

## ELECTIVE III: SOFTWARE QUALITY ASSURANCE

SEMESTER: IV

2018-19 ONWARDS

SUBJECT CODE: 18MIT41E

UNIT IV:

Procedures and work instructions: the need – procedure manuals – work instruction manuals – preparation, implementation and updating. Supporting quality devices: templates – checklists. Staff training and certification: “3S” development team- the objectives – training and certification process – determining training and updating needs – defining positions requiring certification – planning the certification processes – delivery of training and certification programs.

### Procedures and work instructions

#### The need for procedures and work instructions

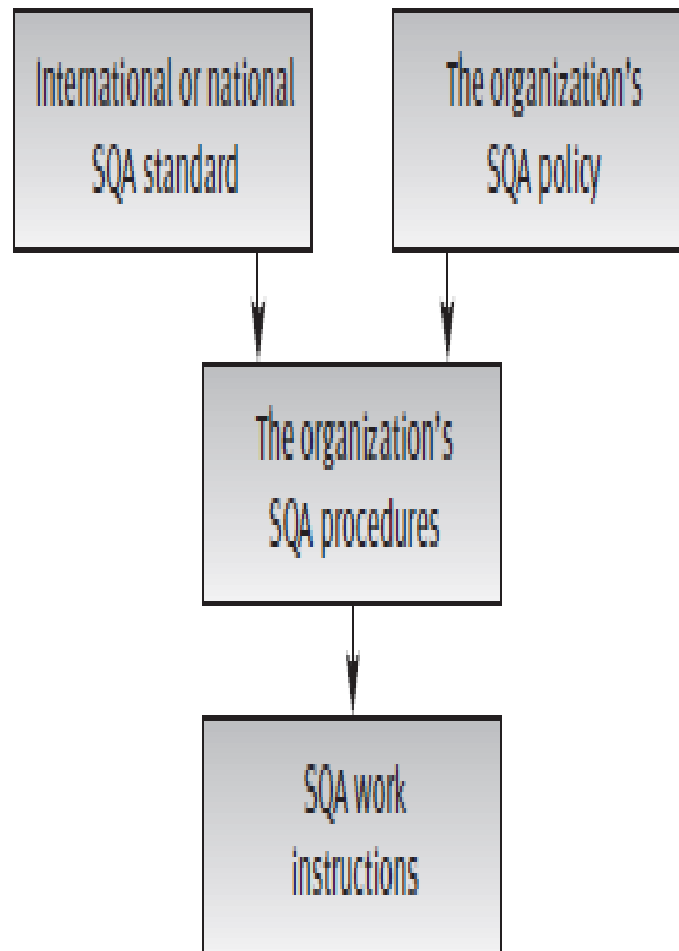
SQA procedures and work instructions aim at:

- 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.

## Hierarchy for the development of procedures and work instructions



## Procedures and procedures manuals

### **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?

### **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

### ***The procedures manual***

- The types of software development and maintenance activities carried out by the organization
- The range of activities belonging to each activity type
- The range of customers (e.g., internal, customers of custom-made software,

COTS software customers) and suppliers (e.g., self-development and maintenance, subcontractors, suppliers of COTS software and reused software modules)

■ The conceptions governing the choice of method applied by the organization to achieve desired SQA objectives.

## SQA procedure manuals – Table Of Contents

ISO 9000.3 – table of contents	SQA procedures manual – table of contents
4.1 Managerial responsibility	1.1 The company's SQA policy
	1.2 Management quality review
4.2 Quality system	2.1 The SQA organization
	2.2 Procedures and work instructions – preparation, approval and distribution
	2.3 The annual quality planning
4.3 Contract review	3.1 Contract review
4.4 Design control	4.1 Development and quality plans
	4.2 Quality assurance of the design
4.5 Document and data control	5.1 Document control
4.6 Purchasing	6.1 Subcontractors and suppliers file management
	6.2 Pre-contract review for subcontractor proposal
	6.3 Acceptance tests for subcontracted software
4.7 Control of customer-supplied products	7.1 Acceptance tests for customer-supplied software
4.8 Product identification and traceability	8.1 Configuration management
4.9 Process control	9.1 Software development process
4.10 Inspection and testing	10.1 Unit tests and integration tests
	10.2 Software system tests
	10.3 Customer's acceptance tests
4.11 Control of inspection, measuring and test equipment	— Not applicable
4.12 Inspection and test status	12.1 Progress control for software development project
4.13 Control of non-conforming product	13.1 Control of design and code corrections
4.14 Corrective and preventive actions	14.1 Corrective and preventive actions
4.15 Handling, storage, packaging, preservation and delivery	15.1 Installation and delivery
4.16 Control of quality records	16.1 Control of quality records

