NASA TECHNICAL STANDARD

NASA-STD-8739.9

National Aeronautics and Space Administration
Washington, DC 20546-0001

Approved: 06-17-2013
Superseding NASA-STD-2202-93

SOFTWARE FORMAL INSPECTIONS STANDARD

MEASUREMENT SYSTEM IDENTIFICATION:
NOT MEASUREMENT SENSITIVE

DOCUMENT HISTORY LOG

Status	Document Revision	Approval Date	Description
Baseline		04-xx-1993	Initial Release
		03-29-2001	Revalidation
Revision	New document number.	06-17-2013	General Revision. Revisions made to address feedback received since the last revision, incorporate new research and best practices, and to align better with NPR 7150.2, NASA Software Engineering Requirements. Changes include: 1. Addressing projects' concerns related to: Software Safety, COTS, and Software Acquisition, 2. Providing more guidance for tailoring inspections for different types of artifacts (e.g. project plans, auto-generated code), 3. Rewording best practices as recommendations rather than requirements, and 4. Removing requirements addressed in other standards. *(MW)*
Change	1	10-07-2016	Added paragraph numbers and new section headers throughout, corrected subsequent paragraph and section numbers in Figure 1 for cross references, corrected typos in paragraphs: 6.3.5.2 and 7.6.2.f, and deleted "(Requirement)." *(MW)*
Revalidated		04-09-2018	Updated references to NPR 7150.2. Removed references to NASA-GB-A302 (cancelled). Updated CMMI language and applicable practice areas referenced for CMMI V2.0. Moved Applicable and Reference Documents from Appendix A to Section 2.

FOREWORD

This Standard is published by the National Aeronautics and Space Administration (NASA) to provide uniform engineering and technical requirements for processes, procedures, practices, and methods that have been endorsed as standard for NASA programs and projects, including requirements for selection, application, and design criteria of an item. This Standard is approved for use by NASA Headquarters and NASA Centers, including Component Facilities and Technical and Service Support Centers.

This Standard establishes requirements for software formal inspections across all NASA Centers, programs, projects, and facilities. It describes the activities necessary to ensure the effectiveness of software formal inspections.

This Standard supersedes NASA-STD 2202-93, Software Formal Inspections Standard, dated April 1993 and revalidated March 29, 2001. Changes in software technology, software development methodology, and the field of computing necessitated updating this Standard. Changes to this Standard were informed by research conducted by the NASA Office of Safety and Mission Assurance and community feedback received since its last revision. Requirements for new technology and methodology areas, such as commercial off-the-shelf software, software reuse, and security, are included. The changes also better align this Standard with NPR 7150.2, NASA Software Engineering Requirements.

Although other industry standards exist concerning software reviews and inspections, this NASA Standard: 1) aligns with existing NASA terminology and practices; 2) explicitly requires a consistent approach which utilizes the best practices shown to be effective through research and practice at NASA; 3) avoids potential confusion regarding which process variants are required under which circumstances; 4) explicitly discusses tailoring to the various types of inspections that occur in the NASA context; and 5) assures that NASA projects use and maintain processes and tools suitable for the state of the practice.

Requests for information, corrections, or additions to this Standard should be submitted to the National Aeronautics and Space Administration, Director, Safety and Assurance Requirements Division, Office of Safety and Mission Assurance, Washington, DC 20546 or via "Feedback" in the NASA Standards and Technical Assistance Resource Tool at http://standards.nasa.gov.

Terrence W. Wilcutt
Chief, Safety and Mission Assurance

6/17/2013
Approval Date

TABLE OF CONTENTS

DOCUMENT HISTORY LOG 2
TABLE OF CONTENTS 4
LIST OF APPENDICES 5
LIST OF FIGURES 5
LIST OF TABLES 5

1. SCOPE 6
1.1 Purpose 6
1.2 Applicability 6
1.3 Requirement Relief 7

2. APPLICABLE AND REFERENCE DOCUMENTS 7
2.1 Applicable Documents 7
2.2 Reference Documents 7

3. ACRONYMS AND DEFINITIONS 9
3.1 Acronyms and Abbreviations 9
3.2 Definitions 10

4. SOFTWARE FORMAL INSPECTION PROCESS 15
4.1 Formal Inspection Description 16
4.2 Characteristics 16
4.3 Comparison to Other Review Practices 17
4.4 Process Conformance and Tailoring 17

5. ROLES AND RESPONSIBILITIES 18
5.1 Qualifications and General Responsibilities 18
5.2 The Formal Inspection Team 18
5.3 Candidates for Formal Inspectors 23

6. PROCESS ELEMENTS 24
6.1 Input and Output 24
6.2 Entry and Exit Criteria 25
6.3 Process Stages 26
6.4 Formal Inspection Procedure Customization 33

7. TYPES OF INSPECTIONS 34
7.1 General 34
7.2 Software Plan Inspections 35
7.3 System Requirements Inspections (R0) 36
7.4 Software Requirements Inspections (R1) 37
7.5 Architectural Design Inspection (I0) 38
7.6 Detailed Design Inspection (I1) 40
7.7 Source Code Inspections (I2) 41
7.8 Test Plan Inspection (IT1) 42

7.9 Test Procedure Inspection (IT2) 43
7.10 Considerations on Inspections of Software that Runs on Programmable Logic Devices Operating Systems 45
7.11 Considerations for Acquisition Projects 46
7.12 Other quality considerations 48

8. PROCESS EVALUATION 48
8.1 Product and Process Measures 48
8.2 Evaluation Analyses 50
8.3 Process Improvement 51

LIST OF APPENDICES

APPENDIX A. INSPECTION TYPE AND PARTICIPANTS MATRIX 52
APPENDIX B. FORMAL INSPECTION COMPLIANCE MATRIX 54

LIST OF FIGURES

Figure 1. Inspection Process 26

LIST OF TABLES

Table 1. Data Collected at Inspection Points 48

SOFTWARE FORMAL INSPECTIONS STANDARD

1. SCOPE

1.1 Purpose

1.1.1 The purpose of this Standard is to define the requirements for a software inspection process aimed at detecting and eliminating defects as early as possible in the software life cycle. This process can be used for any documented product; however, this Standard focuses on its use for software products—i.e., software code, plans, manuals, etc. The process provides for the collection and analysis of inspection data to improve the inspection process as well as the quality of the software.

1.1.2 This Standard provides a core set of requirements that are applicable whenever formal inspections are required. The Software Assurance Electronic Handbook (SAEHB) provides additional information on approaches for implementing a software formal inspection process. The implementation and approach to meeting these requirements will vary to reflect the system to which they are applied.

1.2 Applicability

1.2.1 This standard will be used to insure NASA maintains the rigor and benefits of software formal inspections, when Software Formal Inspections are to be performed on software as specified by agreement or project direction. This Standard is applicable to formal inspections of software products during the development life cycle of software developed, maintained, or acquired by or for NASA, including the incorporation of open source, auto-generated code and test procedures, Commercial Off-The-Shelf (COTS), Government Off-The-Shelf (GOTS), or Modified Off-The-Shelf (MOTS) into NASA systems. Legacy and reuse software products are also covered with a focus on how they fit into the systems under current development. Projects need to choose which software products they will perform software formal inspection on, which will receive other kinds of peer reviews, and which will receive no peer review. These decisions should be documented in the program/project/facility software development or management plan.

1.2.2 This Standard is approved for use by NASA Headquarters and NASA Centers, including Component Facilities and Technical and Service Support Centers, and may be cited in contract, program, and other Agency documents as a technical requirement. This Standard may also apply to the Jet Propulsion Laboratory or to other contractors, grant recipients, or parties to agreements only to the extent specified or referenced in their contracts, grants, or agreements.

1.2.3 Requirements—i.e., mandatory actions—are denoted by statements containing the term "shall" and are numbered [SFI-###]. The terms: "may" or "can" denote discretionary privilege or permission, "should" denotes a good practice and is recommended, but not required, "will" denotes expected outcome, and "are/is" denotes descriptive material.

1.2.4 The project manager is usually called out as the responsible party for ensuring formal inspections are performed on their projects, and for the quality of the formal inspections. Project managers are not expected to personally perform nor run the actual Software Formal Inspections. It is recognized that these requirements and activities may either be delegated by the project manager to a software lead, Software Formal Inspection chief moderator, software assurance lead within the project; or it could be the responsibility of a division or Center Software Engineering Process Group, Software Assurance, or other responsible party assigned this role. The project manager is used within this standard as the one responsible for using Software Formal Inspections (SFIs) on a project and thus is responsible for supporting the principles of SFIs for their projects and the monitoring and improvement of SFI to achieve reduced software risks.

1.3 Requirement Relief

Once invoked, relief from requirements in this Standard for application to a specific program or project shall be formally documented as part of program or project requirements and approved by the Technical Authority [SFI- 001]. This will be in accordance with procedures in NPR 8715.3, paragraph 1.13 and NASA-STD 8709.20, Management of Safety and Mission Assurance Technical Authority.

2. APPLICABLE AND REFERENCE DOCUMENTS

2.1 Applicable Documents

The documents listed in this section contain provisions that constitute requirements of this standard as cited in the text. Use of more recent issues of cited documents may be authorized by the responsible Technical Authority.

2.1.1 Government Documents

NPR 7150.2	NASA Software Engineering Requirements
NPR 8715.3	NASA General Safety Program Requirements
NASA-STD-8709.20	Management of Safety and Mission Assurance Technical Authority (SMA TA) Requirements
NASA-STD-8719.13	Software Safety Standard
NASA-HDBK-2203A	NASA Software Engineering Handbook

2.1.2 Non-Government Documents

None

2.2 Reference Documents

The reference documents listed in this section are not incorporated by reference within this standard, but may provide further clarification and guidance.

2.2.1 Government Documents

NPD 2810.1	NASA Security Information Technology
NPD 8700.1	NASA Policy for Safety and Mission Success
NPR 7123.1	NASA Systems Engineering Processes and Requirements
NPR 8000.4	Agency Risk Management Procedural Requirements
NASA-STD-8709.22	Safety and Mission Assurance Acronyms, Abbreviations, and Definitions
NASA-STD-8739.8	Software Assurance Standard
JSC 31011	
NASA-GB-8719.13	NASA Software Safety Guidebook

2.2.2 Non-Government Documents

IEEE 1028-2008	IEEE Standard for Software Reviews and Audits
IEEE 12207.0	Standard for Information Technology: Software life-cycle processes
SEI-CMMI	Software Engineering Institute Capability Maturity Model Integration™

Basili, V., Green , S., Laitenberger, O., Lanubile, F., Shull, F., Soerumgaard, S., and Zelkowitz, M. "The Empirical Investigation of Perspective-Based Reading." Empirical Software Engineering: An International Journal, 1(2): 133-164, 1996

Ciolkowski, C., Differding, C., Laitenberger, O., and Muench, J. Empirical Investigation of Perspective-based Reading: A Replicated Experiment, International Software Engineering Research Network, Technical Report ISERN-97-13, 1997

Fagan, M.E., "Design and Code Inspections to Reduce Errors in Program Development", IBM Systems Journal, Volume 15, Number 3, 1976

Kelly, J.C., Sherif, J.S., Hops, J. (1992) An Analysis of Defect Densities Found During Software Inspections. Journal of Systems & Software 17(2):111–117

Laitenberger, O., Atkinson, C., Schlich, M., El Emam, K. An experimental comparison of reading techniques for defect detection in UML design documents, Journal of System and Software, 53 (2000), 183-204

Laitenberger, O., El Emam, K., and Harbich, T. An Internally Replicated Quasi-Experimental Comparison of Checklist and Perspective-based Reading of Code Documents, IEEE Transactions on Software Engineering, 2000

Shull, F., Seaman, C., and Diep, M., "Analyzing Inspection Data for Heuristic Effectiveness," Proc. International Symposium on Empirical Software Engineering and Measurement (ESEM),Short paper track, Lund, Sweden, September 2012.

