

Software Quality Assurance Plan

Contents

SECTION

1. Purpose
2. Reference Documents
3. Management
 - 3.1 Organization
 - 3.2 Tasks
 - 3.3 Responsibilities
4. Documentation
 - 4.1 Purpose
 - 4.2 Minimum Documentation Requirements
 - 4.2.1 Software Requirements Specification (SRS)
 - 4.2.2 Software Design Description (SDD)
 - 4.2.3 Software Verification and Validation Plan (SVVP)
 - 4.2.4 Software Verification and Validation Report (SVVR)
 - 4.2.5 User Documentation
 - 4.2.6 Software Configuration Management Plan (SCMP)
 - 4.3 Other
5. Standards, Practices, Conventions and Metrics
 - 5.1 Purpose
 - 5.2 Content
6. Reviews and Audits
 - 6.1 Purpose
 - 6.2 Minimum Requirements
 - 6.2.1 Software Requirements Review (SRR)
 - 6.2.2 Preliminary Design Review (PDR)
 - 6.2.3 Critical Design Review (CDR)
 - 6.2.4 Software Verification and Validation Plan Review (SVVPR)
 - 6.2.5 Functional Audit
 - 6.2.6 Physical Audit
 - 6.2.7 In-Process Audits
 - 6.2.8 Managerial Reviews
 - 6.2.9 Software Configuration Management Plan Review (SCMPR)
 - 6.2.10 Post Mortem Review
 - 6.3 Other
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

1. Purpose

Delineate the specific purpose and scope of the particular Software Quality Assurance Plan (SQAP). List the name(s) of the software items covered by the SQAP and the intended use of the software. State the portion of the software life cycle covered by the SQAP for each software item specified.

2. Reference Documents

Provide a complete list of documents referenced elsewhere in the text of the SQAP.

3. Management

Describe the organization, tasks and responsibilities. (Refer as necessary to ANSI/IEEE Std 1002-1987 and ANSI/IEEE Std 1058.1-1988.)

3.1 Organization

Depict the organizational structure that influences and controls the quality of the software. Include a description of each major element of the organization together with the delegated responsibilities. Clearly describe or depict the organizational dependence or independence of the elements responsible for SQA from those responsible for software development and use.

3.2 Tasks

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. Indicate the sequence of tasks.

3.3 Responsibilities

Identify the specific organizational elements responsible for each task.

4. Documentation

4.1 Purpose

- *Identify the documentation governing the development, verification and validation, use and maintenance of the software.*
- *State how the documents are to be checked for adequacy. 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.*

4.2 Minimum Documentation Requirements

To ensure that the implementation of the software satisfies requirements, the following documentation is required as a minimum:

4.2.1 Software Requirements Specification (SRS)

The SRS shall clearly and precisely define each of the essential requirements (functions, performances, design constraints, and attributes) of the software and the external interfaces. Each requirement shall be

defined such that its achievement is capable of being verified and validated by a prescribed method; for example, inspection, analysis, demonstration or test. *(Refer as necessary to ANSI/IEEE Std 830-1984.)*

4.2.2 Software Design Description (SDD)

The SDD shall depict how the software will be structure 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. *(Refer as necessary to ANSI/IEEE Std 1016-1987.)*

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:

- the requirements in the SRS have been approved by an appropriate authority
- the requirements in the SRS are implemented in the design expressed in the SDD
- the design expressed in the SDD is implemented in the code

1. To validate that the code, when executed, complies wit the requirements expressed in the SRS *(Refer as necessary to ANSI/IEEE Std 829-1983, ANSI/IEEE Std 1008-1987, and ANSI/IEEE Std 1012-1986.)*

4.2.4 Software Verification and Validation Report (SVVR)

The SVVR shall describe the results of the execution of the SVVP.

4.2.5 User Documentation

User documentation (e.g., 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 tot he developer or the owner of the software shall be described. *(Embedded software that has no direct user interaction has no need for user documentation as is therefore exempted from this requirement. Refer as necessary to ANSI/IEEE Std 1063-1988.)*

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 implementatio0n status. *(Refer as necessary to ANSI/IEEE Std 828-1983 and ANSI/IEEE Std 1042-1987; see also to ANSI/IEEE Std 1033-1985.)*

4.3 Other

Other documentation may include the following:

- *Software Development Plan*
- *Standards and Procedures Manual*
- *Software Project Management Plan*
- *Software Maintenance Manual*

5. Standards, Practices, Conventions and Metrics

5.1 Purpose

- *Identify the standards, practices, conventions and metrics to be provided.*
- *State how compliance with these items is to be monitored and assured.*

5.2 Content

The subjects covered shall include the basic technical, design, and programming activities involved, such as documentation, variable and module naming, programming, inspection, and testing. As a minimum the following information shall be provided (refer as necessary to ANSI/IEEE Std 990-1987, ANSI/IEEE Std 982.1-1988 and ANSI/IEEE Std 982.2-1988):

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:*
 - *Branch metric*
 - *Decision point metric*
 - *Domain metric*
 - *Error message metric*
 - *Requirements demonstration metric*

6. Reviews and Audits

6.1 Purpose

- *Define the technical and managerial reviews and audits to be conducted.*
- *State how the reviews and audits are to be accomplished.*
- *State what further actions are required and how they are to be implemented and verified.*

6.2 Minimum Requirements

As a minimum, the following reviews and audits shall be conducted:

6.2.1 Software Requirements Review (SRR)

The SRR is held to ensure the adequacy of the requirements stated in the SRS.

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.

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.

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.

6.2.5 Functional Audit

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

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.

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

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 to the SQAP itself.

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.

6.2.10 Post Mortem Review

This review is held at the conclusion of the process to assess the development activities implemented on that project and to provide recommendations for appropriate actions.

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.

7. Test

Identify all the tests not included in the SVVP for the software covered by the SQAP, and state the methods to be used. (Refer as necessary to ANSI/IEEE Std 829-1983 and ANSI/IEEE Std 1008-1987.)

8. Problem Reporting and Corrective Action

- 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.*

9. Tools, Techniques and Methodologies

Identify the special software tools, techniques and methodologies that support SQA, state their purposes and describe their use.

10. Code Control

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.

11. Media Control

State the methods and facilities to be 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.

12. Supplier Control

State the provisions for assuring that software provided by suppliers meets established requirements. In addition, state the methods that will be used to assure that the software supplier receives adequate and complete requirements. For previously-developed software, state the methods to be used to assure the suitability of the product for use with the software items covered by the SQAP. For the software that is to be developed, the supplier shall be required to prepare and implement a SQAP in accordance with ANSI/IEEE Std 730-1989, "IEEE Standard for Software Quality Assurance Plans". Also, state the methods to be employed to assure that the developers comply with the requirements of that standard.

13. Records Collection, Maintenance and Retention

Identify the SQA documentation to be retained, state the methods and facilities to be used to assemble, safeguard and maintain this documentation, and designate the retention period.

14. Training

Identify the training activities necessary to meet the needs of the SQAP.

15. Risk Management

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.