NASA-STD-2202-93
Effective Date: April 1993
NASA-STD-2202-93
Revalidated March 29, 2001

SOFTWARE FORMAL INSPECTIONS STANDARD

APPROVED: APRIL 1993

FOREWORD

The Software Formal Inspections Standard (hereinafter referred to as the Standard) is designed to support the inspection process of software developed for NASA. Its goal is to provide a framework and model for an inspection process that will detect and eliminate defects as early as possible in the software life cycle. This Standard will have been successfully applied if it accomplishes the following:

- The goals for the project's software inspection process are satisfied.

- Clear descriptions of the software inspection process and products are provided.

- Traceability between products of this process through the software life cycle is maintained.

This Standard describes a uniform process for NASA software formal inspections. It provides:

- A mechanism for ensuring quality is built into the software.

- A means for assuring the quality of the process.

- A means for producing and supporting a software inspection process and the quality assurance aspects of that process for a project.

A common uniform format and content for a software inspection process across NASA projects.

A software inspection process standard tailored to NASA's environment.

General questions concerning this publication should be referred to the Office of Safety and Mission Assurance, NASA Headquarters, Washington, D.C., 20546.

/s/ Frederick D. Gregory

Frederick D. Gregory
Associate Administrator for
Safety and Mission Assurance

1.0 SCOPE, PURPOSE, AND APPLICATION

1.1 SCOPE

This Software Formal Inspections Standard (hereinafter referred to as Standard) is applicable to NASA software. This Standard defines the requirements that shall be fulfilled by the software formal inspections process whenever this process is specified for NASA software.

1.2 PURPOSE

The objective of this Standard is to define the requirements for a process that inspects software products to detect and eliminate defects as early as possible in the software life cycle. The process also provides for the collection and analysis of inspection data to improve the inspection process as well as the quality of the software.

1.3 APPLICATION

The Software Formal Inspections Standard shall be applied to software developed for and by NASA. Refer to Sections 1.3.1 through 1.3.4 for description.

When software is developed for NASA, rather than by NASA this Standard applies when specified in contract clauses and Statements of Work (SOWs). Selection and use of this Standard shall be an option of program or project management (the acquirer), and shall be determined on a program or project basis. The provider shall establish and document inspection procedures that meet these requirements.

When software is developed by NASA, this Standard shall apply if specified in the program plan, memorandum of understanding, etc.

1.3.1 DELIVERABLE SOFTWARE

All new and modified software products deliverable to the acquirer under a contract (i.e., deliverable software) shall be inspected as specified in Section 3.3, "Types of Inspections," during development to demonstrate completeness, correctness, and compliance relative to requirements and adherence to program standards.

1.3.2 SOFTWARE INCLUDED AS PART OF DELIVERABLE HARDWARE (INCLUDING FIRMWARE)

Software included as part of deliverable hardware shall be a candidate for the inspection process.

1.3.3 NONDELIVERABLE SOFTWARE

Software used for development, fabrication, manufacturing process control, testing, or acceptance of deliverable software or hardware (test and acceptance software; software design, test, and analysis tools; compilers; etc.) shall be inspected according to the same inspection requirements as deliverable software to demonstrate completeness, correctness, and compliance relative to requirements and/or adherence to program standards.

1.3.4 COMMERCIAL-OFF-THE-SHELF, REUSED, OR GOVERNMENT-FURNISHED SOFTWARE

Commercial-off-the-Shelf (COTS), reused, or government-furnished software (GFS) products that are modified before being incorporated into deliverable software shall be considered modified software and inspected in the same manner as developed

software.

COTS that is not modified is not normally a candidate for the inspection process.

1.4 TAILORING

This Standard shall be tailored by the acquirer (e.g., NASA program/project manager) in accordance with the classification of the software being developed or acquired. The classification of software shall be determined by the responsible NASA Center or program office per NMI 2410.10A, NASA Software Management, Assurance, and Engineering policy.

Tailoring of this standard consists of the following:

Identifying requirements that are not applicable.

Adding requirements.

Providing quantifiable criteria for the requirements (how often, how many, quality criteria, etc.).