Wiegers, K., Peer Reviews in Software, Boston: Addison-Wesley, 2008.

Zhang, A., Basili, V., and Shneiderman, B. Perspective-based Usability Inspection: An Empirical Validation of Efficacy, Empirical Software Engineering: An International Journal 4(1), 43-70 (March 1999)

3. ACRONYMS AND DEFINITIONS

3.1 Acronyms and Abbreviations

ASIC	Application-Specific Integrated Circuit
CMMI	Capability Maturity Model Integration
COTS	Commercial Off-The-Shelf
DR	Discrepancy Report
FMEA	Failure Modes and Effects Analysis
FPGA	Field Programmable Gate Array
FTA	Fault Tree Analysis
HDL	Hardware Description Language
IEEE	Institute of Electrical and Electronics Engineers
IV&V	Independent Verification and Validation
GOTS	Government Off-The-Shelf
LC	Life Cycle
MOTS	Modified Off-The-Shelf
NASA	National Aeronautics and Space Administration
NPD	NASA Policy Directive

NPR	NASA Procedural Requirements
OSMA	Office of Safety and Mission Assurance
OTS	Off-The-Shelf
PHA	Preliminary Hazard Analysis
SMA	Safety and Mission Assurance
SEI	Software Engineering Institute
SFI	Software Formal Inspection
SFIS	Software Formal Inspection Standard
SA	Software Assurance
SoC	System-On-a-Chip
STD	Standard
SW	Software

3.2 Definitions

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.

Analysis: A method used to verify requirements that are more complex than can be verified by inspection, demonstration, or test. Analysis involves technical review of mathematical models, functional or operational simulation, equivalent algorithm tests, or other detailed engineering analysis.

Application: A group of software elements: components or modules that share a common trait by which they are identified to the persons or departments responsible for their development, maintenance, or testing.

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.

Author: The individual who created or maintains a work product that is being inspected. (see Wiegers 2008)

Checklist: A list of procedures or items summarizing the activities required for an operator or technician in the performance of duties. A condensed guide. An on-the-job supplement to more detailed job instructions.

Component: A constituent element of a system or subsystem.

Configuration Control: The systematic control of work products.

Configuration Item: An aggregation of hardware, software, or both, that is established and baselined, with any modifications tracked and managed. Examples include requirements document, data block, Use Case, or unit of code.

Defect: Any occurrence in a software product that is determined to be incomplete or incorrect relative to the software requirements, expectations for the software, and/or program standards.

Defect Classification: The process where all defects identified during an inspection are classified by severity and type.

Discrepancy: A formally documented deviation of an actual result from its expected result.

Discrepancy Report: An instrument used to record, research, and track resolution of a defect found in a baseline.

Element: The generic term applied to the smallest portions of a software or document product that can be independently developed or modified.

Environment: The components and features that are not part of the product but necessary for its execution such as software, hardware, and tools. (see JSC 31011)

Error: The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. (see IEEE Std 610.12-1990)

Failure: The behavior of the software or system component when a fault is encountered, producing an incorrect or undesired effect of a specified severity.

Fault: A manifestation of an error in software. If encountered, a fault may cause a failure.

Fault Detection: The ability to perform checks to determine whether any erroneous situation has arisen.

Fault Recovery: The response of the system software to an abnormal condition, so that system execution can continue to yield correct results despite the existence of the fault.

Firmware: The combination of a hardware device and computer instructions and/or data that reside as read-only software on that device. This term is sometimes used to refer only to the hardware device or only to the computer instructions or data, but these meanings are deprecated. The confusion surrounding this term has led some to suggest that it be avoided altogether. For the purposes of this Standard Firmware is considered as software. Firmware is NOT the same as Programmable Logic Devices/Complex Electronics.

Formal Inspection: A set of practices used to perform inspections in a precise, repeatable manner which includes the following specific Inspection process steps: (1) Planning, (2) Overview, (3) Preparation, (4) Inspection meeting, (5) Rework, and (5) Follow-up. It also has built in self-improvement process which includes the collection of data with which one can analyze the effectiveness of the process and track and make changes.

Inspection: A technical evaluation process during which a product is examined with the purpose of finding and removing defects and/or discrepancies as early as possible in the software life cycle.

Inspection Package: The physical and/or electronic collection of software products and corresponding documentation presented for inspection as well as required and appropriate reference materials.

Inspection Report: A report used to document and communicate the status (such as time and defect data) of a software formal inspection.

Inspector: Participant in an inspection.

Interface: The boundary, often conceptual, between two or more functions, systems, or items, or between a system and a facility, at which interface requirements are set.

Moderator: The individual who leads an inspection. Responsible for planning the inspection events with the author, scheduling and facilitating meetings, collecting and reporting measurements from inspections he/she moderates, and possibly verifying the author’s rework. (see Wiegers 2008)

Module: A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, input to, or output from, an assembler, compiler, linkage editor, or an executive routine. (see IEEE Std 610.12-1990)

Off-The-Shelf (OTS) Software: Software not developed for the specific project now underway. The software is considered general purpose or developed for a different project or projects. If Commercially created - COTS; created by another government source - GOTS; Modified prior to use - MOTS. OTS may include legacy, heritage, and re-use software.

Peer Review: [1] 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. [2] Independent evaluation by internal or external subject matter experts who do not have a vested interest in the work product under review. Peer reviews can be planned, focused reviews conducted on selected work products by the producer's peers to identify defects and issues prior to that work product moving into a milestone review or approval cycle.

Performance: A measure of how well a system or item functions in the expected environments.

Phase: The period of time during the life cycle of a project in which a related set of software engineering activities is performed. Phases may overlap.

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.

Reader: An inspection participant who describes the product being inspected to the other inspectors during the inspection meeting. (see Wiegers 2008)

Recorder: An inspection participant who documents the defects and issues brought up during the inspection meeting. (see Wiegers 2008)

Release ID: Identification code associated with a product's version level.

Reliability: The probability that a system of hardware, software, and human elements will function as intended over a specified period of time under specified environmental conditions.

Requirement: A precise statement of need intended to convey understanding about a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. The set of all requirements forms the basis for subsequent development of the system or system components.

Severity: A degree or category of magnitude for the ultimate impact or effect of executing a given software fault, regardless of probability.

Software: Computer programs, procedures, scripts, rules, and associated documentation and data pertaining to the development and operation of a NASA component or computer system. Software includes programs and data. This also includes COTS, GOTS, MOTS, reused software, auto generated code, embedded software, firmware, the software which runs on programmable logic devices (PLDs) operating systems, and open source software components.

Software Assurance: The planned and systematic set of activities that ensure that software life cycle processes and products conform to requirements, standards, and procedures. For NASA this includes the disciplines of Software Quality (functions of Software Quality Engineering, Software Quality Assurance, Software Quality Control), Software Safety, Software Reliability, Mission Software Cybersecurity Assurance, Software Verification and Validation, and IV&V.

Software Engineering: The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software: that is, the application of engineering to software.

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 (LC) 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. There are many development LC possibilities including waterfall, agile, model based, and others, all development LCs can reside within the total software lifecycle. The SW LC, usually resides within a system LC and may be the same as the system LC or can be standalone since SW can be updated when the system may not be able to be updated.

Software Product: Software products include requirements, design, code, plans, user manual, etc.

Software System Structure: The specific organization of a software system's components for the purpose of accomplishing an operational capability.

Source Code: The collection of executable statements and commentary that implements detailed software design.

Specification: A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior or other characteristics of a system or component, and, often the procedures for determining whether these procedures have been satisfied. (see IEEE Std 610.12-1990)

System: The combination of elements that function together to produce the capability required to meet a need. The elements include hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose.

Tailoring: [1] The process of assessing the applicability of requirements and evaluating the project's potential implementation in order to generate a set of specific requirements for the project. [2] The process used to refine or modify an applicable requirement by the implementer of the requirement. If the revised requirement meets/exceeds the original requirement, and has no increase in risk from that of the original requirement, then it may be accepted/implemented by appropriate local authority; otherwise a waiver/deviation may be required.

Test Plan: A document prescribing the approach to be taken for intended testing activities. The plan typically identifies the items to be tested, the testing to be performed, test schedules, personnel requirements, reporting requirements, evaluation criteria, the level of acceptable risk, and any risk requiring contingency planning.

Test Procedure: The detailed instructions for the setup, operation, and evaluation of results for a given test. A set of associated procedures is often combined to form a test procedure document.

Traceability: [1] The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor successor or master-subordinate relationship to one another; for example, the degree to which the requirements and design of a given software component match (see IEEE Std 610.12-1990). [2] The characteristic of a system that allows identification and control of relationships between requirements, software components, data, and documentation at different levels in the system hierarchy.

Validation: Confirmation that the product, as provided (or as it will be provided), fulfills its intended use. In other words, validation ensures that "you built the right thing." (see SEI-CMMI)

Verification: Confirmation that work products properly reflect the requirements specified for them. In other words, verification ensures that "you built it right." (see SEI-CMMI)

Walkthrough: A static analysis technique in which a designer or programmer leads members of the development team and other interested parties through a software product, and the participants ask questions and make comments about possible anomalies, violation of development standards, and other problems. (see IEEE Std. 1028-2008)

Work Product: The output of a task. Formal work products are deliverable to the acquirer. Informal work products are necessary to an engineering task but not deliverable. A work product may be an input to a task.

4. SOFTWARE FORMAL INSPECTION PROCESS

This Standard describes a general, standardized formal inspection process that should be customized to a specific project's context. The project manager shall use established and documented formal inspection procedures that meet the requirements specified in this Standard. [SFI-002] Those procedures may be established at a higher organizational level than the project and then modified, if necessary, for a specific inspection by the moderator.

Rationale: The requirements specified in this Standard represent best practices drawn from the experiences and lessons learned from hundreds of inspections across the NASA Centers, as well as being in broad agreement with the processes described in other software industry standards, such as IEEE Std. 1028-2008, "IEEE Standard for Software

Reviews and Audits." Inspections done with full process rigor (as described in this Standard) have a return on investment among the highest of all software engineering practices, since the effort spent on performing the inspection and fixing the defects early is often balanced by less effort spent on rework later, when defects have propagated and are more expensive to fix. This Standard also describes a well-defined process for software inspections encompassing activities that would satisfy requirements for CMMI Maturity Level 2, specifically addressing practice areas such as "Monitoring and Control," "Process Quality Assurance," and "Peer Reviews."

