

MODULE - III Software Quality Infrastructure Components - Procedures and Work Instructions – Staff Training and Certification - Software Configuration Management - Documentation Control.

Software quality infrastructure components

Infrastructure components

- Procedures and work instruction
- Quality support devices like templates and checklists
- Staff SQA training and certification activities
- Preventive and corrective actions
- Software configuration management
- Documentation and quality records control.

Procedures and work instruction

A *procedure* is “a particular way of accomplishing something or of acting” (Webster’s New College Dictionary). In other words, procedures, as transmitted in documents, are the detailed activities or processes to be performed according to a given method for the purpose of accomplishing a task.

Work instructions are used mainly in cases where a uniform method of performing the task throughout the organization is either impossible or undesirable. As a result, work instructions are specific to a team or department; they supplement procedures by providing explicit details that are suitable solely to the needs of one team, department, or unit. The software quality assurance procedures and work instructions of special interest to us are those that affect the quality of a software product, software maintenance or project management.

The need for procedures and work instructions

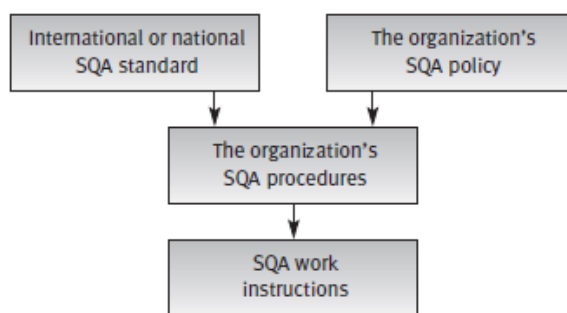


Figure 14.1: A conceptual hierarchy for development of procedures and work instructions

SQA procedures and work instructions aim at:

- Performance of tasks, processes or activities in the most effective and efficient way without deviating from quality requirements.
- Effective and efficient communication between the separate staffs involved in the development and maintenance of software systems. Uniformity in performance, achieved by conformity with procedures and work instructions, reduces the misunderstandings that lead to software errors.
- Simplified coordination between tasks and activities performed by the various bodies of the organization. Better coordination means fewer errors.

Procedures

Procedures supply all the details needed to carry out a task according to the prescribed method for fulfilling that task's function. These details can be viewed as responding to five issues, known as the *Five W's*,

The Five W's: issues resolved by procedures

- What activities have to be performed?
- HoW should each activity be performed?
- When should the activity be performed?
- Where should the activity be performed?
- Who should perform the activity?

Standardization – the application of a fixed format and structure – is the principle applied to all SQA procedures.

Fixed table of contents for procedures

- 1 Introduction *
- 2 Purpose
- 3 Terms and abbreviations *
- 4 Applicable documents

- 5 Method
- 6 Quality records and documentation
- 7 Reporting and follow-up *
- 8 Responsibility for implementation *
- 9 List of appendices *
- Appendices *
- * Sections included only if applicable

Work instructions and work instruction manuals

As mentioned above, work instructions deal with the application of procedures, adapted to the requirements of a specific project team, customer, or other relevant party. While general methodology is defined in a procedure, the precise details that allow its application to a specific project or unit are often laid out in a work procedure. In no case can work instructions contradict their parent procedure, although several instructions can be associated with any given procedure. This means that one can add, change or cancel work instructions without altering the respective procedure.

SQA work instructions subjects – examples

Departmental work instructions

- Audit process for new software development subcontractors (supplier candidates)
- Priorities for handling corrective maintenance tasks
- Annual evaluation of software development subcontractors
- On-the-job instructions and follow-up for new team members
- Design documentation templates and their application
- C++ (or other language) programming instructions

Project management work instructions

- Coordination and cooperation with the customer
- Weekly progress reporting by team leaders
- Special design report templates and their application in the project
- Follow-up of beta site reporting
- Monthly progress reporting to the customer
- Coordination of installation and customer's team instructions

Procedures and work instructions: preparation, implementation and updating

An "active" SQA procedures manual conceals numerous, often ongoing activities that guarantee the procedures' continued applicability: for instance, preparation of the procedures, their implementation and regular updating.