2.0 REFERENCES

2.1 REFERENCED DOCUMENTS

The following references are listed to show their use in the generation of this Standard.

2.1.1 DOCUMENT REFERENCED AS A REQUIREMENT

All NASA software shall satisfy the requirements set forth in:

NASA Software Management, Assurance, and Engineering Policy, NMI 2410.10A, December 12, 1991.

2.1.2 DOCUMENTS REFERENCED AS INFORMATION

The following documents were used in the development of this Standard. Their content is intended to provide supporting information to this Standard but should not be considered to be part of the requirements.

a. IEEE Standard Glossary of Software Engineering Terminology. ANSI/IEEE Standard 729-1983. New York: Institute of Electrical and Electronics Engineers, Inc.

b. IEEE Standard for Software Test Documentation. ANSI/IEEE Standard 829-1983. New York: Institute of Electrical and Electronics Engineers, Inc.

c. IEEE Standard Dictionary of Measures to Produce Reliable Software. IEEE Standard 982.1-1988. New York: Institute of Electrical and Electronics Engineers, Inc.

d. IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard 610.12-1990. New York: Institute of Electrical and Electronics Engineers, Inc.

e. JSC 31011, The Work Package 2 Master Verification Plan, Revision B, April 20, 1990. Houston: NASA-JSC Space Station Projects Office.

f. JSC 31012, Space Station Projects Office, Lexicon, January 1987.

g. NASA Software Acquisition Life Cycle, Version 4.0, 1989. Washington, D.C.: NASA Office of Safety, Reliability, Maintainability, and Quality Assurance.

h. NASA Software Documentation Standard, Software Engineering Program, NASA-STD-2100-91. Washington, D.C.: July 1991.

i. European Space Agency Software Engineering Standards, Issue 2, ESA PSS-05-0, February 1991.

j. Software Maintenance: The Problem and Its Solution, James Martin and Carma McClure. Prentice Hall, 1983.

k. Software Engineering Design, Reliability, and Management, Martin L. Shooman. McGraw Hill, 1983.

l. Formal Inspections for Software Development Course, Revision E, Software Product Assurance, Section 522, MS 301-476; Jet Propulsion Laboratory; Pasadena, CA.

m. Formal Inspections - Manager's Course, Version 1.0, Oct 1989. NASA Software Management and Assurance Program (SMAP). Prepared by John C. Kelly, Ph.D., Software Product Assurance; Jet Propulsion Laboratory; Pasadena, CA.

n. Software Development Formal Inspections Course, Revision G, Software Product Assurance; Section 522, MS125-233; Jet Propulsion Laboratory; Pasadena, CA.

2.2 GLOSSARY

Definitions reprinted in part from IEEE Standard 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, copyright 1990. The information contained herein in italics is copyrighted information of the IEEE, extracted from IEEE Std 610.12-1990, copyright • 1990 by the Institute of Electrical and Electronics Engineers, Inc. This information was written within the context of IEEE Std 610.12-1990. The IEEE takes no responsibility or liability for and will assume no liability for damages resulting from the reader's misinterpretation of said information resulting from the placement and context in this publication. Information is reproduced with the permission of the IEEE.

Acquirer. The person, organization, or company that obtains a product or capability, such as a software system and associated documentation; synonymous with "customer."

Allocation. The process of distributing or assigning for a specific purpose. Examples:

Functional - Allocation of requirements to functions.
Operational - Allocation of functions to operational modes.
Physical - Allocation of requirements or functions to a physical entity (e.g., System, Segment, Element, or Configuration Item).

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.

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 distinct part or element of a computer software configuration item or software product.

Configuration Control. The systematic control of work products.

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

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

Deliverable Software. The code and corresponding documentation that is turned over to the acquirer at specific points throughout the life of a contract.

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. (JSC 31011)

Error. A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically

correct value or condition. (ANSI)

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 programmed instructions and/or computer data that reside in some form of storage element and are required for proper operation of a hardware unit. There are two common types: (1) firmware that requires an integral understanding of the hardware design and its operation, and/or is design implementation-dependent (e.g., machine instructions, control logic, etc.); and (2) firmware that implements system applica tions and/or support functions that do not fall within the limitations in (1) (e.g., database services, task scheduling, etc.), but is packaged in a form of Read Only Memory (ROM) for reasons such as performance, capacity, etc.