4.1 Formal Inspection Description

As applied to software products and associated documentation, inspection is a technical evaluation process during which a product is examined with the purpose of finding and removing defects and discrepancies as early as possible in the software life cycle. Formal Inspection is a set of practices used to perform inspections in a precise, repeatable manner which includes the following specific Inspection process steps: (1) Planning, (2) Overview, (3) Preparation, (4) Inspection meeting, (5) Rework, and (6) Follow-up. It also has a built in self-improvement process which includes the collection of data with which one can analyze the effectiveness of the process and track and make changes.

4.2 Characteristics

The following are characteristics of formal inspections (hereinafter called inspections):

a. Performed with the expectation that all major defects found will be corrected before they are propagated to further products.

b. Planned and performed according to established procedures and schedules.

c. Performed on mature yet draft products and partial products if the product is too large.

d. Performed by inspectors knowledgeable about the inspection process and the inspected product, and having a vested interest in the high quality of the inspected product.

e. Conducted by at least three (3) people, one of whom, the moderator, is responsible for the effectiveness of the inspection.

f. Participated in by the producer of the software product (author).

g. Participated in by inspectors who assume specific roles.

h. Executed in a specific set of stages, as described in Section 6.3, "Process Stages".

i. Performed so as to produce data for project management, quality evaluation, and inspection process improvement, but not for personnel evaluation.

4.3 Comparison to Other Review Practices

4.3.1 Peer reviews, as used in this Standard, refer to the general practice of examining work products to find defects and identify improvement opportunities by someone other than the author(s). The term “peer reviews” encompasses multiple approaches that vary in formality, including walkthroughs and formal inspections. This Standard describes the characteristics, requirements, and practices specific to formal inspections.

4.3.2 Audits and walkthroughs cannot be used as replacements for software inspections. These types of reviews have a different focus than software inspections but could be used in conjunction with inspection. An audit focuses on finding violations to product and process compliance. When audits are conducted by personnel external to the project, they can provide additional objectivity into the assurance activity. Walkthroughs are effective techniques for increasing product comprehension.

4.4 Process Conformance and Tailoring

4.4.1 The applicability of the requirements within this Standard is determined by the class and safety criticality of software that corresponds to the work products to be inspected. Project managers have a choice as to review processes to follow and need to determine which software products receive what kind of review (formal inspections, peer reviews, audits, walkthroughs, etc.).

> *Note: Refer to NASA NPR 7150.2 Appendix “Software Classification” to determine the software classification; and the Appendix “Requirements Mapping Matrix” for the applicability of the Software Peer Reviews/Inspections requirements given the software class. Refer also to NASA STD 8719.13 to begin the process of determining the software safety criticality and the applicability of the requirements to software safety personnel.*

4.4.2 Project managers can tailor and customize the inspection process described in this Standard for each inspection, or set of inspections, to address the needs for process rigor as well as the context of the project. The inspection is tailored by identifying requirements that are not applicable to the project and adding requirements to meet project needs. The formal inspection process is customized as prescribed by Section 6.4, “Formal Inspection Procedure Customization.” However, this standard sets for the minimum requirements and suggested processes for obtaining the rigor and results that come from the recognized formal inspection process.

4.4.3 Project managers shall document all the instances of non-compliances to the process requirements laid out in this Standard and the rationale for each. [SFI-003] Refer to NPR 8715.3 and NASA-STD 8709.20 for requirements regarding the creation of waivers and deviations. The impact to the project due to these non-compliances should be examined by the SMA organization to understand the risks and to ensure that they are acceptable.

> *Rationale: The requirements of this Standard have been formulated so as to be compliant to NASA best practices and other NASA requirements and standards (such as the NPR 7150.2). Multiple studies in the scientific literature have demonstrated a high degree of*

defect removal effectiveness from performing inspections with a process such as the process specified here. Moreover, both experiences on NASA projects and studies in the scientific literature have shown that omitting the key process steps required by this Standard can result in diminished effectiveness of the inspection. For example, studies show that when reviewers do not have checklists or clear quality focus (i.e., a "perspective" for their review) in the preparation step, inspection teams can be 1/3 less effective at finding defects. (This impact due to perspectives was measured in a study with NASA engineers (see Basili 1996); those results have since been replicated with universities (see Ciolkowski 1997), other US government agencies (see Zhang 1999), and companies (see Laitenberger, Atkinson, et al., 2000, and Laitenberger, El Emam, et al., 2000), to give just a few examples.)

4.4.4 The tailored inspection process and the rationale for tailoring should be documented (e.g., if the project is tailoring the process, it should be reflected in the software management plan), and the documentation made available to all project personnel, including the software providers. Appendix B provides a compliance matrix to assist the process of tailoring the requirements.

5. ROLES AND RESPONSIBILITIES

5.1 Qualifications and General Responsibilities

5.1.1 All participants in a formal inspection meeting—i.e., inspectors—shall examine the product presented for inspection and related materials, looking for defects in the product. [SFI-004] The participants in the inspection process are referred to as inspectors.

5.1.2 All inspectors should have sufficient training in the inspection process to be able to execute their responsibilities within that process. While taking a software formal inspection class is recommended for all inspectors, training on the basic principles prior to a software formal inspection as part of the planning and overview can and should be provided by the moderator. Having experienced inspectors who have participated in previous formal inspections is desirable, however, having been placed as a non-participatory observer on one or more inspections is beneficial when official training is not available.

5.1.3 Only persons who have been formally trained in inspections by taking a NASA-recommended class, and have already participated as an inspector in more than one inspection, shall fill the role of inspection moderator. [SFI-005]

5.1.4 All inspectors should have the technical expertise to be able to inspect the product.

5.2 The Formal Inspection Team

5.2.1 The formal inspection team shall consist of no less than three inspectors including a moderator and the author. [SFI-006] The moderator, once chosen, assigns additional inspectors. It is recommended that the inspection team size not exceed six members. However, when an inspection is in need of a greater coverage of expertise, the team size may go as high as ten members.

Rationale: These numbers reflect the fact that teams of less than 3 people are likely to lack important perspectives on the document, while teams larger than 10 are more likely to experience dynamics that make members less likely to be fully active participants – not all inspectors will easily get the chance to speak up, for example. Data from across the Agency have shown that inspections that do not follow this rule find significantly fewer defects.

5.2.2 Managers should not be chosen as part of the inspection team unless they are the authors of the inspected work products or if their expertise is essential to the inspection.

Rationale: Having managers participate in an inspection may hamper defect discovery and discussion (hence reducing the effectiveness of the inspection), if team members fear that defects found may be used to evaluate the author of the work product or if the team is intimidated by the manager's position. If managers must participate in an inspection (e.g., on very small teams), both managers and the other inspectors must be aware that defects found during inspections should never be used to evaluate authors.

5.2.3 Inspectors shall perform the following basic activities for each inspection (referring to the process steps in Figure 1): [SFI-007]

a. During Preparation, review the materials for the inspection.

b. Provide a list of defects, comments and questions on the material under review to the moderator by the assigned time prior to meeting.

c. Participate in the Formal Inspection Meeting, bringing up defects found during preparation and discussing any additional defects found during the meeting.

d. Support the Moderator in classifying and properly recording the defects.

e. Provide support to the author for understanding defect comments upon request.

5.2.4 Inspectors shall fulfill the following minimum set of roles at each inspection: Author, Moderator, Reader, and Recorder. [SFI-008]

5.2.5 Individual inspectors may fulfill more than one inspection role, but the author may not serve as moderator nor reader, and managers may not serve as moderator. The author may only serve as recorder when there are only three inspectors.

Rationale: An important part of the effectiveness of an inspection lies in the fact that it is an objective assessment of a work product. Having an author serve as the inspection moderator can reduce the objectivity and allow biases and misconceptions from the product's creation to be reflected in the product of the inspection team. The author cannot be assigned as the reader to avoid the tendency to read what the author thinks is meant for the work product to contain, rather than what the work product actually states. The recorder should ensure that defects are recorded objectively and as a true reflection of the team consensus. Authors can inadvertently reflect their own biases rather than the *team's view when recording defects. However, when there are only three inspectors it may be preferable for the author to serve as recorder*.

5.2.6 Inspection Moderator

5.2.6.1 The moderator is the conductor and controller of an inspection. The moderator is responsible for the overall effectiveness of the inspection.

5.2.6.2 Specific responsibilities of the moderator include:

a. Choose a qualified team of inspectors that include the author and supporting experts for the software product being inspected.

b. Ensure that the entry criteria specified in Section 6.2,"Entry and Exit Criteria," are met.

c. Ensure that appropriate time periods are set and met for preparation, inspection, follow-up, and third hours.

d. Ensure that all inspectors are prepared prior to the inspection meeting.

e. Focus the inspection meeting on finding defects in the product under inspection.

f. Classify defects as described in Section 6.3.6.7 "Defect Classification."

g. Assess the need for a re-inspection based on established criteria and technical assessment.

h. Once the author fixes the defects found during the inspection meeting, they are reviewed by the Moderator who verifies that all major defects found during inspection, and any other defects agreed during the inspection as needing to be fixed, are corrected prior to re-inspecting the product or authorizing placement of the inspected product under configuration control for delivery to the next phase of the software life cycle.

i. Authorize placement of the inspected product under configuration control (when all conditions in Section 6.2,"Entry and Exit Criteria," have been met) for delivery to the next phase in the software life cycle.

j. Collect the data, and generate and file the inspection report specified in Section 8.1, "Product and Process Measures."

k. Ensure that all defined processes and procedures associated with the inspection are followed.

5.2.7 Recorder

In addition to looking for defects in the product presented for inspection, the recorder documents each defect identified during the inspection meeting, including its classification, and provides the resulting list to the moderator and the author at the end of the inspection meeting.

5.2.8 Reader

The reader is the presenter of the inspection product to the other inspectors. In addition to looking for defects in the product presented for inspection, the reader leads the other inspectors through the inspected product and related materials in a logical progression, paraphrasing and summarizing each section.

5.2.9 Author

The author is the producer of the product being inspected. In addition to looking for defects in the product presented for inspection, the author is responsible for:

a. Generating the work product(s) to be inspected and providing required reference materials for the overview and the inspection meeting.

b. Responding to questions about the function, purpose, and organization of the inspected product and the associated reference materials.

c. Understanding and accepting the defects found during the inspection, their dispositioning, and proposed solutions (when provided); and modifying the inspected product to correct the defects.

d. Reviewing the corrections with the moderator according to the requirements in Section 6.3.10, "Follow-Up."

5.2.10 Software Assurance, including Software Safety

Software Assurance (SA) personnel will verify compliance with the requirements of this Standard. Software Assurance personnel work for a Safety and Mission Assurance organization, separate from Engineering and project management. Software Assurance personnel provide independent reviews of software requirements (including those flowed down from the Agency such as this standard), products and processes and provide the project and requirements owners feedback concerning whether software requirements, plans and processes are correct, comprehensive and the extent and effectiveness to which they are followed. (See NASA-STD-8739.8 for a more complete description of software assurance roles and responsibilities.)

a. Software Assurance personnel will verify compliance of the performed inspection to the documented, tailored and customized inspection procedures by:

(1) Verifying that the data specified in Section 8.1, "Product and Process Measures" have been collected.

