

UNIT 3

DOCUMENTATION OF SYSTEMS

Lesson Structure

- 3.0 Objectives
- 3.1 Introduction
- 3.2 Concepts and Process of Documentation
- 3.3 Types of Documentation
 - 3.3.1 Program Documentation
 - 3.3.2 System Documentation
 - 3.3.3 Operations Documentation
 - 3.3.4 User Documentation
- 3.4 System Requirements Specification
 - 3.4.1 Characteristics of a SRS
 - 3.4.2 Rules for Specifying Software Requirement
 - 3.4.3 Structure of a Typical SRS.
- 3.5 Different Standards for Documentation
- 3.6 Documentation and quality of software
- 3.7 Good Practices for Documentation
- 3.8 Summary
- 3.9 Questions for Exercise
- 3.10 Further Readings

3.0 OBJECTIVES

After going through this unit, you should be able to:

- learn the concept of documentation;
- understand the importance of documentation;
- learn about various documents and process of documentation;
- understand application of various standards of documentation processes;
- differentiate between various documentation processes;
- understand the good practices for documentation process.

3.1 INTRODUCTION

Documentation is one of the systems which are used to communicate, instruct and record the information for any reference or operational purpose. They are very useful for representing the formal flow of the present system. With the help of documentation it is very easy to track the flow of the system's progress and the working of the system can be explained very easily.

It helps to provide the clear description of the work done so far. It is essential that the documents prepared must be updated on regular basis. With appropriate and good documentation it is very easy to understand the aspects of the system how it will work for the company where the system is to be installed. After the system is installed, and if in case the system is not working properly it will be very easy for the administrator to understand the flow of data in the system with documentation which will help him/ her to correct the flaws and get the system working in no time.

3.2 CONCEPTS AND PROCESS OF DOCUMENTATION

Documentation may be defined as the process of communicating about the system.

The person who is responsible for this communication is called documenter. It may be noted that documenter is not responsible for the accuracy of the information, and his job is just to communicate or transfer the information.

The ISO standard ISO/IEC 12207:1995 describes documentation “as a supporting activity to record information produced by a system development life cycle process.”

Documentation is needed because of the following reason given below:

- a means for transfer of knowledge and details about description of the system
- to communicate among different teams of the software project;
- to help corporate audits and other requirements of the organization;
- to meet regulatory demand;
- needed for IT infrastructure management and maintenance; and
- needed for migration to a new software platform.

Document communicates the details about the system targeted at different audience. It explains the system. The ever-increasing complexity of information system requires emphasis on well-established system of documentation. Every information system should be delivered along with an accurate and understandable document to those who will use the software.

Traditionally, documentation was done after the development of the software is completed. However, as the software development process is becoming complex and involved, documentation has become an integral part of each system development

process. Documentation is now carried out at every stage as a part of development process. We will also discuss how documentation affects quality of the software later in this section.

When the process of documentation is undertaken as a separate process, it requires planning in its own right. Design and development activities of software depend on a certain base document. Documentation is to be carried out before actually implementing the design. In such a case, any flaw in design identified can be changed in the document thereby saving cost and time during implementation.

If documentation is being developed for an existing software, then documentation is done along side the software development process.

The Process of Documentation

The following are various steps involved in the process of documentation:

Collection of source material: The very first step of any documentation process is to acquire the required source material for preparation of document. The material is collected including specifications, formats, screen layouts and report layouts. A copy of the operational software is helpful for preparing the documentation for user.

Documentation Plan: The documenter is responsible for preparation of a documentation plan, which specifies the details of the work to be carried out to prepare the document.

Review of Plan: The plan as set out in the process above is reviewed to see that material acquired is correct and complete.

Creation of Document: The document is prepared with the help of document generator.

Testing of Document: The document created is tested for usability as required by the target audience.

Maintain Document: Once the document is created and distributed, it must be kept up to date with new version of the software product. It must be ensured that the latest

document is available to the user of the software.

3.3 TYPES OF DOCUMENTATION

Any software project is associated with a large number of documents depending on the complexity of the project. Documentation that are associated with system development has a number of requirements. They are used by different types of audience in different ways as follows:

- They act as a means of communication between the members of development team
- Documents are used by maintenance engineer
- Documents are used by the user for operation of the software
- Documents are used by system administrator to administer the system.

When it comes to System Design, there are following four main documentations:

1. Program documentation
2. System documentation
3. Operations documentation
4. User documentation

3.3.1 Program Documentation

- It describes inputs, outputs, and processing logic for all the program modules.
- The program documentation process starts in the system analysis phase and continues during implementation.
- This documentation guides programmers, who construct modules that are well supported by internal and external comments and descriptions that can be understood and maintained easily.

3.3.2 System Documentation

System documentation serves as the technical specifications for the IS and how the objectives of the IS are accomplished. Users, managers and IS owners need never reference system documentation.

It includes the following information:

- System documentation provides the basis for understanding the technical aspects of the IS when modifications are made.
- It describes each program within the IS and the entire IS itself.
- It describes the system's functions, the way they are implemented, each program's

purpose within the entire IS with respect to the order of execution, information passed to and from programs, and overall system flow.

