1. Scope and References.

1.1 Scope. The purpose of this standard is to provide uniform, minimum acceptable requirements for preparation and content of Software Quality Assurance Plans (SQAPs).

In considering adoption of this standard, regulatory bodies should be aware that specific application of this standard may already be covered by one or more IEEE or ANSI standards documents relating to quality assurance, definitions, or other matters. It is not the purpose of this document to supersede, revise or amend existing standards directed to specific industries or applications.

This standard applies to the development and maintenance of critical software. For non-critical software, or for software already developed, a subset of the requirements of this standard may be applied.

The existence of this standard should not be construed to prohibit additional content in a SQAP. An assessment should be made for the specific software item to assure adequacy of coverage. Where this standard is invoked for an organization or project engaged in producing several software items, the applicability of the standard should be specified for each of the software items.

1.2 References. The standards listed below should be used for further information. In using these references, the latest revisions should be obtained. Compliance with this standard does not require nor imply compliance with any of those listed.


[9] ANSI/IEEE Std 983-1986, IEEE Guide for Software Quality Assurance Planning. (This is being redesignated to 730.2.)


2. Definitions and Acronyms

2.1 Definitions. The definitions listed below establish meaning in the context of this standard. Other definitions can be found in ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology, or latest revision thereof [3]. For the purpose of this standard, the term "software" includes firmware, documentation, data, and execution control statements (e.g., command files, job control language, etc).

branch metric. The result of dividing the total number of modules in which every branch has been executed at least once by the total number of modules. (This assumes that the modules are essentially the same size.)

critical software. Software whose failure would impact safety or cause large financial or social losses.

decision point metric. The result of dividing the total number of modules in which every decision point has had (1) all valid conditions, and (2) at least one invalid condition, correctly processed, divided by the total number of modules. (This assumes that the modules are essentially the same size.)

domain metric. The result of dividing the total number of modules in which one valid sample and one invalid sample of every class of input data items (external messages, operator inputs, and local data) have been correctly processed, by the total number of modules. (This assumes that the modules are essentially the same size.)

error message metric. The result of dividing the total number of error messages that have been formally demonstrated, by the total number of error messages.

quality assurance. A planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirements.

requirements demonstration metric. The result of dividing the total number of separately identified requirements in the software requirements specification (SRS) that have been successfully demonstrated by the total number of separately identified requirements in the SRS.

2.2 Acronyms. The following alphabetical contractions appear within the text of this standard:

CDR – Critical Design Review.
PDR – Preliminary Design Review.
SCMP – Software Configuration Management Plan
SCMPR – Software Configuration Management Plan Review
SDD – Software Design Description
SQA – Software Quality Assurance
SQAP – Software Quality Assurance Plan
SRR – Software Requirements Review
SRS – Software Requirements Specification
SVVP – Software Verification and Validation Plan
SVVPR – Software Verification and Validation Plan Review
SVVR – Software Verification and Validation Report
UDR – User Documentation Review
3. **Software Quality Assurance Plan**

The Software Quality Assurance Plan shall include the sections listed below to be in compliance with this standard. The sections should be ordered in the described sequence. If the sections are not ordered in the described sequence, then a table shall be provided at the end of the SQAP that provides a cross-reference from the lowest numbered subsection of this standard to that portion of the SQAP where that material is provided. If there is no information pertinent to a section, that following shall appear below the section heading, *This section is not applicable to this plan.* together with the appropriate reasons for the exclusion.

1. **Purpose**
2. **Reference Documents**
3. **Management**
4. **Documentation**
5. **Standards, Practices, Conventions, and Metrics**
6. **Reviews and Audits**
7. **Test**
8. **Problem Reporting and Corrective Action**
9. **Tools, Techniques, and Methodologies**
10. **Code Control**
11. **Media Control**
12. **Supplier Control**
13. **Records Collection, Maintenance, and Retention**
14. **Training**
15. **Risk Management**

Additional sections may be added as required.

Some of the material may appear in other documents should be made in the body of the SQAP. In any case, the contents of each section of the plan shall be specified either directly or by reference to another document.

The SQAP shall be approved by the chief operating officer of each unit of the organization having responsibilities defined within this SQAP or their designated representatives.

Details for each section of the SQAP are described in 3.1 through 3.15 of this standard.

3.1 **Purpose (Section 1 of the SQAP).** This section shall delineate the specific purpose and scope of the particular SQAP. It shall list the name(s) of the software items to be covered by the SQAP and the intended use of the software. It shall state the portion of the software life cycle covered by the SQAP for each software item specified.