Inspection Package. The 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.

Interface. A shared boundary across which information is passed; may be a hardware component, a portion of storage, or registers accessed by two or more computer programs.

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. (ANSI)

Performance. A measure of the ability of a computer system or subsystem to exercise its functions; for example, response time, throughput, number of transactions, etc.

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 person, organization, or company that actually develops the software products to the requirements of the acquirer. The provider may be a contractor or an in-house NASA entity. Because most of NASA software is created, delivered, tested, or maintained under contract, the term is most generally synonymous with "contractor" and "subcontractor."

Quality Assurance (QA). Those assurance activities focused on conformance to standards and procedures.

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

Reliability. The probability that a given software system will operate without failure (of a specified severity) for a specified time in a specified environment.

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 develop ment of the system or system components.

Segment. Relative to a system: an entity consisting of interrelated elements for which a design-to specification is prepared. Segment is the second level in the generic system hierarchy (i.e., System, Segment, Element, and Configuration Item).

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, rules, and associated documentation and data pertaining to the operation of a computer system. Includes programs and operational data contained in firmware. Examples of software include: flight software, ground support equipment software, testing station software, scientific data software for data reduction and modeling analysis, systems software, applications software, etc.

Software Engineering. A generic reference to the discipline and efforts associated with design, code, and test of software developed from requirements defined in a Software Requirements Specification. Software engineering also references the organization that conducts the software development activities for a specific program.

Software Life Cycle. The period of time that starts when a software product is conceived and ends when the product is no longer available for use. The software life cycle typically traditionally includes the following eight phases:

- Concept and Initiation Phase
- Requirements Phase
- Architectural Design Phase
- Detailed Design Phase
- Implementation Phase
- Integration and Test Phase
- Acceptance and Delivery Phase
- Sustaining Engineering and Operations Phase.

Software Product. A software product is defined as either: a. The set of software that has a logical stand-alone identity and function; or, b. The complete set of computer programs, procedures, and associated documentation and data designated for delivery to a user.

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. (IEEE Standard 610.12-1990)

Subapplication. Each of the smaller groups of software into which an application may be divided for the purpose of assigning maintenance responsibility or testing responsibility.

System. The total aggregation of hardware, software, communications, data, human and support elements, and procedures that comprise a complete operational capability.

Test Documentation. The documentation describing the plans for, or results of, the testing of a system or component. Types include test incident report, test log, test plan, test procedure, and test report.

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.

a. 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. (IEEE Standard 610.12-1990)

b. 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.

Verification. The process of evaluating a system or component to determine whether the product of a given life cycle phase satisfies the conditions imposed at the start of that phase.

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.

2.3 ABBREVIATIONS AND ACRONYMS

ANSI American National Standards Institute

COTS Commercial-off-the-Shelf

GFS Government-Furnished Software

IEEE Institute of Electrical and Electronics Engineers, Inc.

ISO International Standards Organization

IT1 Symbol for Test Plan Inspection

IT2 Symbol for Test Procedures Inspection

I0 Symbol for Architectural Design Inspection

I1 Symbol for Detailed Design Inspection

I2 Symbol for Source Code Inspection

JSC Johnson Space Center

NASA National Aeronautics and Space Administration

QA Quality Assurance

ROM Read Only Memory

R0 Symbol for System Requirements Inspection

R1 Symbol for Software Requirements Inspection

SQA Software Quality Assurance

STD Standard

3.0 REQUIREMENTS

This section contains the requirements for implementing this Standard.

3.1 SOFTWARE INSPECTION PROCESS

3.1.1 DEFINITION

As applied to software products and/or associated documentation, inspection is a technical examination process during which a product is examined with the purpose of finding and removing defects as early as possible in the software life cycle. A defect is any occurrence in a software product that is determined to be incomplete or incorrect relative to software requirements and/or program standards.