## ISO 9000.3 – table of contents

4.17 Internal quality audits

4.18 Training

4.19 Servicing

4.20 Statistical techniques

## SQA procedures manual – table of contents

17.1 Internal quality audits

18.1 Training and certification of employees

19.1 Maintenance plan

19.2 Change requests management

19.3 Dealing with customers' complaints

20.1 Quality metrics

20.2 Quality assurance costs

---

## Work instructions and work instruction manuals

### **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

### *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.

### *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. In many cases, distribution of the material in printed or e-mail form and instruction of the team or unit are insufficient to assure full or nearly full conformity.

### *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

## Supporting quality devices

### TEMPLATE:

In other areas of work, a *template* is “a gauge, pattern or mold (as a thin plate or board) used as a guide to the form of a piece being made” (Webster’s New College Dictionary). When applied to software engineering, the term template refers to a format (especially tables of contents) created by units or organizations, to be applied when compiling a report or some other type of document.

Three examples of templates are

1. Software test plan (STP)
2. Software test description (STD)
3. Software test report (STR).

### **The contribution of templates to software quality**

Template use is quite advantageous to development teams and to review teams. For development teams, template use:

■ **Facilitates the process of preparing documents** by saving the time and energy required to elaborate the report's structure. Most organizations allow templates to be copied from a SQA public file or downloaded from the organization's intranet files, which even saves keying the table of contents to the new document.

■ **Ensures that documents prepared by the developer are more complete** as all the subjects to be included in the document have already been defined and repeatedly reviewed by numerous professionals over the course of the template's use. Common errors, such as overlooking a topic, are less likely to occur.

■ **Provides for easier integration of new team members** through familiarity. The document's standard structure, prepared according to templates that may be known to the new member from previous work in another of the organization's units or teams, makes finding information much easier. It also smoothes ongoing document preparation in cases where parts of the document have been prepared by another team member who may or may not have left.

■ **Facilitates review of documents** by eliminating the need to study a document's structure and confirm its completeness, if the document is based on the appropriate template. It also simplifies review of the completed document as its structure is standard and reviewers are familiar with its expected contents (chapters, sections and

appendices). As a result of this consistency, the review is expected to be more thorough yet less time-consuming.

For software maintenance teams, template use:

- **Enables easier location of the information** required for performing maintenance tasks.

## **The organizational framework for preparing, implementing and updating templates**

### *Preparation of new templates*

Development of a template infrastructure naturally centers on the work of a group of professionals devoted to the task. This group (or committee) should include senior staff who represent the various software development lines, the department's chief software engineer and SQA unit members. Informal developers of "template services" should likewise be encouraged to join the group.

The most common information sources used in preparing a template are as follows:

- Informal templates already in use in the organization
- Template examples found in professional publications
- Templates used by similar organizations

### *Application of templates*

Several fundamental decisions are involved in the implementation of new or updated templates:

- What channels should be used for advertising the templates?
- How should the templates be made available to the organization's internal "consumers"?
- Which templates will be compulsory and how can their application be enforced?

All professional internal means of communication can be used for advertising templates internally within the organization: leaflets, e-mail, SQA intranet as well as short presentations at meetings. One of the most efficient methods of making templates available to the organization is the internal net (**intranet**), to be preferred to any paper-based route. Distribution through the internal net ensures user choice of the updated version of the template needed and, at the same time, saves keying in (required for paper-based templates) of the document's table of contents.

### ***Updating templates***

The decision to update an existing template may be considered a reactive measure, stemming from any of the following:

- User proposals and suggestions
- Changes in the organization's areas of activity
- Proposals initiated by design review and inspection teams based on their review of documents prepared according to the templates
- Analysis of failures as well as successes

- Other organizations' experience
- SQA team initiatives.

The process of updating templates is quite similar to that of template preparation.