3.2 **Reference Documents (Section 2 of the SQAP).** This section shall provide a complete list of documents referenced elsewhere in the text of the SQAP.

3.3 **Management (Section 3 of the SQAP).** This section shall describe organization, tasks, and responsibilities.

3.3.1 **Organization.** This paragraph shall depict the organizational structure that influences and controls the quality of the software. This shall include a description of each major element of the organization together with the delegated responsibilities. Organizational dependence or independence of the elements responsible for SQA from those responsible for software development and use shall be clearly described or depicted.

3.3.2 **Tasks.** This paragraph shall describe (a) that portion of the software life cycle covered by the SQAP, (b) the tasks to be performed with special emphasis on software quality assurance activities, and (c) the relationships between these tasks and the planned major check points. The
sequence of tasks shall be indicated.

3.3.3 Responsibilities. This paragraph shall identify the specific organizational elements responsible for each task.

3.4 Documentation (Section 4 of the SQAP)

3.4.1 Purpose. This section shall perform the following functions:
(1) Identify the documentation governing the development, verification and validation, use, and maintenance of the software.
(2) State how the documents are to be checked for adequacy. This shall include the criteria and the identification of the review or audit by which the adequacy of each document shall be confirmed, with reference to Section 6 of the SQAP.

3.4.2 Minimum Documentation Requirements. To ensure that the implementation of the software satisfies requirements, the following documentation is required as a minimum:

3.4.2.1 Software Requirements Specification (SRS). The SRS shall clearly and precisely describe each of the essential requirements (functions, performances, design constraints, and attributes) of the software and external interfaces. Each requirement shall be defined such that its achievement is capable of being objectively verified by a prescribed method; for example, inspection, analysis, demonstration, or test.

3.4.2.2 Software Design Description (SDD). The SDD shall depict how the software will be structured to satisfy the requirements in the SRS. The SDD shall describe the components and subcomponents of the software design, including data bases and internal interfaces. The SDD shall be prepared first as the Preliminary SDD (also referred to as the Top-Level SDD) and shall be subsequently expanded to produce the Detailed SDD.

3.4.2.3 Software Verification and Validation Plan (SVVP). The SVVP shall identify and describe the methods (for example, inspection, analysis, demonstration, or test) to be used:
(1) To verify that (a) the requirements in the SRS have been approved by an appropriate authority, (b) the requirements in the SRS are implemented in the design expressed in the SDD; and (c) the design expressed in the SDD is implemented in the code.
(2) To validate that the code, when executed, complies with the requirements expressed in the SRS.

3.4.2.4 Software Verification and Validation Report (SVVR). The SVVR shall describe the results of the execution of the SVVP.

3.4.2.5 User Documentation. User documentation (eg, manual, guide, etc) shall specify and describe the required data and control inputs, input sequences, options, program limitations, and other activities or items necessary for successful execution of the software. All error messages shall be identified and corrective actions described. A method of describing user-identified errors or problems to the developer or the owner of the software shall be described. (Embedded software that has no direct user interaction has no need for user documentation and is therefore exempted from this requirement.)

3.4.2.6 Software Configuration Management Plan (SCMP). The SCMP shall document methods to be used for identifying software items, controlling and implementing changes, and recording and reporting change implementation status.

3.4.3 Other. Other documentation may include the following:
(1) Software Development Plan
(2) Standards and Procedures Manual.
3.5 Standards, Practices, Conventions and Metrics (Section 5 of the SQAP)

3.5.1 Purpose. This section shall:
(1) Identify the standards, practices, conventions, and metrics to be applied.
(2) State how compliance with these items is to be monitored and assured.

3.5.2 Content. The subjects covered shall include the basic technical, design, and programming activities involved, such as documentation, inspection, and testing. As a minimum, the following information shall be provided:
(1) Documentation Standards
(2) Logic Structure Standards
(3) Coding Standards
(4) Commentary Standards
(5) Testing Standards and Practices
(6) Selected Software Quality Assurance Product and Process Metrics such as:
   (a) Branch Metric
   (b) Decision Point Metric
   (c) Domain Metric
   (d) Error Message Metric
   (e) Requirements Demonstration Metric

3.6 Reviews and Audits (Section 6 of the SQAP)

3.6.1 Purpose. This section shall:
(1) Define the technical and managerial reviews and audits to be conducted.
(2) State how the reviews and audits are to be accomplished.
(3) State what further actions are required and how they are to be implemented and verified.