3.1.2 CHARACTERISTICS

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

Performed routinely and according to established procedures and schedules.

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

Performed by inspectors knowledgeable about the inspection process and the inspected product.

Conducted by at least 3 people, one of whom, the moderator, is

responsible for the effectiveness of the inspection.

Participated in by the producer of the software product who is present at the inspection.

Participated in by inspectors who assume specific roles.

Executed in a specific set of stages.

Designed to produce data for project management, quality evaluation, and inspection process improvement, but not for personnel evaluation.

3.2 ROLES AND RESPONSIBILITIES

3.2.1 INSPECTORS

All participants in an inspection meeting shall be trained in the inspection process and shall be called inspectors. Inspectors shall examine the product presented for inspection and related materials looking for defects in the product. All inspectors shall be able to technically inspect the product.

3.2.1.1ROLES

Inspectors shall fulfill the following minimum set of roles at each inspection: Author, Moderator, Reader, and Recorder. Individual inspectors may fulfill more than one inspection role.

Inspections shall be performed by a minimum of three inspectors, one of whom shall be the author and another shall be the moderator. The roles of reader and recorder shall be fulfilled by any combination of the third inspector, the moderator, and additional inspectors beyond the minimum.

The following responsibilities shall be fulfilled for inspection roles at each inspection:

3.2.1.1.1 AUTHOR.

The author is the producer of the product being inspected. Normally, only persons trained as inspectors shall be allowed to be authors. In addition to looking for defects in the product

presented for inspection, the author shall be responsible for:

Generating all work products to be inspected and provide required reference materials for the overview and the inspection meeting.

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

Modifying the inspected product to correct defects found during the inspection.

Reviewing the corrections with the moderator according to the requirements in Section 3.4.2.7, "Follow-up."

3.2.1.1.2 MODERATOR.

The moderator is the conductor and controller of an inspection. Only inspectors who have been additionally trained and explicitly authorized to serve as moderators shall be allowed to fulfill the role of moderator at inspections. The moderator shall be responsible for the overall effectiveness of each inspection moderated.

The role of moderator in an inspection shall be performed by a person other than the author. Specific responsibilities of the moderator shall be:

Ensure that the entry criteria specified in Section 3.4.1, "Entry Criteria," are met.

Ensure that all inspectors are prepared prior to the inspection meeting.
Focus the inspection meeting on finding defects in the product under inspection.

Classify defects according to requirements in Section 3.4.2.4.2, "Defect Classification."

Disposition defects according to requirements in Section 3.4.4, "Customization," item g.

Assign defects dispositioned for correction to the author.

Verify, personally or by delegation to other inspectors, that all defects dispositioned for correction are corrected prior to re-inspecting and/or authorizing placement of the inspected product under configuration control for delivery to the next phase of the software life cycle; also verify that no new defects are inserted in the correction.

Authorize placement of the inspected product under configuration control (when all conditions in Section 3.4.3, "Exit Criteria," have been met) for delivery to the next phase in the software life cycle.

Collect the data, and generate and file the inspection report specified in Section 3.7, "Required Data."

3.2.1.1.3 READER.

The reader is the presenter of the inspection product to the other inspectors. The role of the reader in an inspection shall be performed by a person other than the author. In addition to looking for defects in the product presented for inspection, the reader shall lead the other inspectors through the inspected product and related materials in a logical progression, paraphrasing and summarizing each section.

3.2.1.1.4 RECORDER.

The role of recorder in an inspection shall be performed by a person other than the author. In addition to looking for defects in the product presented for inspection, the recorder shall document each defect identified during the inspection meeting, including its classification, and provide the resulting list to the moderator at the end of the inspection meeting.

3.2.1.2CANDIDATES

Inspectors not fulfilling the roles of author or moderator shall be chosen from the other candidates listed below.

3.2.1.2.1 PEERS.

Peers are persons working on the same phase of the software life cycle as the author but are not directly responsible for generating the inspected product.

3.2.1.2.2 REPRESENTATIVES FROM PREVIOUS PHASES IN THE SOFTWARE LIFE CYCLE.

The representatives from the previous phases in the software life cycle shall look for defects in the product presented for inspection from the perspective of their areas of expertise and knowledge of the intended characteristics of the product.