Preparation of new procedures

The initial steps taken in development of a new SQA procedures manual should deal with the conceptual and organizational frameworks that determine the menu of the proposed procedures and who will be responsible for their preparation, updating and approval. This framework is usually also formulated as a procedure (frequently called *the procedure of procedures*). The subsequent steps will, naturally, deal with specific procedures. A common approach to preparation of procedures is the appointment of an *ad hoc* committee of professionals working in the units involved, SQA unit members and experts in the respective topics to be dealt with. The committee pours over the proposed drafts until a satisfactory version is reached, and ceases its work only after the procedure is approved by the authorized person(s). An alternative approach to procedure manual preparation is dependence on consulting, where an outside expert is assigned the responsibility of preparing one procedure, some procedures or the complete manual.

Implementation of new or revised procedures

Approval of a new or revised procedure says little about the ease of that procedure's implementation, which is a separate and often difficult issue. The fact that members of a team or department were involved in the procedure's preparation helps convince their colleagues to abide by the new requirements but this, too, is often inadequate. Follow-up and individual instruction of those who lack or disregard the new procedure is mandatory for the procedure's integration within daily routines.

Updating procedures

The motivation to update existing procedures is based, among other things, on the following:

- Technological changes in development tools, hardware, communication equipment, etc.
- Changes in the organization's areas of activity
- User proposals for improvement
- Analysis of failures as well as successes
- Proposals for improvements initiated by internal audit reports
- Learning from the experience of other organizations

■ Experiences of the SQA team

Staff Training and Certification

The objectives of training and certification

- To develop the knowledge and skills new staff need to perform software development and maintenance tasks at an adequate level of efficiency and effectiveness. Such training facilitates integration of new team members.
- To assure conformity to the organization's standards for software products (documents and code) by transmitting style and structure procedures together with work instructions.
- To update the knowledge and skills of veteran staff in response to developments in the organization, and to assure efficient and effective performance of tasks as well as conformity to the organization's style and structure procedures and work instructions.
- To transmit knowledge of SQA procedures.
- To assure that candidates for key software development and maintenance positions are adequately qualified.

The training and certification process

The operation of a successful training and certification system demands that the following activities be regularly performed:

- Determine the professional knowledge requirements for each position
- Determine the professional training and updating needs
- Plan the professional training program
- Plan the professional updating program
- Define positions requiring certification
- Plan certification processes
- Deliver training, updating and certification programs
- Perform follow-up of trained and certified staff.

All these activities converge into an integrated process in which feedback from past activities and information about professional developments stimulate a cycle of continuous training, certification and adaptation to changing quality requirements.

Training and certification activities are meant to fill the needs of veteran staff and new employees. Comprehensive follow-up of the outcomes of current programs as well as keeping track of developments in the profession are required to make sure that programs are adequately up-to-date. A detailed discussion of each of these activities is presented in the next sections. The training and certification process is displayed in Figure 16.1

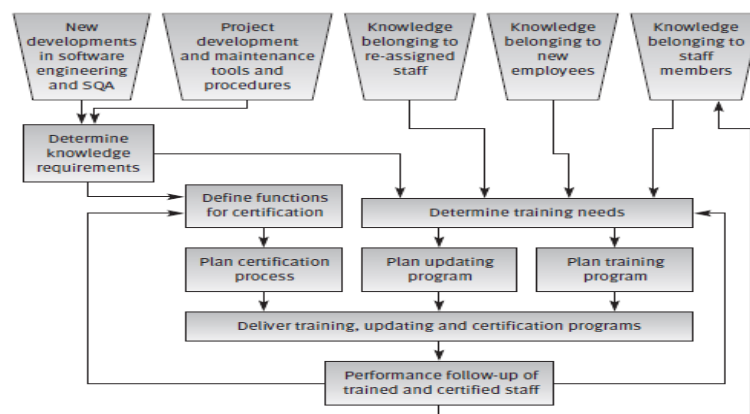


Figure 16.1: The training and certification process

Determining training and updating needs

Training and updating needs are determined by comparison of the staff's current knowledge with the updated knowledge requirements. The type of training is adapted to the needs of three distinct groups of staff:

- Training: for new employees, according to their designated assignment
- Retraining: for employees assigned to new positions or receiving new assignments
- Updating: for staff members as demanded by their position.

The need to update staff should be assessed regularly to facilitate planning of the required programs.