- It includes data dictionary entries, data flow diagrams, object models, screen layouts, source documents, and the systems request that initiated the project
- Most of the system documentation is prepared during the system analysis and system design phases.
- During systems implementation, an analyst must review system documentation to verify that it is complete, accurate, and up-to-date, and including any changes made during the implementation process.

3.3.3 Operations Documentation

Operations documentation contains all the information needed for processing and distributing online and printed output. Operations documentation should be clear, concise, and available online if possible.

It includes the following information:

- Program, systems analyst, programmer, and system identification.
- Scheduling information for printed output, such as report, execution frequency, and deadlines.
- Input files, their source, output files, and their destinations.
- E-mail and report distribution lists.
- Special forms required, including online forms.
- Error and informational messages to operators and restart procedures.
- Special instructions, such as security requirements.

3.3.4 User Documentation

It includes instructions and information to the users who will interact with the system. For example, user manuals, help guides, and tutorials. User documentation is valuable in training users and for reference purpose. It must be clear, understandable, and readily accessible to users at all levels.

The users, system owners, analysts, and programmers, all put combined efforts to develop a user's guide.

User documentation should include:

- A system overview that clearly describes all major system features, capabilities, and limitations.
- Description of source document content, preparation, processing, and, samples.

- Overview of menu and data entry screen options, contents, and processing instructions.
- Examples of reports that are produced regularly or available at the user's request, including samples.
- Security and audit trail information.
- Explanation of responsibility for specific input, output, or processing requirements.
- Procedures for requesting changes and reporting problems.
- Examples of exceptions and error situations.
- Frequently asked questions (FAQs).
- Explanation of how to get help and procedures for updating the user manual

3.4 SYSTEM REQUIREMENTS SPECIFICATION

System Requirements Specification is a set of complete and precisely stated properties along with the constraints of the system that the software must satisfy. A well designed software requirements specification establishes boundaries and solutions of system to develop useful software. All tasks, however minute, should not be underestimated and must form part of the documentation.

Requirements of SRS: The SRS should specify only the external system behaviour and not the internal details. It also specifies any constraints imposed on implementation. A good SRS is flexible to change and acts as a reference tool for system developer, administrator and maintainer.

3.4.1 Characteristics of a System Requirements Specification (SRS)

1. All the requirements must be stated unambiguously. Every requirement stated has only one interpretation. Every characteristic of the final product must be described using a single and unique term.
2. It should be complete. The definition should include all functions and constraints intended by the system user. In addition to requirements of the system as specified by the user, it must conform to any standard that applies to it.
3. The requirements should be realistic and achievable with current technology.

There is no point in specifying requirements which are unrealisable using existing hardware and software technology. It may be acceptable to anticipate some

hardware developments, but developments in software technology are much less predictable.

4. It must be verifiable and consistent. The requirements should be shown to be consistent and verifiable. The requirements are verified by system tester during system testing. No requirement should conflict with any other requirement.

5. It should be modifiable. The structure and style of the SRS are such that any necessary changes to the requirements can be made easily, completely and consistently.

6. It should be traceable to other requirements and related documents. The origin of each requirement must be clear. The SRS should facilitate the referencing of each requirement for future development or enhancement of documentation. Each requirement must refer to its source in previous documents.

7. SRS should not only addresses the explicit requirement but also implicit requirements that may come up during the maintenance phase of the software. It must be usable during operation and maintenance phase. The SRS must address the needs of the operation and maintenance phase, including the eventual replacement of the software.

3.4.2 Rules for Specifying Software Requirements

The following are the rules for specifying software requirements:

- Apply and use an industry standard to ensure that standard formats are used to describe the requirements. Completeness and consistency between various documents must be ensured.
- Use standard models to specify functional relationships, data flow between the systems and sub-systems and data structure to express complete requirements.
- Limit the structure of paragraphs to a list of individual sentences to increase the tractability and modifiability of each requirement and to increase the ability to check for completeness. It helps in modifying the document when required.

- Phrase each sentence to a simple sentence. This is to increase the verifiability of each requirement stated in the document.

3.4.3 Structure of a Typical SRS Document:

1. Introduction

- System reference and business objectives of the document.
- Goals and objectives of the software, describing it in the context of the computer-based system.
- The scope of the document.

2. Informative description about the system

- Information flow representation.
- Information content and structure representation.
- Description of sub-systems and System interface.
- A detailed description of the problems that the software must solve.
- Details of Information flow, content, and structure are documented.
- Hardware, software, and user interfaces are described for external system.

3. Functional Description of the system

- Functional description.
- Restrictions/limitations.
- Performance requirements.
- Design constraints.
- Diagrams to represent the overall structure of the software graphically.

4. Test and validation criteria

- Performance limitation, if any.
- Expected software response.
- It is essential that time and attention be given to this section.

5. Glossary

- Definitions of all technical or software-specific terms used in the document.