3.6.2 Minimum Requirements. As a minimum, the following reviews and audits shall be conducted:

3.6.2.1 Software Requirements Review (SRR). The SRR is held to ensure the adequacy of the requirements as stated in the SRS.

3.6.2.2 Preliminary Design Review (PDR). The PDR (also known as top-level design review) is held to evaluate the technical adequacy of the preliminary design (also known as top-level design) of the software as depicted in the preliminary software design description (SDD).

3.6.2.3 Critical Design Review (CDR). The CDR (also known as detailed design review) is held to determine the acceptability of the detailed software designs as depicted in the detailed software design description in satisfying the requirements of the SRS.

3.6.2.4 Software Verification and Validation Plan Review (SVVPR). The SVVPR is held to evaluate the adequacy and completeness of the verification and validation methods defined in the SVVP.

3.6.2.5 Functional Audit. This audit is held prior to the software delivery to verify that all requirements in the specified in the SRS have been met.

3.6.2.6 Physical Audit. This audit is held to verify that the software and its documentation are internally consistent and are ready for delivery.
3.6.2.7 **In-Process Audits.** In-process audits of a sample of the design are held to verify consistency of the design, including:

1. Code versus design documentation
2. Interface specifications (hardware and software)
3. Design implementations versus functional requirements
4. Functional requirements versus test descriptions

3.6.2.8 **Managerial Reviews.** Managerial reviews are held periodically to assess the execution of all of the actions and the items identified in the SQAP. These reviews shall be held by an organizational element independent of the unit being reviewed, or by a qualified third party. This review may require additional changes in the SQAP itself.

3.6.2.9 **Software Configuration Management Plan Review (SCMPR).** The SCMPR is held to evaluate the adequacy and completeness of the configuration management methods defined in the SCMP.

3.6.2.10 **Post Mortem Review.** This review is held at the conclusion of the project to assess the development activities implemented on that project and to provide recommendations for appropriate actions.

3.6.3 **Other.** Other reviews and audits may include the user documentation review (UDR). This review is held to evaluate the adequacy (e.g., completeness, clarity, correctness, and usability) of user documentation.

3.7 **Test (Section 7 of the SQAP).** This section shall identify all the tests not included in the SVVP for the software covered by the SQAP and shall state the methods to be used.

3.8 **Problem Reporting and Corrective Action (Section 8 of the SQAP).** This section shall:

1. Describe the practices and procedures to be followed for reporting, tracking, and resolving problems identified in both software items and the software development and maintenance process.
2. State the specific organizational responsibilities concerned with their implementation.

3.9 **Tools, Techniques, and Methodologies (Section 9 of the SQAP).** This section shall identify the special software tools, techniques, and methodologies that support SQA, state their purposes, and describe their use.

3.10 **Code Control (Section 10 of the SQAP).** This section shall define the methods and facilities used to maintain, store, secure, and document controlled versions of the identified software during all phases of the software life cycle. This may be implemented in conjunction with a computer program library. This may be provided as part of the SCMP. If so, an appropriate reference shall be made thereto.

3.11 **Media Control (Section 11 of the SQAP).** This section shall state the methods and facilities used to (a) identify the media for each computer product and the documentation required to store the media, including the copy and restore process, and (b) protect computer program physical media from unauthorized access or inadvertent damage or degradation during all phases of the software life cycle. This may be provided as part of the SCMP. If so, an appropriate reference shall be made thereto.

3.12 **Supplier Control (Section 12 of the SQAP).** This section shall state the provisions for assuring that software provided by suppliers meets established requirements. In addition, this section shall state the methods that will be used to assure that the software supplier receives adequate and complete requirements. For previously-developed software, this section shall state the methods the methods to be used to assure the suitability of the product for use with the
software items covered by the SQAP. For software that is to be developed, the supplier shall be required to prepare and implement a SQAP in accordance with this standard. This section shall also state the methods to be employed to assure that the developers comply with the requirements of this standard.

3.13 Records Collection, Maintenance, and Retention (Section 13 of the SQAP). This section shall identify the SQA documentation to be retained, shall state the methods and facilities to be used to assemble, safeguard, and maintain this documentation, and shall designate the retention period.

3.14 Training (Section 14 of the SQAP). This section shall identify the training activities necessary to meet the need of the SQAP.

3.15 Risk Management (Section 15 of the SQAP). This section shall specify the methods and procedures employed to identify, assess, monitor, and control areas of risk arising during the portion of the software life cycle covered by the SQAP.