3.2.1.2.3 REPRESENTATIVES FROM FOLLOWING PHASES IN THE SOFTWARE LIFE CYCLE.

The representatives from the following phases in the software life cycle shall look for defects in the product presented for inspection from the perspective of their areas of expertise, and from knowledge of the needs of the following phases in the software life cycle.

3.2.1.2.4 REPRESENTATIVES FROM INTERFACING COMPONENTS OR CONFIGURATION ITEMS.

The representatives from interfacing components or configuration items shall look for defects in the product presented for inspection from the perspective of their areas of expertise, and from knowledge of the interface requirements of the interfacing components or configuration items.

3.2.2 USERS

The user of the software product presented for inspection may participate in System Requirements (R0), Software Requirements (R1), and Test Plan (IT1) inspections (defined in Section 3.3, "Types of Inspections") to ensure that user needs and expectations are satisfied and that the desired product will be produced.

The extent of user participation in other types of inspections shall be determined by the provider. The user of the software product may fulfill any of the inspector roles defined in Section 3.2.1.1, "Roles," except for those of author and moderator.

3.2.3 SOFTWARE QUALITY ASSURANCE

Software Quality Assurance (SQA) shall assure compliance with process requirements by working with management (providing process and procedural reviews, and recommendations) in defining the inspection procedures and records.

SQA shall assure compliance to documented inspection procedures by:

Verifying that the data specified in Section 3.7, "Required Data" have been collected.

Selectively reviewing inspection packages for required inspection materials.

Participating in inspection meetings to whatever extent is deemed necessary by SQA, including fulfillment of any of the inspection roles except author.

By performing, participating in, and/or assuring the analysis in Section 3.5, "Process Evaluation," SQA shall provide an independent evaluation of the effectiveness of the inspection process and the product quality.

SQA shall assure that:

Reports of inspection process evaluation/analysis are:

1. Defined and scheduled.
2. Provided as needed to:
   a) Validate positive trends in the inspection process.
   b) Address adverse trends in the inspection process.
3. Reviewed with appropriate management and/or technical personnel.
4. Considered in inspection process improvements.

All inspection process improvements are documented and tracked for analysis and incorporation, and that inspection anomalies are documented and tracked for analysis and correction.

3.3 TYPES OF INSPECTIONS

The following are generally recognized types of inspections. Additional types of inspections may be conducted using the inspection process.

3.3.1 SYSTEM REQUIREMENTS INSPECTION (R0)

The work products inspected shall be the high-level requirements for software systems. This type of inspection may be applied to multiple levels of system requirements.

The purpose of the high-level requirements inspection shall be to:

- Ensure proper allocation of functions to software, firmware, hardware, and operations.

- Validate all external usage interfaces.

- Verify that all the software system functions are identified and broken into configuration items.

- Ensure all configuration items within the software system are identified.

- Verify that the identified configuration items provide all functions required of them.

- Ensure that all interfaces between configuration items within the software system are identified.

- Verify correctness of the software system structure.

- Ensure that all quantifiable requirements and requirement attributes have been specified.

- Ensure that the requirements are verifiable.

3.3.2 SOFTWARE REQUIREMENTS INSPECTION (R1)

The work products inspected shall be the detailed requirements for specific software components and/or modules.

The purpose of the software requirements inspection shall be to:

Verify a complete and accurate specification of each of the following:

1. Software functions
2. Input and output
3. States and modes
4. Response time requirements
5. Interfaces.

Ensure specifications are included for error detection and recovery, reliability, maintainability, performance, and accuracy.

Ensure the traceability of requirements from higher level documents.

Verify that the requirements provide a sufficient base for the software design.

Verify that the requirements are measurable, consistent, and testable.

3.3.3 ARCHITECTURAL DESIGN INSPECTION (I0)

The work product inspected shall be the high-level software system design.

The purpose of the architectural design inspection shall be to:

Ensure the design meets approved requirements.

Validate all interfaces among modules within each component.