Finally, follow-up of staff performance in the wake of training and updating provides major input to be used in redefining training needs.

Planning training and updating programs

Practically speaking, two basic programs should be devised – one for software engineering topics and one for SQA topics.

Planning training and updating programs for software engineering

Topics

topics

The timing of many training and retraining activities cannot be determined in advance because new personnel are recruited and veteran staff are shifted often after relatively short notice. However, updating activities can be scheduled well ahead (the audience is known), with contents finalized close to the date of their implementation.

Planning training and updating programs for SQA topics

Training programs for SQA topics include training for new employees as well as updating for veteran staff members. The general characteristics of SQA training programs allow them to be organized periodically, every one or two months, and delivered to all new staff recruited in the interim. Typical SQA updating programs are carried out once a year or once every six months, depending on the pace of change. The SQA unit or others responsible for SQA issues in the organization usually prepare these training and updating programs.

Defining positions requiring certification

One of the procedures used to guarantee the suitability of candidates is certification. Examples of positions frequently requiring certification of their occupants are software development team leader, programming team leader, software testing team leader, software maintenance technician and internal quality auditor. The last two positions are particularly sensitive because their occupants' activities are usually performed by one staff member, acting alone, and subject to little close control or support by superiors.

A certification committee (or a designated senior staff member) defines the list of positions that require certification and whether the certification will be effective permanently or for a limited period. Renewal of limited period certification demands that staff members demonstrate up-to-date knowledge and skills according to the current certification requirements.

The list of positions that require certification naturally varies by firm or organization. Some use certification sparingly while others apply this tool on a large scale, even to standard programmers.

Planning the certification processes

The details of the certification process are unique to the organization; they reflect its special characteristics, areas of specialization, software development and maintenance tools, customers and so on. Because the process is geared toward the needs and decisions of specific organizations, internal certification cannot be automatically substituted by the general certification that is granted by professional societies and leading suppliers of development tools and network communication software or their equivalents. The certification process, in every detail and for every position, requires approval as defined in the certification procedure.

Typical certification requirements

For the individual undergoing certification, a typical certification process entails meeting some or even all of the following requirements:

- Professional education: academic or technical degrees and in some cases certification by a professional organization or by a leading commercial software producer
- Internal training courses
- Professional experience in the organization (may be partially or completely replaced by experience in other organizations)
- Assessment of achievements and ability as noted in periodic performance appraisals
- Evaluation by the candidate's direct superior (often by completion of a special questionnaire)
- Demonstration of knowledge and skills by means of a test or a project
- Mentor's supervision for a specified period of time.

Functions of the certification committee

The responsibilities of the certifying body include:

- To perform the certification process on the basis of requests made by individual applicants or units and grant certification to those who qualify
- To follow up certification activities (such as mentoring) carried out by others
- To update certification requirements in response to developments in the organization as well as the profession
- To revise the list of positions requiring certification.

Delivery of training and certification programs

Training and updating can cover topics such as software engineering, software quality assurance and management skills (within the framework of certification or for general information), all of which are

coordinated with the organization's or firm's needs. How training and updating are carried out varies accordingly. Courses can be transmitted in formats that range from short lectures and demonstrations, often lasting only half a day, to lengthy courses held over several weeks or months. These may be conducted in-house, by the organization's training unit, or externally, by vocational or academic institutions that prepare programs attuned to the organization's requirements. More about organizing and delivering training and certification programs can be found in the human resources management literature.

Follow-up subsequent to training and certification

The follow-up is necessary to provide feedback to the professional units. Such feedback indicates whether the training efforts were justified at the same time that it assures continuous improvement of training and certification activities.

The information provided by follow-up relates to:

- All training activities and certification procedures conducted – records of the performance of the participants in the program.
- Information about special cases of training activities that proved to be either highly successful or clearly unsuccessful in improving staff performance.
- Information about proven cases of failures of certified staff in the performance that point to clearly inadequate certification requirements.

The units responsible for training and certification should regularly perform follow-up using instruments such as the following.

- Collection of regular performance metrics – such as errors and productivity statistics, corrective maintenance statistics and resources invested – prepared by the respective units. For a discussion of software quality metrics in general and the specific issue of performance metrics.
- Questionnaires completed by staff members who received training, their superiors, customers and others.
- Analysis of outstanding achievements as well as failures.
- Specialized review of software products (documents and code) prepared by certified and trained employees.