(2) Selectively reviewing inspection packages for required inspection materials.

(3) Participating in inspection meetings, including fulfillment of any of the inspection roles.

(4) Performing, participating in, and assuring the analysis in Section 8.2, "Evaluation Analyses" which will provide an independent evaluation of the effectiveness of the inspection process and the product quality.

b. Software Assurance personnel will assure that:

(1) Reports of inspection process evaluations and analyses are:

(a) Defined and scheduled.

(b) Provided as needed to:

(i) Validate positive trends in the inspection process.

(ii) Address adverse trends in the inspection process.

(c) Reviewed with appropriate management and technical personnel.

(d) Considered in inspection process improvements.

(2) All improvements to the inspection process are documented and tracked for analysis and incorporation, and that inspection anomalies are documented and tracked for analysis and correction. This information is given to the organization responsible for the level of SFI process effected by these possible improvements.

Note: The noted improvement(s) may be at the project level, the division, Center, and even headquarters level if it can be applied across the Agency.

c. Software Safety personnel will participate in the inspections of safety-critical software products to:

(1) Ensure compliance to requirements defined in NASA-STD 8719.13.

(2) Ensure that preventive and safety measures are being implemented.

(3) Verify that requirements include error detection and recovery methods.

(4) Validate fault detection, identification, mitigation and recovery requirements.

5.2.11 Supporting Roles

This section provides guidance on additional considerations that can increase the effectiveness of the software inspection process as implemented at a given Center. The following supporting roles may be considered to provide support outside of the inspection process:

a. Chief Moderator: Develops and implements an effective, efficient, consistent inspection process for his or her center or organization; and serves as the overall inspection process owner by coordinating the inspection process, inspection training, and tools developed for inspections such as checklists and forms.

b. Educator: Provides training and certification of the software inspection process to inspectors, moderators, and managers. Educators may provide extended training for moderators by observing moderators during their first three inspections and providing feedback.

c. Data Manager: Assists a project in maintaining a software inspection metrics database. The data manager can work with the moderators from different projects at the Center to ensure that data is collected, and assists them in analyzing their inspection data.

d. Librarian: Assists the moderator and data manager in:

(1) Scheduling overview and inspection meetings.

(2) Distributing inspection packages.

(3) Providing reference documentation for inspections.

(4) Storing and retrieving project inspection data.

(5) Providing data to the data manager.

e. Scheduler: Assists the project managers in setting up inspections and choosing suitable teams of inspectors and moderators on a rotating basis.

5.3 Candidates for Formal Inspectors

5.3.1 Identify Stakeholders

5.3.1.1 As a basis for the selection of inspectors, the inspection moderator shall identify the key stakeholders in the product under inspection. [SFI-009]

5.3.1.2 Key stakeholders are persons with one or more of the following characteristics:

a. Will perform software development and verification work based upon the work product under inspection, and whose work could be affected by defects in this work product.

b. Have performed key technical work that was an input to the work product under inspection, and which needs to be reflected correctly in this work product.

c. Have knowledge about system-level or hardware-related needs and constraints, which the software component must satisfy.

d. Have other key technical expertise that is required for this work product to be evaluated. For example, developers working on work products which interface with this one, users of the system to be built using this work product, or domain experts.

e. Represent an independent entity to ensure the compliance of the performed inspection.

f. Have the key technical knowledge and expertise to determine software safety, reliability, and quality of the work product

5.3.2 Identify Formal Inspectors

5.3.2.1 The project manager will select a moderator that can meet the responsibilities in Section 5.2.6.

5.3.2.2 The inspection moderator should select inspectors in such a way that the perspective of each key stakeholder is represented on the team. This may be performed in consultation with the author to identify appropriate individuals. In such cases, the moderator should ensure that objectivity in the selection is maintained. One individual may represent multiple perspectives.

5.3.2.3 The moderator will ensure that all inspectors participating in the inspection are prepared. The moderator will take steps to ensure that last-minute substitution of inspectors, if it must occur, has little or minimal impact to the quality of inspection. If a key stakeholder perspective has been identified as important for the work product under inspection, the inspector representing that perspective must be present and prepared for each relevant stage of the inspection process.

5.3.3 Assign Responsibilities

5.3.3.1 The moderator shall ensure that each inspector has a checklist or other work aid (e.g., set of questions, review scenario) available. [SFI-010] The items on each checklist should be relevant to the assigned perspective and serve as reminders of the types of quality problems for which to look. Thinking beyond the checklist is not only encouraged but expected.

5.3.3.2 The quality problems listed on the work aid should be based on the quality issues of most importance to the development team and the type of work product. Quality issues may be drawn from checklists already approved for use at NASA, from historical data about previous quality issues on the same or similar projects, or from expert judgment. Different work aids may be provided for different inspectors. (That is, each inspector may have a tailored set of quality issues on which to focus.)

6. PROCESS ELEMENTS

The inspection procedure is established with the following elements. It can be established at the Center, organizational, program, or project level. The authority for the appropriate level of designating these formal inspection procedures needs to work with their software community to establish these process elements and document them. Once established, the moderators work to ensure they are followed for each inspection or to adjust them if needed.

6.1 Input and Output

The inspectors should have access to the following inputs before proceeding with the inspection:

a. The work product(s) to be inspected (provided by the author).

b. Any reference documents (or the location of those documents) that have been used by the authors for development of the software products, e.g., higher level work products, standards, guidelines, or materials from previous inspection for re-inspection (provided by the author).

c. Checklists and a set of work aids, as described in Section 5.3.3, “Assign Responsibilities.” Refer to Section 7, “Types of Inspections”, for typical checklist items.

d. Any other work products as decided by the moderator.

6.2 Entry and Exit Criteria

6.2.1 The moderator shall specify in the inspection procedure a set of measurable actions that must be completed prior to each type of inspection. [SFI-011] The moderator needs to work in concert with the project manager to assure their expectations are part of the entry and exit criteria. Most of the general entry and exit criteria will be established in a Center, organizational, program, or project level inspection procedure and then any specific criteria needed per inspection would be added and discussed at a pre-inspection meeting. Completion of these actions ensures that all activities related to the preceding phase of the software life cycle have been completed or addressed prior to the corresponding inspection.

Note: The following are examples of entry criteria:

a. A cursory check by the moderator on the work product demonstrates that it is complete and conforms to the project-defined formats and templates.

b. A cursory check by the moderator on the work product demonstrates that it is relatively free of defects (e.g., < 1 major defect per page).

c. The document has been spell-checked.

d. When the work product is code, there is proof that the code has been compiled and is verified free from compiler errors.

6.2.2 Furthermore, the moderator shall specify in the inspection procedure a set of measurable actions that must be completed following each of the required inspections. [SFI-012] These actions are to ensure that all major defects have been corrected. The moderator needs to work in concert with the project manager to assure their expectations are met as well. The moderator is responsible for ensuring that the defined exit criteria are met upon exiting the Follow-Up phase, the final phase in an inspection (see Section 6.3.10, “Follow-Up”).

Note: The following are examples of exit criteria:

a. All the required inspection activities were performed and documented as appropriate.

b. Inspection reports and forms, including any change requests, waivers, and deviations, are completely filled out and filed.

c. All major defects found in the inspection, as well as any non-major defects that the inspection team has indicated must be corrected, were verified as fixed.

The moderator should expect the author to make a good faith effort to fix the remaining minor defects, but they are not absolutely necessary to close out the inspection.

6.3 Process Stages

6.3.1 Figure 1 illustrates an activity diagram for the inspection process. The number in parentheses corresponds to the related section in this document.

6.3.2 All inspectors should participate in each relevant inspection stage according to their responsibilities as defined in Section 5.2, “The Inspection Team”. The moderator may need to delay any inspection stage for which all the needed inspectors are not available or prepared.

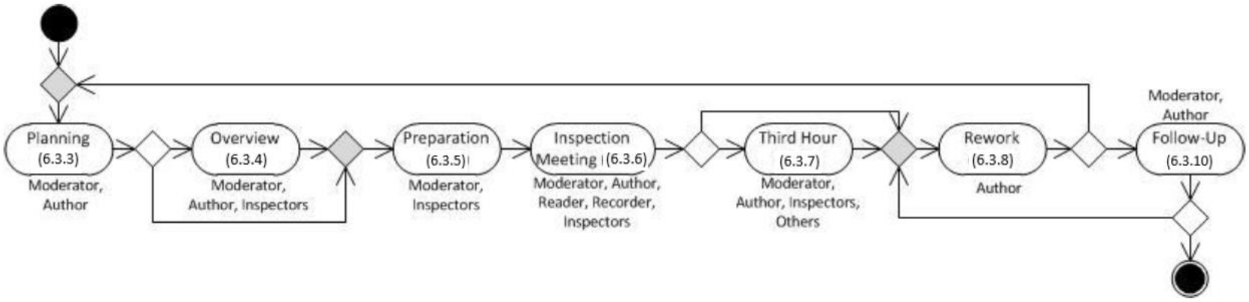

Figure 1. Inspection Process

6.3.3 Planning

Planning is the stage in which the package contents, required support, and scheduling of an inspection are defined. The following activities are to be completed during the Planning stage.

6.3.3.1 Entry Criteria Check

6.3.3.1.1 The moderator shall ensure that the entry criteria, as specified in Section 6.2, “Entry and Exit Criteria”, have been met. [SFI-013]

6.3.3.1.2 If the moderator determines that the product does not meet the entry criteria, the product shall be returned to the author for further development. [SFI-014]

6.3.3.2 Inspection Package Contents

6.3.3.2.1 The moderator should choose the number of product elements to be inspected at any given inspection so as to allow the corresponding inspection meeting to cover all of the material in approximately two hours or less and at a rate of inspection less than or equal to the maximum rate allowed for this type of inspection, as recommended in Section 7, ”Types of Inspections".

Note: The inspection rate should be calculated only on the technical content of the work product to be inspected, and should exclude parts of the work products such as title pages and table of content.

Rationale: Studies have shown that at higher page rates, the detection effectiveness of inspection teams is dramatically reduced. (This effect has been demonstrated at NASA based on analyses of data from across multiple Centers, both historically (see Kelly 1992) and on contemporary projects (see Shull 2012).) Giving teams too many pages to comprehend in too little time ensures that team members cannot thoroughly understand the technical details. It ensures that preparation cannot be completed in adequate detail and thus also diminishes the utility of the inspection meeting.

6.3.3.2.2 The moderator should put together an inspection package in a form (e.g., electronic, physical, etc.) deemed suitable by the project, and containing:

a. An inspection announcement that includes:

(1) A statement of objectives for the inspection;

(2) The inspection schedule.

b. Physical or electronic inspection reporting forms.

c. Inspection work aids applicable to the product being inspected and its quality needs.

d. The documents that are input to the Planning stage.

6.3.3.2.3 The inspection team should inspect only the versions of materials found in the inspection package. Any further changes that occur to the work product cannot be brought into the inspection, unless it is negotiated with the moderator as a needed change prior to the inspection.

6.3.3.3 Inspectors

6.3.3.3.1 Based on the contents of the inspection package, the moderator identifies the inspectors for each inspection as required in Section 5.3, “Candidates for Inspectors” and notifies them of their roles and responsibility to support the inspection.

