that fault tolerance and redundancy have been specified, implemented correctly, and verified by testing.	Acquirer and Provider SA Engr				
	7.3.2	Include in appropriate status reports, software reliability analyses, and measurements	Acquirer and Provider SA Engr				
	7.3.3	Maintain the collection and classification of defects found during/from software assurance and programmatic/project formal and informal reviews	Acquirer and Provider SA Engr				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	7.3.4	Document, monitor, analyze, and track the use of software quality metrics during each stage of development and across development and operational phases	Acquirer and Provider SA Engr				
	7.3.5	Perform trend analyses on software quality metrics	Acquirer and Provider SA Engr				
Software Verification and Validation	7.4	*Not a requirement*	N/A				
	7.4.1	Assure that software verification and validation activities occur according to established plans, policies, procedures, and standards.	Acquirer and Provider SA Engr				
	7.4.2	Participate in the formal and informal reviews.	Acquirer and Provider SA Engr				
	7.4.3	Witness or review/audit results of software testing and demonstration.	Acquirer and Provider SA Engr				
	7.4.4	Collect and use defect data to analyze software quality metrics.	Acquirer and Provider SA Engr				
	7.4.5	Collect and maintain software quality records showing the participation of software assurance staff in verification and validation efforts	Acquirer and Provider SA Engr				
	7.4.6	Provide objective evidence to the project and NASA SMA of the software's readiness for operational release.	Acquirer and Provider SA Mgr				
Independent Verification and Validation	7.5	*Not a requirement*	N/A				
	7.5.1	All software projects that are identified as safety-critical or software Class A by the Software Assurance Classification Assessment shall be candidates for IV&V with safety criticality as the highest criterion.	IV&V				

Downloaded from http://www.everyspec.com

NASA-STD-8739.8 Requirements Compliance Matrix							
Section	**No.**	**Requirement**	**Role/Responsibility***	**Compliance**			**Comments**
				Full	**Partial**	**None**	
	7.5.2	IV&V work shall be performed by the contractors selected and managed by the IV&V Facility.	IV&V				
	7.5.3	When the IV&V function is required, the provider shall provide all required information to NASA IV&V Facility personnel. (This requirement includes specifying on the contracts and subcontracts, IV&V's access to system and software products and personnel.)	Provider Mgmt				
	7.5.4	The IV&V Facility shall initially conduct a planning and scoping exercise to determine the specific software components to be analyzed and the tasks to be performed. The IV&V approach will be documented in an IV&V plan.	IV&V				
	7.5.5	The IV&V team shall provide input to the appropriate software assurance personnel, as well as provide feedback to the project manager as agreed in the IV&V Plan.	IV&V				

*** Role/Responsibility Definitions:**

Center SMA Director	Center Safety and Mission Assurance Director
IV&V	IV&V Facility
Mgmt	Program/Project/Facility Management
SA Mgr	Software Assurance Manager
SWA Plan	Requirements for what is included in the Software Assurance Plan
SA Engr	Software Assurance Engineer assigned responsibility for Software Assurance activity