Review the list of modules and the general function(s) of each module.

Validate fault detection, identification, and recovery requirements.

Verify the component structure meets the requirements.

Validate the selection of reusable components.

Ensure traceability of the design to the approved requirements.

Validate input and output interfaces.

3.3.4 DETAILED DESIGN INSPECTION (I1)

The work product inspected shall be the software component and/or module design at the detailed level.

The purpose of the detailed design inspection shall be to:

Ensure that the design meets the approved requirements.

Validate all logic algorithms, data structures, and calls within each module.

Verify that the detailed design is complete for each module.

Ensure traceability of the design to the approved requirements.

Ensure that all required and/or applicable programming standards are followed.

Ensure that detailed design meets requirements and is traceable to the high-level software system design.

3.3.5 SOURCE CODE INSPECTION (I2)

The work product inspected shall be the module source code. The purpose of the source code inspection shall be to:

Ensure that the code meets the approved requirements.

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

Verify that the code implements the detailed design, and that all required/applicable standards are satisfied.

Ensure traceability of the code to the approved requirements.

Ensure that the code meets requirements and is traceable to the detailed design.

3.3.6 TEST PLAN INSPECTION (IT1)

The work product inspected shall be a software test plan for the software capabilities required by the detailed level of requirements.

The purpose of the test plan inspection shall be to:

Detect defects and misconceptions in the definition of the test plan.

Ensure that all new and modified software functions operate correctly within the intended environment and according to approved requirements.

Ensure that all new and modified interfaces will be verified.

Identify and eliminate extraneous or obsolete test plans.

Ensure that each requirement will be tested.

3.3.7 TEST PROCEDURES INSPECTION (IT2)

The work products inspected shall be test procedures for software capabilities required by the detailed level of requirements.

The purpose of the test procedures inspection shall be to:

Verify that the set of test procedures meets the objective of the test plan.

Verify 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.

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

requirement(s).

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

Ensure that each test procedure will execute without errors.

3.4 PROCESS ELEMENTS

3.4.1 ENTRY CRITERIA

The inspection procedure shall specify a set of measurable actions that must be completed prior to each type of inspection. Completion of these actions shall ensure that all activities related to the preceding phase of the software life cycle have been completed or addressed prior to the corresponding inspection.

3.4.2 STAGES

Data required in Section 3.7, "Required Data," shall be recorded at the appropriate stage of the inspection process.

The inspection process shall consist of the following chronological stages for each type of inspection required as shown below.

3.4.2.1PLANNING

Planning shall be the stage at which the package contents, required support, and scheduling of an inspection are defined.

The inspection process shall require completion of the following activities during the planning stage:

3.4.2.1.1 ENTRY CRITERIA CHECK.

The moderator shall ensure that the entry criteria have been met. If the product does not meet the entry criteria, or if the moderator does not think that the product is ready for inspection, the moderator shall return the product to the author for further development.

3.4.2.1.2 INSPECTION PACKAGE CONTENTS.

The number of product elements to be inspected at any given inspection shall be chosen to allow the corresponding inspection meeting to cover all of the material in 2 hours or less at a rate of inspection less than or equal to the maximum rate allowed for this type inspection, as required in Section 3.4.4, "Customization," item e. The products and documentation to be inspected, as well as the reference materials required for the specific type of inspection, shall be generated.

3.4.2.1.3 INSPECTORS.

Based on the contents of the inspection package, inspectors shall be identified for each inspection, notified of their responsibility to support the inspection, and of their role(s). Stages shall be delayed until designated inspectors are available to participate and provide their support. The inspection procedures shall define the method by which the reader of an inspection shall be appointed.

3.4.2.1.4 INSPECTION SCHEDULING.

Inspection meetings shall be scheduled at times when all inspectors can attend. Inspection meetings shall be scheduled far enough in the future to allow at least the minimum lead time required for the specific type of inspection, as required in Section 3.4.4, "Customization," item d.

3.4.2.1.5 DISTRIBUTION.

During this step, the inspection package shall be delivered to inspectors.