6.3.3.4 Inspection Scheduling

6.3.3.4.1 The moderator should schedule the inspection meetings far enough in the future as required in Section 6.4, “Inspection Procedure Customization,” to allow each inspector to have sufficient preparation time.

6.3.3.5 Distribution

6.3.3.5.1 During this step, the moderator should deliver the inspection package to inspectors.

6.3.4 Overview

6.3.4.1 The overview is an educational briefing, either oral or written, provided at the time the inspection product and related materials are distributed, and prior to the inspection meeting, which explains the distributed materials at a high level. The purpose of the overview is to bring all of the inspectors to the point where they can read and analyze the inspection product and related materials. This may include a refresher on Formal Inspection or domain-specific topics.

6.3.4.2 The moderator determines, during the Planning stage, whether the overview is needed for the inspection, and whether it should be presented as a formal meeting. The moderator shall ensure that all inspectors receive and understand the content of the inspection package. [SFI-015]

6.3.5 Preparation

6.3.5.1 Preparation is the stage when inspectors individually review and comment on the material under review in preparation for the inspection meeting. The inspectors are also expected to review any supporting material and provide comments to the moderator prior to the actual Inspection Meeting. The author's participation in the preparation stage is optional. As discussed in Section 6.3.3.2, inspections should only be performed on the work products that were originally distributed to the team at the beginning of this stage. That is, updates to the software products after these have been submitted for inspection will not be reviewed, unless the moderator decides to distribute them and restart the preparation process.

6.3.5.2 During this stage, inspectors will focus on detecting defects and developing questions by utilizing the inspection work aids, and examining the inspected product for technical accuracy, fulfillment of requirements, and adherence to standards and conventions. The moderator must give sufficient time for preparation by the inspectors and for their own review of the incoming defects. The preparation will depend largely on the work product itself and the current work load of the inspectors. However, more than a week for preparation may mean the product and supporting material is too large or complex and may result in setbacks to the work flow. The moderator must work with the project manager to assure that adequate time is allowed for inspectors to fully review the work product and make adjustments as needed.

6.3.5.3 During this stage, inspectors who have been assigned as the reader and recorder for the inspection meeting prepare for their roles. The reader decides how to present the work product during the Inspection Meeting. The recorder reviews the checklist and defect type taxonomy to assist in defect recording and classification.

6.3.5.4 Inspectors shall examine the work product and document possible defects and questions, and deliver those inputs to the moderator at least one work day prior to the start of the inspection meeting. [SFI-016]

6.3.5.5 The moderator shall review and organize the inspectors' defects, questions, and preparation efforts in order to organize the inspection meeting and to decide whether and how to proceed with the inspection meeting. [SFI-017] If the moderator does not believe that the inspection team is adequately prepared, then the moderator should consider rescheduling the inspection meeting, or excluding the unprepared inspectors. The moderator should exclude inspectors only after weighing the tradeoffs of conducting the meeting without key technical experts being present: This can substantially affect the completeness and the efficiency of the meeting.

6.3.6 Inspection Meeting

6.3.6.1 The inspection meeting, which is conducted and controlled by the moderator, is a formal meeting at which inspectors examine the inspected product as a group.

Rationale: The synergy gained from sharing comments out loud often allows the group to not only come up with even more thoughts and possible defects within the work product, but also engenders greater understanding of the software and the system being built. Inspection meetings enable the members of the development team to learn about other skills and areas of expertise outside of their usual experience.

6.3.6.2 The reader shall lead the inspectors through the inspected product and related materials. [SFI-018] The inspectors should only discuss technical defects. Clerical defects should only be discussed if they prevent an understanding of technical aspects of the work products being inspected. Inspectors should focus their discussion on identifying the problems to be fixed in the work product. Generally, the discussion of what the fixes should look like is to be avoided, except in so far as it helps to understand the technical issues. The moderator should determine whether additional discussion is to be taken off line, such as to the 3rd hour.

Rationale: The time of inspection participants is a precious resource, and care must be taken to use the meeting time in the most cost-effective way. Discussion of clerical defects often takes significant time, far out of proportion to their impact on system quality, and is to be avoided. Discussion of solutions can be extremely time-consuming, and should only be done up to the point where the author is clear on what should be fixed. It is the author's job to fix the problem outside the inspection meeting. If the defects reflect a possible extensive rewrite or redesign, then either the author or the moderator may take the product and defects to the project manager. Outside the inspection meeting, the author may request other inspectors for suggestions and explore best solutions.

6.3.6.3 The recorder shall record all defects identified during the inspection meeting. [SFI-019]

6.3.6.4 The inspectors shall classify the defects according to severity and type as described in Section 6.3.6.7, "Defect Classification." [SFI-020] The inspection team should discuss the classification of all defects with the determination being made by the moderator.

6.3.6.5 Inspection conclusion

6.3.6.5.1 At the conclusion of each inspection meeting, the moderator shall decide, based on the criteria established in accordance with Section 6.4, "Inspection Procedure Customization," whether a re-inspection will be performed. [SFI-021]

6.3.6.5.2 The moderator should compile the following artifacts as output of the inspection meeting:

a. From the recorder, a list of classified anomalies or defects.

b. From the recorder, a list of open issues including potential discrepancy reports and a list of defects found in supporting work products that may have been put under configuration management.

c. From individual inspectors, the inspected work product marked with the clerical defects provided as part of their inputs to the moderator prior to the inspection meeting.

d. A list of inspection participants with checklists (if required) from each.

6.3.6.5.3 The moderator shall ensure that the author(s) receive the list of classified anomalies or defects and the work product marked with any clerical defects. [SFI-022]

6.3.6.6 Inspection Continuation – Additional Meetings

6.3.6.6.1 The moderator should control the inspection meeting with a goal of completing within approximately 2 hours or stopping the meeting at that point and scheduling a continuation for a later date.

Rationale: It has been found that teams are much less effective at finding defects after 2 hours of meeting time. If the moderator determines that an inspection meeting is likely to last substantially longer than 2 hours, the moderator should find a logical stopping point and reschedule the rest of the meeting for a later time.

6.3.6.7 Defect Classification

6.3.6.7.1 The following describes the severity and defect type classification to be used when recording defect information:

a. Severity of Defect: Each defect in the inspected product is classified according to its severity as one of the following:

(1) Major Defect: A defect in the product under inspection which, if not corrected, would either cause a malfunction which prevents the attainment of a primary mission objective or system safety, or would result in a significant budget or schedule impact.

(2) Minor Defect: A defect in the product under inspection which, if not fixed, would not prevent the attainment of a primary mission objective or system safety,

or would not result in a significant budget or schedule impact, but could result in difficulties in terms of operations, maintenance, and future development.

(3) Clerical Defect: A defect in the product under inspection at the level of editorial errors, such as spelling, punctuation, and grammar.

b. Types of Defects: Defects are further classified according to a pre-defined defect taxonomy. This defect taxonomy would be defined as part of developing the inspection procedure. Headings on the checklists used for the inspection can be used to derive the defect taxonomy. The following is an example of error taxonomy for code-related defects:

(1) Algorithm or method: An error in the sequence or set of steps used to solve a particular problem or computation, including mistakes in computations, incorrect implementation of algorithms, or calls to an inappropriate function for the algorithm being implemented.

(2) Assignment or initialization: A variable or data item that is assigned a value incorrectly or is not initialized properly or where the initialization scenario is mishandled (e.g., incorrect publish or subscribe, incorrect opening of file, etc.)

(3) Checking: Software contains inadequate checking for potential error conditions, or an inappropriate response is specified for error conditions.

(4) Data: Error in specifying or manipulating data items, incorrectly defined data structure, pointer or memory allocation errors, or incorrect type conversions.

(5) External interface: Errors in the user interface (including usability problems) or the interfaces with other systems.

(6) Internal interface: Errors in the interfaces between system components, including mismatched calling sequences and incorrect opening, reading, writing or closing of files and databases.

(7) Logic: Incorrect logical conditions on if, case or loop blocks, including incorrect boundary conditions ("off by one" errors are an example) being applied, or incorrect expression (e.g., incorrect use of parentheses in a mathematical expression).

(8) Non-functional defects: Includes non-compliance with standards, failure to meet non-functional requirements such as portability and performance constraints, and lack of clarity of the design or code to the reader - both in the comments and the code itself.

(9) Timing or optimization: Errors that will cause timing (e.g., potential race conditions) or performance problems (e.g., unnecessarily slow implementation of an algorithm).

(10) Coding Standard Violation: when reviewing code, the coding standards need to be reviewed and verified that the code meets them.

(11) Other: Anything that does not fit any of the above categories that is logged during an inspection of a design artifact or source code.

6.3.6.8 Inspection Meeting Decorum

6.3.6.8.1 The moderator should maintain the decorum and focus of the inspection meeting such that the inspection meeting remains a professional environment during which common courtesy and personal regard govern the identification of needed changes.

6.3.6.8.2 The inspection should not be refocused from its stated objective and format to accommodate inexperienced staff attending for training value. As important as it is for new inspectors to experience real inspections, they should remain as observers and should not disrupt the flow of the inspection, and should refrain from asking questions until after the formal inspection is completed and ended by the moderator.

6.3.7 Third Hour

6.3.7.1 The third hour is an optional and informal meeting or activity that is separate from the inspection meeting. During the third hour, resolutions to open issues recorded in the inspection meeting may be obtained, and solutions for defects identified during the inspection may be discussed.

6.3.7.2 The author shall determine if a third hour is needed. [SFI-023]

6.3.7.3 Participants at the third hour may be any subset of the inspection meeting inspectors plus any additional persons whose expertise would help resolve open issues or find solutions to the defects identified during the inspection meeting.

6.3.8 Rework

The author shall correct all defects identified as major defects. [SFI-024] The author should correct all other non-major defects at the moderator’s discretion when it is cost- and schedule-efficient to do so. The author should update the anomaly or defect list with the defect resolutions. The author should complete the rework in a timely manner.

6.3.9 Re-inspection

Re-inspection is a repetition of the inspection process for a complete or partial set of products that have been previously inspected. The need for re-inspection is determined by the moderator by evaluating the outcomes of inspections against the criteria previously defined by the project manager. The inspection team should generate a separate inspection report for each re-inspection.

6.3.10 Follow-Up

6.3.10.1 The moderator, with the author's assistance, should verify, personally or by delegation, that all major defects and other non-major defects dispositioned as to be fixed have been corrected, and that no additional defects have been introduced.

6.3.10.2 The moderator shall generate and report all the required data for the inspection report, as defined in Section 8.1, "Product and Process Measures." [SFI-025]

6.3.10.3 The moderator shall ensure that the exit criteria have been met and the work product returned to the author for rework if the criteria are not met. [SFI-026]

6.4 Formal Inspection Procedure Customization

6.4.1 Customizing an inspection procedure is necessary for an inspection to be effective. Inspection customization can occur at the Center level and at the project level.

6.4.2 At the project level, for each project, the project manager should define:

a. The required minimum lead time between distribution of the product to be inspected and its related materials, and the occurrence of the corresponding inspection meeting. This time should be long enough to allow adequate preparation by the inspectors.

b. The criteria by which a decision will be made at the end of each inspection meeting whether to re-inspect all or part of the just inspected products.

c. A method to document, track, resolve, and measure open issues identified during inspections which are not classified as defects.