The Corrective Action Board (CAB), based on follow-up subsequent to training and certification and other sources of information, may initiate training and/or updating activities subsequent to analysis of the cases presented to it.

Software Configuration Management

Software configuration items and software configuration

versions – definitions

■ Software configuration item (SCI) or configuration item (CI)

An approved unit of software code, a document or piece of hardware that is designed for configuration management and treated as a distinct entity in the software configuration management process.

■ SCI version

The approved state of an SCI at any given point of time during the development or maintenance process.

■ Software configuration version

An approved selected set of documented SCI versions that constitute a software system or document at a given point of time, where the activities to be performed are controlled by software configuration management procedures. The software configuration versions are released according to the cited procedures. A unit of software code, a document or piece of hardware is defined as an SCI if it is assumed that it may be needed for further development of the software system and/or its maintenance.

The SCIs are generally placed into four classes, as follows:

- Design documents
- Software code
- Data files, including files of test cases and test scripts
- Software development tools.

Common types of software configuration items

Design documents

- Software development plan (SDP)
- System requirements document
- Software requirements document (SRD)
- Interface design specifications
- Preliminary design document (PDD)
- Critical design document (CDD)
- Database description

- Software test plan (STP)
- Software test procedure (STPR)
- Software test report (STR)
- Software user manuals
- Software maintenance manuals
- Software installation plan (SIP)
- Software maintenance requests (including problem reports)
- Software change requests (SCRs) and software change orders (SCOs)
- Version description document (VDD)

Software code

- Source code
- Object code
- Prototype software

Data files

- Test cases and test scripts
- Parameters, codes, etc.

Software development tools

(the versions applied in the development and maintenance stages)

- Compilers and debuggers
- Application generators
- CASE tools

Software configuration management – definition

An SQA component responsible for applying (computerized and non-computerized) technical tools and administrative procedures that enable completion of the tasks required to maintain SCIs and software configuration versions. Software configuration management – tasks and organization

The tasks of software configuration management

The tasks of software configuration management may be classified into four groups:

- Control software change
- Release of SCI and software configuration versions
- Provision of SCM information services
- Verification of compliance to SCM procedures

The software configuration authority

It is practically self-evident that an authority to oversee implementation of the above tasks is vital in software developing and/or maintaining organizations. SCM procedures specify who is responsible for SCM issues. This responsibility is usually assigned to a senior professional or a committee dedicated to SCM issues. In many organizations, software change control is dealt with by a special committee set up for such matters, commonly called the software change control authority (SCCA) or the software change control board (SCCB). This body is frequently called the change control authority (CCA) or the change control board (CCB). During the development stage, the project manager may be charged with the authority to carry out SCM responsibilities.

Software change control

Software change management controls the process of introducing changes mainly by doing the following:

- Examining change requests and approving implementation of appropriate requests.
- Assuring the quality of each new version of software configuration before it becomes operational.

Release of software configuration versions

The need to release a new software configuration version usually stems from one or more of the following conditions:

- Defective SCIs
- Special features demanded by new customers
- The team's initiatives to introduce SCI improvements.

The following issues, all of which are part of the process of software configuration version release,

- Types of software configuration releases
- Software configuration management plans (SCMPs)
- Software configuration evolution models
- Documentation of software configuration versions.

Types of software configuration releases

Among software configuration releases, baseline versions, intermediate versions and revisions are considered to be the three main types of release.

Baseline versions

Baseline software configuration versions are planned early, during a system's development or operating stage. As part of the process, they are reviewed, tested and approved, as are their SCIs. Baseline versions serve as milestones in the software system's life cycle, and represent the foundations for further system development.

Intermediate versions

When problems arise that require immediate attention – such as the need to correct defects identified in an important SCI, or perform immediate adaptations as defined in a contract with a new customer – an intermediate version of the software is often prepared. Usually, intermediate versions serve only a portion of a firm's customers, and then for a limited period, until replaced by a new baseline version. Naturally, we can expect that these versions will not receive the attention and investment of efforts usually devoted to the release of baseline versions. An intermediate software configuration version can thus serve as a "pilot" or springboard to the next baseline version.