3.4.2.2OVERVIEW

The inspection procedure shall specify, for each type of inspection, the conditions under which an overview shall be presented in a formal meeting. The overview shall be an educational briefing, either oral or written, given prior to the inspection meeting, which shall explain the product to be inspected, and related 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. The overview shall be provided at the time the inspection product and related materials are distributed.

3.4.2.3PREPARATION

Preparation shall be the stage at which inspectors individually get ready for the inspection meeting. The author's participation in the preparation stage is optional. During this stage, inspectors shall focus on detecting defects and developing questions by examining the inspected product for technical accuracy, fulfillment of requirements, and adherence to standards and conventions. Possible defects and questions shall be documented and submitted to the moderator prior to the start of the inspection meeting to help ensure the inspection team is adequately prepared and the inspection meeting can be held.

3.4.2.4INSPECTION MEETING

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

The inspectors shall be led through the inspected product and related materials by the reader. The inspectors shall identify defects; however, they shall not provide solutions.
All defects identified during the inspection meeting shall be recorded by the recorder.

Defects shall be dispositioned according to the requirements in Section 3.4.4, "Customization," item g. All other issues that are not defects shall be dispositioned according to the requirements in Section 3.4.4, "Customization," item h. The defects shall be classified according to severity and type as described in Section 3.4.2.4.2, "Defect Classification."

At the conclusion of each inspection meeting, the moderator shall decide, based on the requirements in Section 3.4.4, "Customization," item f, if a re-inspection of all or part of the product shall be performed after the corrections of the defects identified by the first inspection have been made.

3.4.2.4.1 INSPECTION CONTINUATION - ADDITIONAL MEETINGS.

The inspection meeting shall be controlled by the moderator so that if it exceeds 2 hours, it is stopped and a continuation meeting scheduled for a later date.

3.4.2.4.2 DEFECT CLASSIFICATION.

All defects identified in the inspected product during an inspection shall be classified by severity and type of defect.

a. Severity of Defect: Each defect in the inspected product shall be 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, or prevent the attainment of a required result, and would result in a Discrepancy Report.

   2. Minor Defect

   A defect in the product under inspection which, if not fixed, would not cause a malfunction, would not prevent the attainment of a required result, and would not result in a Discrepancy Report.

b. Type of Defects: Defects shall be further classified to include at least the following minimum set:

   1. Data
   2. Requirements compliance
   3. Interfaces
   4. Logic
   5. Standards compliance
   6. Performance
   7. Readability.

3.4.2.5THIRD HOUR

The third hour shall be an optional, informal, additional meeting or activity that shall be 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.

The author shall determine if a third hour is needed.

Participants at the third hour shall 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.

3.4.2.6REWORK

Rework shall be the stage at which all defects dispositioned for correction are corrected.

3.4.2.6.1 RE-INSPECTION.

Re-inspection is a repetition of the inspection process for a complete or partial set of products that have been previously inspected. Separate inspection reports shall be generated for each re-inspection.

3.4.2.7FOLLOW-UP

Follow-up shall be the stage at which the moderator verifies, personally or by delegation, that all defects dispositioned for correction have been corrected, and that no additional defects have been introduced.

All required data for the inspection report shall be generated and reported at this stage, and the moderator shall ensure that the exit criteria have been met.

3.4.3 EXIT CRITERIA

The inspection procedure shall specify a set of measurable actions that must be completed following each of the required inspections before the inspected product may be placed under configuration control so its development can proceed to the next phase of the software life cycle. These actions shall ensure that all major defects have been corrected.

3.4.4 CUSTOMIZATION

The inspection process shall be customized as follows for each type of inspection:

The need, applicability, and contents of checklists for each type of inspection.

The required contents of the inspection package for each type of inspection in terms of:

1. Products and documentation to be inspected
2. Reference materials
3. Inspection meeting notice and scheduling information.

The mandatory number of inspectors who must participate in an inspection and the roles each of them must fulfill.
The required minimum lead time that shall be allowed between distribution of the product to be inspected and related materials, and the occurrence of the corresponding inspection meeting. This time shall be long enough to allow adequate preparation by the persons doing the inspection.