6.4.3 For each inspection type, depending on the type of work products for which the inspection is conducted, the following inspection aspects can be varied:

a. The required contents of the inspection package in terms of:

(1) Products and documentation to be inspected.

(2) Reference materials.

(3) Inspection meeting notice and scheduling information.

(4) Inspection work aids, including checklist content.

b. The mandatory number of inspectors who will participate in an inspection. The recommended size for an inspection team is between 3 and 6 (or at most 10) inspectors.

c. The maximum rate at which the inspection to be performed (in terms of pages or lines per hour). This rate should be based on available data and project experience.

6.4.4 How the inspection process may be customized depends on the following factors:

a. The type of work product (plans, requirement specifications, designs, code, test plans, test procedures). Refer to Sections 7.2-7.9 for customization considerations when inspecting these work products.

b. The degree of in-house development versus acquisition. Refer to Section 7.11 for customization considerations for performing inspections in acquisition projects or when COTS software products are used.

c. The importance of various assurance considerations (e.g., reliability, safety, quality, security, performance). Refer to Section 7.12 for assurance-based customization considerations.

d. The team and project size.

7. TYPES OF INSPECTIONS

7.1 General

7.1.1 This section describes the generally recognized types of inspections based on the products which can be inspected. Not all possible software products are represented here, however, additional types of inspections may be conducted using the inspection process defined within this Standard. In the following sections specific considerations as to the inspection checklist, additional input, needed perspectives, and additional planning considerations (e.g., inspection rate, team size) specific for each inspection type are described. See Appendix A for a table of Inspection types and participation recommendations.

7.1.2 NPR 7150.2 requires peer reviews and inspections to be performed on many of the key software products, such as requirements, software plans, test procedures, and selected portions of the designs, and code for projects developing the following software:

a. All safety critical software

b. Software with classifications of A, B, or C

c. Some business and IT software (See NPR 7150.2)

Note: In addition to the list above, NPR 7150.2 also recommends inspections for any software components that are safety or mission-success related. In the selection of portions of the design or code identified for inspections, a prime consideration is the complexity of the component. Inspections are also a best practice for identifying security vulnerabilities.

7.1.3 It is up to the program/project/facility manager to determine the type of inspections or peer reviews to conduct and on which software products. Software Formal Inspections are recommended for the most critical products and should be performed early, on models, requirements, and preliminary design products, to provide the greatest return on finding and correcting errors and defects before they are propagated into detailed design, code and operations.

7.2 Software Plan Inspections

7.2.1 NPR 7150.2 requires inspections or peer reviews to be performed on the software plans. Typical plans that might be inspected are software development or management plan(s), software configuration management plan, software maintenance plan, software assurance plan, and software safety plan. This applies to the same groups of software identified in 7.1.2.

7.2.2 Checklists for software plan inspections should contain items which:

a. Address that the effort, schedule, and cost estimates for each activity (e.g., development, configuration management, maintenance, assurance, safety, security) are reasonable.

b. Address the allocations of appropriate resources, including tools and personnel with the needed skills and knowledge.

c. Check that all risks have been identified and documented along with their probability of occurrence, impact, and mitigation strategy.

d. Assure sufficient management of the produced project data.

e. Check that an appropriate and feasible plan exists for sustainment and retirement of the software, if applicable.

f. Check that the plan fulfills the corresponding NPR 7150.2 recommended contents, found in NASA-HDBK-2203A, NASA Software Engineering Handbook, located at http://swehb.nasa.gov/ . The Software Engineering Handbook "Software Topics" section contains a topic called "Documentation Guidance" that has recommended contents for many software plans, including software development or management plan(s), software configuration management plan, software maintenance plan, software assurance plan, and software safety plan).

7.2.3 The inspection should include participants representing the following stakeholders:

a. Software engineers

b. Systems Engineers

c. Software Assurance personnel

d. Safety engineers (when appropriate)

e. Software Safety personnel (when appropriate)

f. Configuration management personnel (when inspecting configuration management plan)

Note: When necessary, other experts may be included as participants to provide important technical expertise. This can include the project management for both the *software and the system, although in this case, please review the caveats in Section 5.2, "The Inspection Team"*

7.2.4 The moderator should limit the amount of work product to be inspected in order to maintain an acceptable inspection rate. Prior data and experiences suggest a starting metric for this type of inspection of at most 15 pages per hour.

7.3 System Requirements Inspections (R0)

7.3.1 Checklists for system requirement inspections should contain items which:

a. Describe proper allocation of functions to software, firmware, hardware, and operations.

b. Address the validation of all external usage interfaces.

c. Check that all the software system functions are identified and broken into configuration items, and that the boundary between components is well-defined.

d. Check that all configuration items within the software system are identified.

e. Check that the identified configuration items provide all functions required of them.

f. Check that all interfaces between configuration items within the software system are identified.

g. Address the correctness of the software system structure.

h. Check that all quantifiable requirements and requirement attributes have been specified.

i. Address the verifiability of the requirements.

j. Check for the traceability of requirements from mission needs (e.g., use cases, etc.).

k. Check for the traceability of requirements from system safety and reliability analyses (e.g., Preliminary Hazard Analysis (PHA), Fault Tree Analysis (FTA), Failure Modes and Effects Analysis (FMEA), hazard reports, etc.).

7.3.2 In addition to the system definition and requirements document, inspection of system requirements should include the following materials or links to those materials as reference documents:

a. Any existing interface definition or specification document.

b. Verification and Validation plan, if available.

c. For software which has been determined safety critical, the system safety and reliability analyses (e.g., PHA, FTA, FMEA, hazard reports, etc.).

7.3.3 The inspection should include participants representing the following stakeholders:

a. Persons who participate in defining requirements of the system to be built (e.g., system requirement engineer, users)

b. Persons who are responsible for integrating the sub-systems.

c. Persons who are responsible for implementing the requirements allocated to the various sub-systems.

d. Persons who understand the quality needs of the system (e.g., quality assurance, software assurance, and reliability).

e. Persons who are responsible for implementing Software Safety (if the inspection involves a safety critical system).

7.4 Software Requirements Inspections (R1)

7.4.1 Checklists for software requirement inspections should contain items which:

a. Check that the specification of each of the following is complete and accurate:

(1) Software functions.

(2) Input and output parameters.

(3) States and modes.

(4) Timing and sizing requirements for performance.

(5) Interfaces.

(6) Use Cases if available.

b. Check that specifications are included for error detection and recovery, reliability, maintainability, performance, safety, and accuracy.

c. Check that safety requirements are identified as such.

d. Check that safety-critical modes and states, and any safety-related constraints, are identified.

e. Address the traceability of requirements from higher level documents.

f. Check that the requirements provide a sufficient base for the software design.

g. Check that the requirements are measurable, consistent, complete, clear, concise, and testable.

h. Check that the content of the software requirement specification fulfills the NPR 7150.2 recommendations, found in NASA-HDBK-2203A, NASA Software Engineering Handbook.

7.4.2 Documented requirements are not always used. Models can be created along with use cases and scenarios. Some models are fed into autocode generators which in turn create code and test scripts; other models are used to generate architectural models and designs. All these can also be inspected. For agile development, the stories, product backlog, and even each sprint's requirements set or stories could undergo inspections to assure the customer, developers, the product owners, experts, and scrum masters are addressing the right needs overall and for each sprint. The above requirements checklist can be adapted and applied to assure the models, stories and product backlogs have covered these essentials.

7.4.3 The inspection should include the following materials as reference documents:

a. Work products documenting the system requirement specifications.

b. Specification document(s) or system models of the interface(s) with which the software will be interacting.

c. For software which has been determined safety critical, the system safety and reliability analyses (e.g., PHA, FTA, FMEA, hazard reports, etc.).

7.4.4 The inspection should include participants representing the following stakeholders:

a. Persons who participate in the allocation of the system requirements to the sub-systems (e.g., system requirement engineer).

b. Persons who are responsible for defining the software component's requirements.

c. Persons who are responsible for defining the software's requirements.

d. Persons who understand the quality needs of the software component (e.g., quality assurance, software assurance, and, reliability).

e. Persons who are responsible for designing tests for the software (e.g., Test engineers)

f. Persons who are responsible for implementing Software Safety (if the inspection involves a safety critical system).

7.4.5 The moderator should limit the amount of work product to be inspected in order to maintain an acceptable inspection rate. Prior data and experiences suggest a starting metric for this type of inspection of at most 15 pages per hour.

7.5 Architectural Design Inspection (I0)

7.5.1 Checklists for architectural (preliminary) design inspections should contain items which:

a. Check that the design meets approved requirements.

b. Address the validation of all interfaces among modules within each component.

c. Address the completeness of the list of modules and the general function(s) of each module.

d. Address the validation of fault detection, identification, and recovery requirements.

e. Check that the component structure meets the requirements.

f. Address the validation of the selection of reusable components.

g. Address the traceability of the design to the approved requirements.

h. Address the validation of the input and output interfaces.

i. Check that each design decision is a good match to the system's goal.

j. Check that the content of the design description fulfills the NPR 7150.2 recommendation, found in NASA-HDBK-2203A, NASA Software Engineering Handbook.

k. Check that safety controls and mitigations are clearly identified in the design document, when a safety critical system is under inspection (Review system safety analyses in supporting documentation).

l. When inspecting object-oriented or other design models:

(1) Check that the notations used in the diagram comply with the agreed-upon model standard notation (e.g., UML notations).

(2) Check that the design is modular.

(3) Check that the cohesion and coupling of the models are appropriate.

(4) Check that architectural styles and design patterns are used where possible. If design patterns are applied, validate that the selected design pattern is suitable.

(5) Check the output of any self or external static analysis tool outputs.

7.5.2 The inspection should include participants representing the following stakeholders:

a. Persons responsible for defining requirements for the software components.

b. Persons who are responsible for detailed design of the software.

c. Persons who understand the quality needs of the software component (e.g., quality assurance, software assurance, and, reliability).

d. Persons who are responsible for verifying and validating systems interfaces (e.g., system test engineers).

e. Persons who are responsible for implementing Software Safety (if the inspection involves a safety critical system).

7.5.3 The inspection should include the following materials as reference documents:

a. Work products documenting the software requirement specifications.

b. Specification document(s) of the interface(s) with which the software will be interacting.

c. For safety critical software, the system hazard report and any software portions therein.

d. Any software reliability analyses available along with their critical items list and software design recommendations for meeting fault tolerance.

7.5.4 The moderator should limit the amount of work product to be inspected in order to maintain an acceptable inspection rate. Prior data and experiences suggest a starting metric for this type of inspection of at most 20 pages per hour.

7.6 Detailed Design Inspection (I1)

7.6.1 Checklists for detailed design inspections should contain items which:

a. Check that the design meets the approved requirements.

b. Address the validation of the choice of data structures, logic algorithms (when specified), and relationships among modules.

c. Check that the detailed design is complete for each module.

d. Address the traceability of the design to the approved requirements.

e. Check that the detailed design meets the requirements and is traceable to the architectural software system design.

f. Check that the detailed design is testable.

g. Check that design can be successfully implemented within the constraints of the selected architecture.

h. Check output from any static analysis tools available.