Revisions

Revisions introduce minor changes and corrections to a given software configuration version. In some cases, several successive revisions are released before a new baseline version is released.

Numeration conventions for identification of SCI and software versions

Numeration conventions have been formulated to identify SCIs; the most commonly used is decimal numeration, which indicates the successive version and revision numbers and is registered accordingly. For example, an SCI design document captioned DD-7 may have several versions and revisions, identified as DD-7 Ver.1.0, DD-7 Ver.1.1, DD-7 Ver.2.0, DD-7 Ver.3.0, DD-7 Ver.3.1, DD-7 Ver.3.2, etc., where the first number represents the version and the second the revision. Put simply, an SCI is identified by its name in combination with its version and revision numbers.

Software configuration management plans (SCMPs)

The main objective of a software configuration management plan (SCMP) is to plan ahead the schedule of baseline version releases and the required resources to carry out all the activities required for the software configuration releases. An additional objective of the SCMP is to enable one to follow up the progress of activities involved in software version release. SCMPs are required during the development stage as well as the operation (maintenance) stage. Accordingly, an SCMP usually includes:

- An overview of the software development project or existing software system.
- A list of scheduled baseline version releases.
- A list of SCIs (documents, code, etc.) to be included in each version.
- A table identifying the relationship of software development project plans and maintenance plans to scheduled releases of new SCIs or SCI versions.
- A list of assumptions about the resources required to perform the various activities required by the SCMP.
- Estimates of the human resources and budget needed to perform the SCMP.

SCMP for the development stage

Based on the project plan, the SCMP sets the release dates of baseline versions, which usually coincide with the conclusion of one or more of the following three events: the design stage, the coding stage and the system test stage. Quite commonly, these plans represent a segment of the entire system's development plans, prepared at a project's initiation. External participants in the project are required to comply with the SCMP or to suggest an alternative SCMP that is appropriate for their part of the project, contingent on its acceptance by the project manager.

All the instructions and procedures necessary for performing SCM tasks at this stage are documented in the SCMP. The project manager is usually the person responsible for carrying out these tasks.

SCMP for the operation (maintenance) stage

During the operation (maintenance) stage, further releases of software baseline versions are required in order to introduce improved software versions released after accumulation of SCI changes made during regular customer use. The plan generally schedules new baseline releases periodically, either annually, semi-annually, or according to the anticipated number of accumulated changes in SCIs.

Software configuration evolution models

Successive development or evolution of a software system's configuration versions should be undertaken according to a route that is planned in advance by the system's developer. The choice of routes depends on the system's characteristics, the customer population and the firm's intentions regarding the system's market. Two fundamental software configuration evolution models – the line model and the tree model – are generally applied. We discuss these next.

■ The linear evolution model

According to the linear model, only one unique software system's configuration version serves all customers at any given time. The model is also applied to popular software packages, which tend to be uniform in structure, where the need to meet a wide range of maintenance demands for a single version is a great advantage.

■ The tree evolution model

According to this model, several parallel versions of the software are developed to serve the needs of different customers simultaneously throughout the system's life cycle. Tree models are typically applied in firmware configuration versions, where each branch serves a different product or product line. Documentation of software configuration versions Within the framework of software configuration management, the project manager must see to it that all documentation tasks are properly performed. Two of these tasks – documentation of SCI versions and documentation of software configuration releases (versions and revisions) – represent the two main types of tasks to be completed.

Software configuration release documentation –

VDD template

Identification and installations

- Release version and revision number
- Date of the new version's release
- List of installations where the release was entered (site, date, name of technician who installed the version), if applicable

Configuration of the released version

- List of SCIs in the released version, including identification of each SCI version
- List of hardware configuration items required for operating the specified version, including specification of each hardware configuration item
- List of interfacing software systems (including version) and hardware systems (including model)
- Installation instructions for the new release

Changes in the new version

- Previous software configuration version
- List of SCIs that have been changed, new SCIs introduced for the first time, and deleted SCIs
- Short description of introduced changes
- Operational and other implications of changes introduced in the new release

Further development issues

- List of software system problems that have not been solved in the new version
- List of SCRs and proposals for development of the software system for which implementation of development was delayed