The maximum rate at which each type of inspection shall be performed in terms of pages or lines per hour, based on available data and project experience.

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

A set of options for dispositioning minor defects identified during the inspection meeting as well as the criteria for selecting each type of disposition.

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

A formal method to authorize, and document in the inspection report, deviations from specific required inspection stages or actions.

3.4.5 TRAINING REQUIREMENTS

All inspectors shall receive training in the inspection process and in their responsibilities within that process. Persons who may act as moderators shall receive additional training on the responsibilities of that role. Only persons who have been trained as moderators shall be allowed to moderate inspections.

3.5 PROCESS EVALUATION

SQA shall assure the following minimum set of trend analyses is performed to identify positive or adverse trends in the inspection process at the earliest possible opportunity using the data collected in Section 3.7, "Required Data":

Total defects (major/minor) by delivery/release ID

Total defects (major/minor) by delivery/release ID by inspection type

Defect density of products (number of major/minor defects per lines/pages)
Defect density of defect types sorted by:

1. Inspection type
2. Inspection type and application
3. Inspection type and department.

Labor hours (overview, planning, preparation, inspection, third hour, follow-up, and rework) versus number of:

1. Defects found
2. Lines/pages inspected.

Effective rates for:

1. Preparation
2. Inspection
3. Number of lines/pages inspected per inspection.

Phase in the software life cycle where defects should have been found

Number of inspections complete versus planned.

3.6 PROCESS IMPROVEMENT

Results of the analyses required in Section 3.5, "Process Evaluation," shall be:

Documented in reports.

Reviewed with appropriate management and/or technical personnel.

Used to promote continuous improvement of the inspection process through recommendations for refinement of:

1. Process stages
2. Rates and/or volumes for inspection stages
3. Re-inspection criteria.

3.7 REQUIRED DATA

The moderator shall collect and file the following data for each inspection in an inspection report. Each inspection report shall be signed by the moderator.

3.7.1 DESCRIPTION OF ORGANIZATIONAL AREA GENERATING PRODUCT INSPECTED

The following characteristics of the organizational area responsible for producing the inspected product shall be included in the inspection report:

Project name at contract level
Manager responsible for product
System: Functional Project
Department producing product
Application - to be customized
Subapplication - to be customized.

3.7.2 INSPECTED PRODUCT DESCRIPTION

The following characteristics of the product to be inspected

shall be included in the inspection report:

- Inspection type
- Element descriptions
- Element names and versions
- Size of work product
- Targeted delivery/release identification
- Change authorization document(s).

3.7.3 DEFECT INTRODUCTION PHASE

The phase in the software life cycle where each defect was introduced.

3.7.4 DEFECT DETECTION PHASE

The phase in the software life cycle where each defect was found.

3.7.5 DEFECT DISPOSITION

The disposition of each defect identified during the inspection according to the requirements in Section 3.4.4, "Customization," item g.

3.7.6 INSPECTION PROCESS DESCRIPTION

The following characteristics of the inspection process shall be collected by the moderator and included in the inspection report:

- Inspection date
- First or re-inspection
- Prior inspection date, if applicable
- Overview date, if applicable
- Names of inspectors, excluding author
- Roles of inspectors
- Planning time for author and moderator
- Inspection meeting duration
- Overview meeting duration
- Preparation time for each inspector
- Third Hour time for each inspector
- Rework time
- Follow-up time
- Re-inspection required; target date if it is

Number and type of major defects found
Number and type of minor defects found
Number of major defects corrected
Number of minor defects corrected
Authorized deviations list
Inspection close date.

4.0 QUALITY ASSURANCE PROVISIONS

4.1 SOFTWARE QUALITY ASSURANCE

Although product quality is the responsibility of the development organizations, the responsibility for Software Quality Assurance (SQA) is vested in independent SQA groups. SQA performance and activity relevant to formal inspections are described in Sections 3.2.3, "Software Quality Assurance," and 3.5, "Process Evaluation," of this document.

5.0 PACKAGING

Not applicable. There are no packaging requirements associated with this Standard.

6.0 ADDITIONAL INFORMATION

Not applicable. There is no additional information furnished in this document.