7.6.2 The inspection should include participants representing the following stakeholders:

a. Persons who are responsible for defining the software component’s requirements.

b. Persons who are responsible for defining the software’s requirements.

c. Persons who are responsible for implementing the software’s requirements.

d. Persons who understand the quality needs of the software component (e.g., quality assurance, software assurance, safety, and, reliability).

e. Persons who are responsible for implementing Software Safety (if the inspection involves a safety critical system).

f. Persons who are responsible for designing tests for the software (e.g., Test Engineers).

7.6.3 The inspection should include the following materials as reference documents:

a. Work products documenting the software architectural and preliminary designs.

b. For safety critical software, the system hazard report and any software portions therein.

c. Any software reliability analyses available along with their critical items list and software design recommendations for meeting fault tolerance.

7.6.4 The moderator should limit the amount of work product to be inspected in order to maintain an acceptable inspection rate. Prior data and experiences suggest a starting metric for this type of inspection of at most 20 pages per hour.

7.7 Source Code Inspections (I2)

7.7.1 Checklists for source code inspections should contain items which:

a. Address the technical accuracy and completeness of the code with respect to the requirements.

b. Check that the code implements the detailed design.

c. Check that all required standards (including coding standards) are satisfied.

d. Check that latent errors are not present in the code, including errors such as index out-of-range errors, buffer overflow errors, or divide-by-zero errors.

e. Address the traceability of the code to the approved requirements.

f. Address the traceability of the code to the detailed design.

g. When static or dynamic code analysis is available, check the results of these tools.

7.7.2 Since the correctness of automatically generated code may be difficult to verify, it is recommended that such code should be subject to inspection, especially if the code is part of a safety-critical system. In addition to addressing the above checklist items, inspections of automatically generated code should also:

a. Address the correctness of the model used to generate the code.

b. Check that the code generator is correctly configured for the target environment.

c. Check that the interface between the generated code and the rest of the code base is consistent and correct.

d. Check that any known problems with the code generator are avoided or mitigated.

e. Check that any known issues or existing problems with the code generator are documented.

7.7.3 The inspection should include participants representing the following stakeholders:

a. Persons who are responsible for defining the software's requirements.

b. Persons who are responsible for implementing the software's requirements.

c. Persons who are responsible for designing tests for the software (e.g., Test engineers)

d. Persons who understand the quality needs of the software component (e.g., quality assurance, software assurance, and, reliability), when appropriate.

e. Persons who are responsible for implementing Software Safety (if the inspection involves a safety critical system).

7.7.4 The moderator should limit the amount of work product to be inspected in order to maintain an acceptable inspection rate. Prior data and experiences suggest a starting metric for this type of inspection of at most 10 pages per hour.

7.7.5 The inspection should include the following materials as reference documents:

a. Work products documenting the detailed software design.

b. Output from static analysis of the code, if available.

7.8 Test Plan Inspection (IT1)

7.8.1 Checklists for test plan inspections should contain items which:

a. Check that the purpose and objectives of testing are identified in the test plan and they contribute to the satisfaction of the mission objectives.

b. Check that all new and modified software functions will be verified to operate correctly within the intended environment and according to approved requirements.

c. Check that the resources and environments needed to verify software functions and requirements correctly are identified.

d. Check that all new and modified interfaces will be verified.

e. Address the identification and elimination of extraneous or obsolete test plans.

f. Check that each requirement will be tested.

g. Check that tester has determined the expected results before executing the test(s).

h. For safety critical software systems:

(1) Check that all software safety critical functions or hazard controls and mitigations will be tested. This testing should include ensuring that the system will enter a safe state when unexpected anomalies occur.

(2) Check that safety and reliability analyses have been used to determine which failures and failure combinations to test for.

i. Check that the content of the test plan fulfills NPR 7150.2 recommendations, found in NASA-HDBK-2203A, NASA Software Engineering Handbook.

7.8.2 The inspection should include participants representing the following stakeholders:

a. Persons who are responsible for defining the software component's requirements

b. Persons who are responsible for implementing the software's requirements.

c. Persons who understand the quality needs of the software component (e.g., quality assurance, software assurance, and, reliability).

d. Persons who are responsible for designing tests for the software (e.g., Test engineers)

e. Persons who are responsible for implementing Software Safety (if the inspection involves a safety critical system).

f. Persons who are responsible for verifying and validating systems interface (e.g., system test engineers).

7.8.3 The inspection should include the following materials as reference documents:

a. Work products documenting the software requirement specifications.

b. Specification document(s) of the interface(s) with which the software will be interacting.

c. For safety critical software, the system hazard report and any software portions therein.

7.8.4 The moderator should limit the amount of work product to be inspected in order to maintain an acceptable inspection rate. Prior data and experiences suggest a starting metric for this type of inspection of at most 20 pages per hour.

7.9 Test Procedure Inspection (IT2)

7.9.1 Checklists for test procedure inspections should contain items which:

a. Check that the set of test procedures meets the objective of the test plan.

b. Check that each test procedure provides:

(1) A complete and accurate description of its purpose

(2) A description of how it executes

(3) All expected results.

c. Check that each test procedure identifies which requirement(s) it is testing and correctly tests the listed requirement(s).

d. Check that each test procedure identifies the required hardware and software configurations.

e. Check that test procedures exist to verify the correctness of the safety critical controls as well as any software controls or mitigations of hazards (HW, SW or CPLD) and that the system can obtain a safe state from different modes, states and conditions.

f. Check that each test procedure will objectively verify the implementation of the requirement with expected outcome.

g. Check that the content of the software test procedure fulfills NPR 7150.2 recommendations, found in NASA-HDBK-2203A, NASA Software Engineering Handbook.

7.9.2 Automatically generated test procedures and scripts (especially for safety-critical software) should also be subject to inspection. In addition to verifying the above checklist items, inspections of auto-generated test procedures should also:

a. Address the correctness of the model used to generate the test procedure.

b. Check that the test generator is correctly configured for the target environment.

c. Check that any known or existing problems with the test generator are avoided or mitigated.

d. Check that any known issues or existing problems with the test generator are documented.

7.9.3 The inspection should include participants representing the following stakeholders:

a. Persons who are responsible for implementing the software’s requirements.

b. Persons who understand the quality needs of the software component (e.g., quality assurance, software assurance, and, reliability).

c. Persons who are responsible for designing tests for the software (e.g., Test engineers)

d. Persons who are responsible for implementing Software Safety (if the inspection involves a safety critical system).

e. Persons who are responsible for verifying and validating systems interface (e.g., system test engineers).

7.9.4 The inspection should include the following materials as reference documents:

a. Work products documenting the software requirement specifications.

b. Work products documenting the software test plan.

c. For software which has been determined safety critical, the system hazard report and any software portions therein with attention to the verifications specified.

d. Any software reliability analyses available along with their critical items list and software design recommendations for meeting fault tolerance.

7.9.5 The moderator should limit the amount of work product to be inspected in order to maintain an acceptable inspection rate. Prior data and experiences suggest a starting metric for this type of inspection of at most 20 pages per hour.

7.10 Considerations on Inspections of Software that Runs on Programmable Logic Devices Operating Systems

7.10.1 Many projects develop complex programmable logic devices with operating systems running on them. While the Programmable Logic Devices (PLD) themselves are not considered software by NASA’s community of practice, the software that runs on an operating system built into one is. There are a few additional considerations for this software as opposed to software which runs on other computing devices. The inspections may be performed during the following development activities:

a. Requirements development to verify that the requirements for the software are compatible with functions ascribed to that device and traceable to higher level requirements.

b. Software design development to review the software meets any device specific functionality and structure (e.g. initiation and restart timing considerations, error handling).

c. Implementation, to review that the software and the various simulations and analyses are compatible and will work with the PLD and its Operating System.

7.10.2 Testing, to review the software test plans and procedures are complete and assure the test strategy for pre and post burn in as applicable. The inspection of software to run on the operating systems of programmable logic devices should include participants who cover the expertise needed by the following roles:

a. Persons who define the system requirements from which the requirements were derived (e.g., system engineers)

b. Persons who have expertise with the electronics hardware with which software will interface (e.g., electronics designer)

c. Persons who are responsible for defining and developing programmable logic devices and their operating systems (e.g., Programmable Logic Device Specialist)

d. Persons who have other key technical expertise that is required for this work product to be evaluated, such as safety engineers, quality assurance personnel, configuration management staff, reliability engineers, and performance engineers.

7.11 Considerations for Acquisition Projects

7.11.1 As many of NASA's projects are done via a contract, agreement, or task order it is important to consider putting this standard on the contract, agreement, or task order, or allowing an industry standard equivalent if the contractors' propose one that meets the requirements of this standard.

7.11.2 It is highly recommended that each project ensure that inspections are part of the software provider's required software assurance activities and that they are performed in compliance to this Standard. Specifically, it should be required that all data related to the inspection process and outcome of the process is either regularly reported or readily accessible for use by the project manager, to periodically evaluate the projects' progress in compliance with this Standard. The reporting of inspection results can occur as part of a standard project status briefing to NASA, and or as a report during major milestone reviews. In addition, formal inspection records and results need to be available for audits and surveys of the project or Center. The various methods, deliverables and schedule for reporting SW Formal Inspection results needs to be documented in both the NASA and contractor project planning documents.

7.11.3 Several considerations are important for assuring proper inspection usage by software provider(s):

a. Not all software provider(s) may be able or willing to provide all of the data specified in Section 8.1, "Product and Process Measures" In such cases, a subset of inspection data should be defined that still provide sufficient oversight of the project. Examples of such data may include: The number of inspections planned versus accomplished; the number of components and modules that have undergone inspection; the number of change requests and problem reports found as a result of inspections.

Note: NPR 7150.2 requires projects to provide software management and technical metrics, based on their goals which should include obtaining data that provides insight into software functionality, software progress tracking, software quality, and process improvement opportunities. In the context of an acquisition project, such metrics need to be defined upfront and included in the RFP and the contract.

b. The data to be furnished by software provider(s) must be specified in the contract.

c. To reduce the cost of reporting and analyzing data, consider using a tiered data reporting scheme. A subset of the data of interest should be reported periodically by the provider, however, the remaining detailed data should be available upon request for more detailed analysis.

d. It must be clear and agreed upon ahead of time whether or not software providers can define their own defect taxonomies. If providers may use their own taxonomy, request the software providers to furnish the definition or the data dictionary of the taxonomy. It is also important (especially when the provider team contains subcontractors) to ensure that consistent definitions are used for: defect types; defect severity levels; inspection effort reporting (how comprehensive or restrictive are the activities that are part of the actual inspection).

7.11.4 Also in acquisition contexts, special considerations should be taken for the composition of the inspection team. It is highly recommended for each project to:

a. Include a representative (with relevant expertise) from each software provider during the inspections of the system requirements.

b. Assign an acquirer representative (with relevant expertise) to participate in the software provider’s inspections of the software requirements.