Software configuration release documentation –

VDD template

Identification and installations

- Release version and revision number
- Date of the new version's release
- List of installations where the release was entered (site, date, name of technician who installed the version), if applicable

Configuration of the released version

- List of SCIs in the released version, including identification of each SCI version
- List of hardware configuration items required for operating the specified version, including specification of each hardware configuration item
- List of interfacing software systems (including version) and hardware systems (including model)
- Installation instructions for the new release

Changes in the new version

- Previous software configuration version
- List of SCIs that have been changed, new SCIs introduced for the first time, and deleted SCIs
- Short description of introduced changes
- Operational and other implications of changes introduced in the new release

Further development issues

- List of software system problems that have not been solved in the new version
- List of SC.Rs and proposals for development of the software system for which implementation of development was delayed

Provision of SCM information services

The SCM is required to provide information to professionals, mainly developers, maintenance teams and customer representatives, who have requested that changes be introduced in a software system. The information provided may be classified into information related to software change control and information dealing with SCI and software configuration versions.

Information related to software change control

- Change request status information
- Change order progress information

Information about SCIs and software configuration versions

- Accurate copies of SCI versions (code SCIs, document SCIs, etc.) and entire software configuration versions.
- Full reports of changes introduced between successive releases (versions and/or revisions) of code SCIs as well as between successive releases of other types of SCIs.
- Copies of SCI version documentation and software configuration version documentation (VDDs).
- Detailed version and revision history for SCIs and software configurations for any specific SCI or software system.
- Progress information about planned versions and releases (usually included in the SCMP).
- Information correlated about versions installed at a given site and about the site itself.
- List of sites where a given software configuration version is installed. Software configuration management audits

SCM audits check whether and how these tasks were performed for samples of change requests, SCIs, and software configuration versions. SCM audits may be also performed for a sample of planned releases, as specified in the SCMP.

The following is a list of typical bits of control information that SCM audits are meant to discover and transmit to management:

- Percentage of unapproved changes introduced in the system during development or operation.
- Percentage of SCOs not carried out according to instructions and not fully complying with procedures.
- Percentage of design reviews and software tests of changed SCIs that have not been performed according to the relevant procedures.
- Percentage of SCOs that have been completed on schedule.
- Percentages of cases where SCIs affected by changes have not been checked, with some necessary changes not implemented.
- Percentages of properly documented new SCIs and software configuration versions.
- Percentage of properly documented installations of new software configuration versions.
- Percentage of cases of failure to transmit all version-related information to the customer.
- Number of cases recorded annually where the SCI work coordination mechanisms failed (i.e., did not prevent different teams from simultaneously introducing changes in the same SCI).

Computerized tools for managing software configuration

Computerized SCM tools have been on the market for many years. These computerized tools differ in their level of comprehensiveness, flexibility of application and ease of use. The computerized SCM tools also operate the mechanisms coordinating the work on an SCI's changes and prevent different teams from simultaneously introducing changes in the same SCI. An additional benefit of the use of a computerized SCM system is the high security level it is able to provide:

- It secures the code version and documentation files versions by protecting them from any changes, deletions and other damages.
- It activates back-up procedures required for safe SCM file storage.

Documentation Control

Software development and maintenance processes involve production and use of a multitude of documents; some are vital immediately while others may become vital for software quality assurance over the life cycle of the system.

Special procedures for documentation control (usually called *documentation procedures*, *documentation control procedures* or *control of documents procedures*) are therefore introduced to indicate which documents are indeed expected to be vital at some point and to assure their appropriate preparation and availability.

Documents that display these characteristics and that are treated according to these procedures are called *controlled documents*.

One type of controlled document – *quality records* – is aimed mainly to provide evidence that the development and maintenance processes were performed in conformity to requirements and that the software quality system is operating fully and effectively.

Controlled document

A document that is currently vital or may become vital for the development and maintenance of software systems as well as for the management of current and future relationships with the customer. Hence, its preparation, storage, retrieval and disposal are controlled by documentation procedures. The main objectives for managing controlled documents are:

- To assure the quality of the document.
- To assure its technical completeness and compliance with document structure procedures and instructions (use of templates, proper signing, etc.).
- To assure the future availability of documents that may be required for software system maintenance, further development, or responses to the customer's (tentative) future complaints.
- To support investigation of software failure causes and to assign responsibility as part of corrective and other actions.