## Checklists

The checklist used by software developers refers to the list of items specially constructed for each type of document, or a menu of preparations to be completed prior to performing an activity (e.g., installing a software package at the customer site). Checklists are planned to be comprehensive if not complete. Usually, checklist use tends to be considered an optional infrastructure tool, depending mainly on the list's professional attributes, user acquaintance with the list and availability.

Proposal draft reviews – Subjects checklist

Contract draft review – Subjects checklist

## Goldenbug Ltd

## Checklist for requirement specification report

Project name: \_\_\_\_\_

The reviewed document: \_\_\_\_\_ Version: \_\_\_\_\_

Item no.	Subject	Yes	No	N.A.*	Comments
<b>1</b>	<b>The document</b>				
1.1	Prepared according to configuration management requirements				
1.2	Structure conforms to the relevant template				
1.3	Reviewed document is complete				
1.4	Proper references to former documents, standards, etc.				
<b>2</b>	<b>Specifying the requirements</b>				
2.1	Required functions were properly defined and clearly and fully phrased				
2.2	Designed inputs conform with required outputs				
2.3	Software requirement specifications conform with product requirements				
2.4	Required interfaces with external software packages and computerized equipment are fully defined and clearly phrased				
2.5	GUI interfaces are fully defined and clearly phrased				
2.6	Performance requirements – response time, input flow capacity, storage capacity – are correctly defined and fully and clearly phrased				
2.7	All error situations and required system reactions are correctly defined and fully and clearly phrased				
2.8	Data interfaces with other existing or planned software package or products components are correctly defined and fully and clearly phrased				
2.9	Procedures to test fulfillment of the specified requirement are correctly and fully defined and clearly phrased				
<b>3</b>	<b>Project feasibility</b>				
3.1	Are the specified requirements feasible considering the project's resources, budget and timetable?				
3.2	Are the specified performance requirements (see 2.6) feasible considering the constraints imposed by other system components and by external systems interfaced with the system?				

Comments:

\*N.A. = Not applicable

Signed: Name: \_\_\_\_\_ Date: \_\_\_\_\_ Signature: \_\_\_\_\_

## The contribution of checklists to software quality

- **Helps developers carrying out self-checks of documents or software code**

- **Assists developers in their preparations for tasks** such as installation of software at customer sites, performance of quality audits at subcontractors' sites or signing contracts with suppliers of reused software modules

The advantages to review teams are:

- **Assures completeness of document reviews by review team members** as all the relevant review items appear on the list.

- **Facilitates improves efficiency of review sessions** as the subjects and order of discussion are defined and well known in advance.

## The organizational framework for preparing, implementing and updating checklists

### *Preparation of new checklists*

- Informal checklists already in use in the organization

- Checklist examples found in books and other professional publications

- Checklists used by similar organizations.

The process of preparing a new checklist is similar to that for templates.



### *Promotion of checklist use*

As the use of checklists is rarely mandatory, promotion of their use is based on advertising and guaranteed availability. All internal channels of communication can be used for publicizing the checklists: leaflets, e-mail, SQA intranet as well as professional meetings. The internal net remains, however, the preferred and most efficient method for making checklists available to the organization's internal "consumers".

### *Updating checklists*

Like templates and procedures, initiatives to update an existing checklist generally flow from the following sources:

- User proposals and suggestions
- Changes in technology, areas of activity and clientele
- Proposals initiated by design review and inspection teams emanating from document reviews
- Analysis of failures as well as successes
- Other organizations' experience
- SQA team initiatives.

The process of updating checklists is quite similar to their preparation.

## Staff training and certification

### Introduction: surprises for the “3S” development team

Team 7 of “3S – Sahara Software Specialists” started a new project for Apollo Ltd three weeks late because of delays in completion of the previous project.

At this point, seven weeks behind schedule, the team leader concluded that the “**super saver strategy**” applied to training, as well as the “super short-cut procedure” implemented in recruitment, instruction and follow-up, had proven to be quite costly.