7.11.5 If a project is incorporating COTS or reused software products into the system under development, additional checklist items may be included at the appropriate inspections to:

a. Ensure that the COTS or reused software products meet any conditions required by the NPR 7150.2.

b. Verify the existence of trade studies prior to the selection of COTS or reuse products.

c. Verify the sufficiency of the trade studies, specifically by assessing the evaluation criteria used in the studies, such as:

(1) The degree of requirements that are met by candidate COTS or reused products.

(2) Constraints regarding the applicability of the candidate COTS or reused products are documented, and meet the needs of the system.

(3) Risks due to remaining defects or workarounds are acceptable.

d. Verify that the selected COTS or reuse products meet the requirements that were allocated to them.

e. Verify the quality of the code that integrates the COTS or reuse products with other parts of the system.

7.12 Other quality considerations

The project manager should identify and prioritize the non-functional needs of the delivered software, such as performance, safety, and reliability, and customize the inspection process so to meet these quality needs. The customization can be done through:

a. Customizing the inspection checklist to include statements that specifically address the achievement of quality requirements.

b. Customizing the inspection team composition by including experts who can assure the quality needs can be and are met.

8. PROCESS EVALUATION

While this Standard ascribes the responsibility of process improvement to the project manager, these requirements may be handled instead at a program or Center Level.

8.1 Product and Process Measures

8.1.1 Table 1 lists the data to be collected as part of the inspection process.

Note: Some of the data needs to only be collected once for an entire project, some of the data will be normal information needed to configuration manage each inspection. In addition, part of the formal inspection process is to provide metrics for determining effectiveness and where improvement may be needed in the inspection process itself as well as in the products being inspected. It is recommended that one or more standardized inspection data sheets or templates be developed and used for a project, program or Center organization.

8.1.2 The project manager shall define in the inspection procedure where and in which format the data is to be stored. [SFI-027]

Table 1. Data Collected at Inspection Points

Measurement Name	Data to collect	When to Collect	Who to Collect
Description of organization	• Project name at contract level • Manager responsible for product • System (only for functional project) • Organization producing product • Application (to be customized) • Sub-application (to be customized)	At the beginning of the planning stage	Project Manager

Measurement Name	Data to collect	When to Collect	Who to Collect
Description of inspected product	• Inspection type • Product element names • Product element versions • Size of product elements • Targeted delivery/release identification • Change authorization document(s)	At the end of the planning stage	Moderator
Description of the inspection process	• Inspection time and date • First or re-inspection • If re-inspection, prior inspection date • Overview date • Names of inspectors, excluding author • Participant's area of expertise • Roles of inspectors • Inspection meeting duration • Overview meeting duration • Outcome (pass or re-inspection) • If re-inspection is determined as needed, the target date • Inspection close date	At the end of the inspection meeting stage	Moderator
Inspection efforts	• Planning time for author and moderator • Preparation time for each inspector • Third Hour time for each inspector • Rework time • Follow-up time	At the end of each inspection stage	Moderator With input from inspectors and author Should correlate with amount of material reviewed
Description of defects	• For each defect: o its disposition o its defect type (when determined) o whether major or minor	At the end of inspection meeting	Moderator
Defect metrics	• Number of major defects found • Number of minor defects found	At the end of the Inspection meeting	Moderator

Measurement Name	Data to collect	When to Collect	Who to Collect
	• Number of major defects corrected. • Number of minor defects corrected. • Authorized deviations list (list of all inspection defects accepted)	At the end of the follow-up	

8.2 Evaluation Analyses

8.2.1 The project manager, with assistance from the moderator(s), should periodically perform the following set of trend analyses so as to identify positive or adverse trends in the inspection process at the earliest possible opportunity using the data collected in Section 8.1, "Product and Process Measures":

a. Number of inspections complete versus planned.

b. Total defects (for major and minor) sorted by:

(1) Delivery or release ID

(2) Inspection type

c. Defect density of products (number of major and minor defects per line or page) sorted by:

(1) The defect type

(2) The inspection type

(3) The application type

(4) Development organization.

d. Inspection labor hours (overview, planning, preparation, inspection, third hour, follow-up, and rework) versus number of:

(1) Defects found

(2) Lines and pages inspected.

e. Effective rates for:

(1) Preparation

(2) Inspection

(3) Number of lines and pages inspected per inspection.

f. Number of inspections with outcome of pass versus re-inspection.

8.2.2 For each analysis, baselines and thresholds should be defined and updated often as to allow early warning about potential risks. Baselines are also important for understanding and evaluating process improvement initiatives, such as the ones described in Section 8.3.1.c.

8.3 Process Improvement

8.3.1 The project manager should perform the following activities using the results of the analyses defined in Section 8.2, "Evaluation Analyses":

a. Document the analysis results in reports.

b. Review the analysis results with appropriate management and technical personnel.

c. Use the analysis results to promote continuous improvement of the inspection process, such as through recommendations for refinement of:

(1) Effort allocation by inspection process stages

(2) Rates of inspection

(3) Inspection team size and composition

(4) Inspection entry and exit criteria

(5) Re-inspection criteria

(6) Inspection checklist

8.3.2 When possible, the project manager is encouraged to perform root cause analysis on the defects found during an inspection to determine if slippage has occurred (that is, whether the defect was injected during an earlier phase of the lifecycle), and the cause of the slippage. The outcome of such root cause analyses may be used to drive improvements to both the inspection process and the development process itself.

8.3.3 The Center may also perform such analyses to improve Center-level processes.

APPENDIX A. INSPECTION TYPE AND PARTICIPANTS MATRIX

This table summarizes the recommendations for participants in the types of inspections contained in Sections 7.3 through 7.9, by showing which types of expertise should participate in various types of common inspections.

Please note that this table does not contain information regarding inspections of software plans (Section 7.2), or acquisition projects (Section 7.11). These inspections have unique implications regarding the expertise required.

Roles marked as "(SO)" designate that this role participates in inspections of safety-critical components only.

Participants	Inspection Types						
	System Reqt.	SW Reqt.	Arch. Design	Detailed Design	Code	Test Plan	Test Proc.
Persons who participate in defining requirements of the system to be built, and allocating them to sub-systems	X	X					
Persons who are responsible for defining requirements of the software or a software component.		X	X	X	X	X	
Persons who are responsible for integrating the sub-systems	X						
Persons who are responsible for implementing the requirements allocated to the various sub-systems.	X						
Persons who are responsible for implementing the software's requirements.				X	X	X	X
Persons who are responsible for detailed design of the software.			X				
Persons who understand the quality needs of the **system**	X						

Persons who understand the quality needs of the **software component**		X	X	X	X	X	X
Persons who are responsible for designing tests for the software		X		X	X	X	X
Persons who are responsible for verifying and validating systems interfaces			X			X	X
Persons who are responsible for implementing Software Safety	(SO)	(SO)	(SO)	(SO)	(SO)	(SO)	(SO)

APPENDIX B. FORMAL INSPECTION COMPLIANCE MATRIX

Requirement	Location	Requirement	Compliance Y/N/ partial	Tailoring/waiver/deviation comments
1	1.3	Once invoked, relief from requirements in this Standard for application to a specific program or project shall be formally documented as part of program or project requirements and approved by the Technical Authority [SFI- 001].		
2	4.	The project manager shall use established and documented inspection procedures that meet the requirements specified in this Standard. [SFI-002]		
3	4.4	Project managers shall document all the instances of non-compliances to the process requirements laid out in this Standard and the rationale for each. [SFI-003]		
4	5.1	All participants in a formal inspection meeting—i.e., inspectors—shall examine the product presented for inspection and related materials, looking for defects in the product. [SFI-004]		
5	5.1	Only persons who have been formally trained in inspections by taking a NASA-recommended class, and have already participated as an inspector in more than one inspection, shall fill the role of inspection moderator. [SFI-005]		

Requirement	Location	Requirement	Compliance Y/N/ partial	Tailoring/waiver/deviation comments
6	5.2	The formal inspection team shall consist of no less than three inspectors including a moderator and the author. [SFI-006]		
7	5.2	Inspectors shall perform the following basic activities for each inspection (referring to the process steps in Figure 1): [SFI-007]		
8	5.2	Inspectors shall fulfill the following minimum set of roles at each inspection: Author, Moderator, Reader, and Recorder. [SFI-008]		
9	5.3.1	As a basis for the selection of inspectors, the inspection moderator shall identify the key stakeholders in the product under inspection. [SFI-009]		
10	5.3.3	The moderator shall ensure that each inspector has a checklist or other work aid (e.g., set of questions, review scenario) available. [SFI-010]		
11	6.2	The moderator shall specify in the inspection procedure a set of measurable actions that must be completed prior to each type of inspection. [SFI-011]		
12	6.2	Furthermore, the moderator shall specify in the inspection procedure a set of measurable actions that must be completed following each of the required inspections. [SFI-012]		

Requirement	Location	Requirement	Compliance Y/N/ partial	Tailoring/waiver/deviation comments
13	6.3.3.1	The moderator shall ensure that the entry criteria, as specified in Section 6.2, "Entry and Exit Criteria", have been met. [SFI-013]		
14	6.3.3.1	If the moderator determines that the product does not meet the entry criteria, the product shall be returned to the author for further development. [SFI-014]		
15	6.3.4	The moderator shall ensure that all inspectors receive and understand the content of the inspection package. [SFI-015]		
16	6.3.5	During this stage, inspectors shall focus on detecting defects and developing questions by utilizing the inspection work aids, and examining the inspected product for technical accuracy, fulfillment of requirements, and adherence to standards and conventions. [SFI-016]		
17	6.3.5	The moderator shall review and organize the inspectors' defects, questions, and preparation efforts in order to organize the inspection meeting and to decide whether and how to proceed with the inspection meeting. [SFI-017]		
18	6.3.6	The reader shall lead the inspectors through the inspected product and related materials. [SFI-018]		

Requirement	Location	Requirement	Compliance Y/N/ partial	Tailoring/waiver/deviation comments
19	6.3.6	The recorder shall record all defects identified during the inspection meeting. [SFI-019]		
20	6.3.6	The inspectors shall classify the defects according to severity and type as described in Section 6.3.4.3, "Defect Classification." [SFI-020]		
21	6.3.6.1	At the conclusion of each inspection meeting, the moderator shall decide, based on the criteria established in accordance with Section 6.4, “Inspection Procedure Customization,” whether a re-inspection will be performed. [SFI-021]		
22	6.3.6.1	The moderator shall ensure that the author(s) receive the list of classified anomalies or defects and the work product marked with any clerical defects. [SFI-022]		
23	6.3.7	The author shall determine if a third hour is needed. [SFI-023]		
24	6.3.8	The author shall correct all defects identified as major defects. [SFI-024] The author should correct all other non-major defects at the moderator’s discretion when it is cost- and schedule-efficient to do so.		

Requirement	Location	Requirement	Compliance Y/N/ partial	Tailoring/waiver/deviation comments
25	6.3.10	The moderator shall generate and report all the required data for the inspection report, as defined in Section 8.1, "Product and Process Measures." [SFI-025]		
26	6.3.10	The moderator shall ensure that the exit criteria have been met and the work product returned to the author for rework if the criteria are not met. [SFI-026]		
27	8.1	The project manager shall define in the inspection procedure where and in which format the data is to be stored. [SFI-027]		