Quality record

A quality record is a special type of controlled document. It is a customer-targeted document that may be required to demonstrate full compliance with customer requirements and effective operation of the software quality assurance system throughout the development and maintenance processes.

Typical controlled documents (including quality records)

Pre-project documents

- Contract review report
- Contract negotiation meeting minutes
- Software development contract
- Software maintenance contract
- Software development subcontracting contract
- Software development plan

Project life cycle documents

- System requirements document
- Software requirements document
- Preliminary design document
- Critical design document
- Database description
- Software test plan
- Design review report
- Follow-up records of design review action items
- Software test procedure
- Software test report
- Software user manuals
- Software maintenance manuals
- Software installation plan
- Version description document
- Software change requests
- Software change orders
- Software maintenance requests
- Maintenance services reports
- Records of subcontractor evaluations

SQA infrastructure documents

- SQA procedures
- Template library
- SQA forms library
- CAB meeting minutes

Software quality management documents

- Progress reports
- Software metrics reports

SQA system audit documents

- Management review report
- Minutes of management review

meeting

- Internal quality audit report
- External SQA certification audit report

Customer documents

- Software project tender documents
- Customer's software change Requests

These document types are produced during the implementation of a variety of SQA processes, to mention but a few:

- Contract and negotiation process
- Development process
- Software change process
- Maintenance services
- Software quality metrics
- Internal quality audits.

Documentation control procedures

The SQA tools that regulate the handling of a controlled document from its creation to its final disposal are called *documentation control procedures*.

Typical components of documentation control procedures

- Definition of the list of the document types and updates to be controlled
- Document preparation requirements
- Document approval requirements
- Document storage and retrieval requirements, including controlled storage of document versions, revisions and disposal

Two documentation control tasks – namely, storage and retrieval – are included among the organization's software configuration management procedures and performed with a variety of software configuration management tools.

The components of the documentation control procedure, namely:

- The controlled documents list
- Controlled document preparation
- Controlled document approval issues
- Issues of controlled document storage and retrieval issues.

The controlled documents list

The key to management of controlled documents (including quality records) is the controlled document types list. Proper construction of the list is based on the establishment of an authority to implement the concept, whether embodied in a person or a committee.

Specifically, this authority is responsible for:

- Deciding which document type is to be categorized as a controlled document and which controlled document types are to be classified as quality records.
- Deciding whether the level of control is adequate for each document type categorized as a controlled document.
- Following up of compliance with the controlled document types list. This subject can be incorporated in the internal quality audits plan .
- Analyzing follow-up findings and initiating the required updates, changes, removals and additions to the controlled documents types list.

Controlled document preparation

The documentation requirements involved in the creation of a new document or the revision of an existing document focus on completeness, improved readability and availability. These requirements are realized in the documents:

- Structure
- Identification method
- Standard orientation and reference information.

The document's **structure** may be free or defined by a template.

An **identification method** is devised to provide each document, version and revision with a unique identity. The method usually entails notation of (a) the software system or product name or number, (b) the document (type) code and (c) the version and revision number. The method can vary for different types of documents.

The document's **orientation and reference information** may be required as well. Orientation and reference information support future access of required documents by supplying information about the content of the document and its suitability to the needs of the future user.

Issues of controlled document storage and retrieval

Requirements pertaining to controlled storage and retrieval of documents are set mainly to assure a document's security and its continued availability. The same requirements should apply to paper documents as well as electronic and other media. They refer to:

- Document storage *per se*
- Circulation and retrieval of documents
- Document security, including document disposal.

Document storage requirements apply to (1) the number of copies to be stored, (2) the unit responsible for storage of each copy, and (3) the storage medium. Storage on electronic media is usually much more efficient and more economical than storage on paper.

Circulation and retrieval of documents requirements refer to (1) instructions for circulating a new document, on time, to the designated recipients, and (2) efficient and accurate retrieval of copies, in full compliance with security restrictions. The procedures should apply to the circulation of paper documents as well as use of e-mail, intranet and the Internet.

Document security, including document disposal requirements, (1) provide restricted access to document types, (2) prevent unauthorized changes to stored documents, (3) provide back-up for stored paper as well as electronic files, and (4) determine the storage period.