### **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.

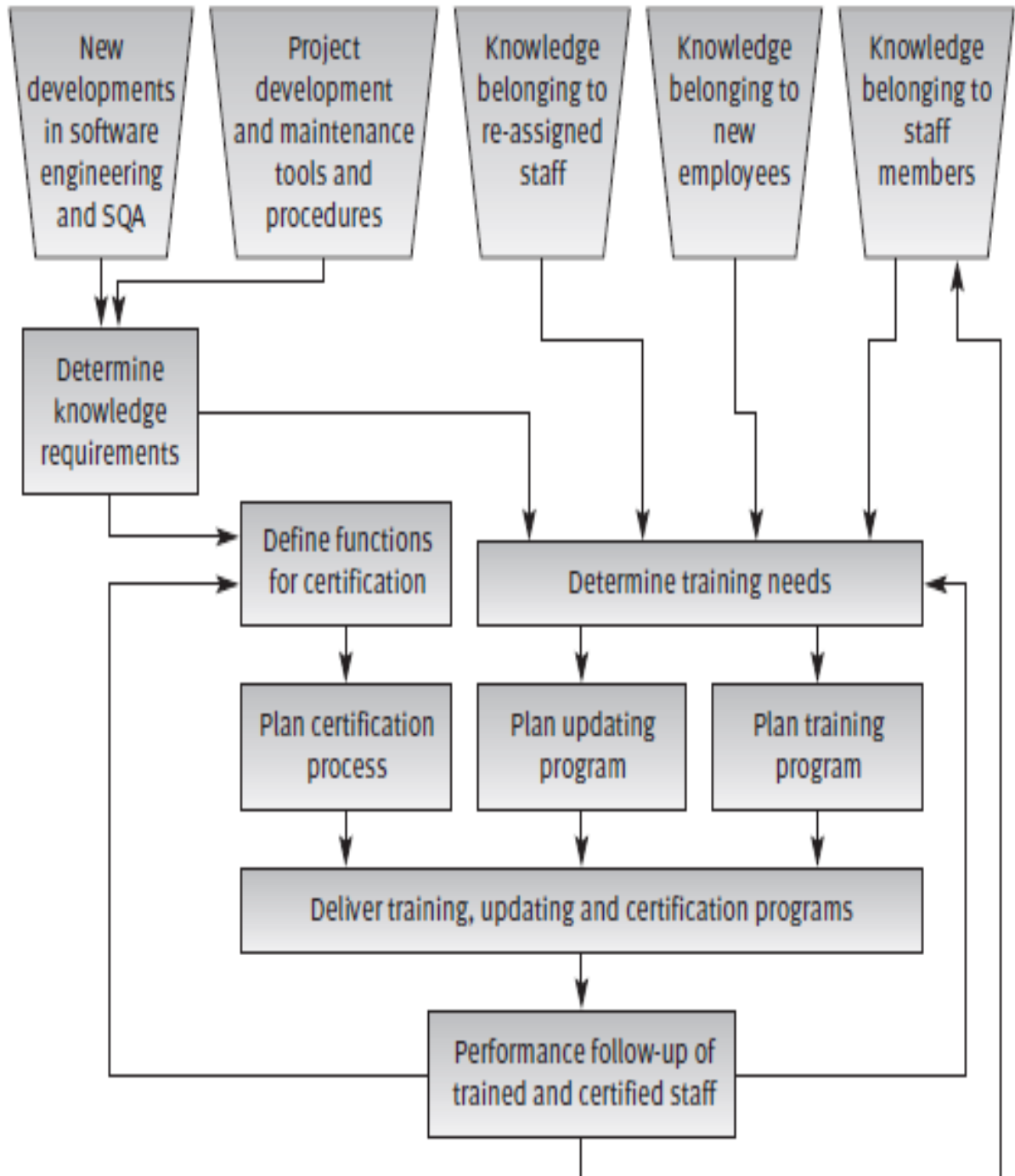
- To transmit knowledge of SQA procedures.
- To assure that candidates for key software development and maintenance positions are adequately qualified.

### The training and certification process

- 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.

Training and certification activities are meant to fill the needs of veteran staff and new employees.

# 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.

## Defining positions requiring certification

It is commonly accepted that assignment of personnel to key positions in software development and maintenance organizations requires extreme care.

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. Considering the volatility of the profession, this list should be revised periodically. 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

Certification is intended to provide a framework for the thorough investigation of a candidate's qualifications and a demonstration of his or her professional knowledge and skills. 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*

Similar to the pattern recommended for training and retraining programs, the person or committee members responsible for certification are usually senior software development and maintenance staff. 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.