6. Bibliography

- List and reference of all documents that relate to the software.

7. Appendix

- Supplementary information to the specification.

3.5 DIFFERENT STANDARDS FOR DOCUMENTATION

This software documentation standard is used in the organization for uniform practices for documentation preparation, interpretation, change, and revision, to ensure the inclusion of essential requirements of different standards. Sometimes, documentation as per various standards is stated in the contractual agreement between the software vendor and the customer.

This standard will also aid in the use and analysis of the system/sub-system and its software documentation during the system/software life cycle of a software project.

Documentation comes in many forms, e.g., specifications, reports, files, descriptions, plans, source code listings, change requests, etc. and can be in electronic or paper form.

The Documentation Standard defines various aspects of documentation such as style, format, and the document revision/change process of these documents.

The International Standards, ISO/IEC 12207 – Software life cycle process, describes documentation as one of the supporting parallel process of software development process. It may be noted that this standard is not documentation standard but describes the process of documentation during the software development process. The following are other documentation standards:

1. **ISO/IEC 18019:** Guidelines for the design and preparation of user documentation for application software.

This standard describes how to establish what information users need, how to determine the way in which that information should be presented to the users, and then how to prepare the information and make it available. It covers both on-line and printed documentation. It describes standard format and style to be adopted for

documentation. It gives principles and recommended practices for documentation.

2. ISO/IEC 15910: Software user documentation

It process this standard specifies the minimum process for creating user documentation for software that has a user interface, including printed documentation (e.g., user manuals), on-line documentation, help text and on-line documentation systems.

3. IEEE 1063: Software user Documentation

It provides minimum requirement for structure, information content and format for user documentation. It does not describe the process to be adopted for documentation.

It is applicable for both printed and on-line documentation.

Components of software user documentation as described in IEEE 1063:

Software user Documentation:

Components of software user documentation:

1. Identification data (e.g., Title Page)
2. Table of contents
3. List of illustrations
4. Introduction
5. Information for use of the documentation such as description of software etc.
6. Concept of operations
7. Procedures
8. Information on software commands
9. Error messages and problem resolution
10. Glossary (to make the reader acquainted with unfamiliar terms)
11. Related information sources
12. Navigational features
13. Index
14. Search capability (for electronic document).

Documentation involves recording of information generated during the process of

software development life cycle. Documentation process involves planning, designing, developing, distributing and maintaining documents.

During planning phase, documents to be produced during the process of software development are identified. For each document, following items are addressed:

- Name of the document
- Purpose
- Target audience
- Process to develop, review, produce, design and maintain.

The following form part of activities related to documentation of development phase:

- All documents to be designed in accordance with applicable documentation standards for proper formats, content description, page number, figure/table.
- Source and accuracy of input data for document should be confirmed.
- Use of tools for automated document generation.
- The document prepared should be of proper format. Technical content and style should be in accordance to documentation standards.

Production of the document should be carried out as per the drawn plan. Production may be in either printed form or electronic form. Master copy of the document is to be retained for future reference.

Maintenance

As the software changes, the relevant documents are required to be modified.

Documents must reflect all such changes accordingly.

3.6 DOCUMENTATION AND QUALITY OF SOFTWARE

Inaccurate, incomplete, out of date, or missing documentation is a major contributor to poor software quality. That is why documentation and document control has been given due importance in ISO 9000 standards, SEI CMM software Maturity model. In SEI CMM Process Model and assessment procedure, the goal is to improve the documentation process that has been designed. A maturity level and documentation

process profile is generated from the responses to an assessment instrument.

One basic goal of software engineering is to produce the best possible working software along with the best possible supporting documentation. Empirical data show that software documentation products and processes are key components of software quality. Studies show that poor quality, out of date, or missing documentation are a major cause of errors in software development and maintenance. Although everyone agrees that documentation is important, not everyone fully realizes that documentation is a critical contributor to software quality. Documentation developed during higher maturity levels produces higher quality software.

3.7 GOOD PRACTICES FOR DOCUMENTATION

1. Documentation is the design document. The time to document is before actually implementing any design. A lot of effort can be saved in such cases.
2. Good documentation projects the quality of software. Many people take poor, scanty, or illiterate documentation for a program as a sign that the programmer is sloppy or careless of potential users' needs. Good documentation, on the other hand, conveys a message of intelligence and professionalism. If your program has to compete with other programs, better make sure that your documentation is at least as good as your competitors.

3.8 SUMMARY

The documentation process itself requires proper planning like the software development process. Documentation is an integral part of software development process and should not be taken lightly. Incomplete and inaccurate documentation may pose serious hurdle to the success of a software project during development and implementation.

Various documents like System requirement specification (SRS), system design specification (SDS), test design document and user manuals are produced during the life cycle of a software development process. SRS documents the high level requirements of the system without going into details of implementation issues,

whereas system design document describes how the requirements are finally to be implemented. It also describes the implementation issues with help of various system design tools. Test design document documents the requirement to be tested and procedure to be followed for testing. We have also discussed various ISO and IEEE standards on user documentation for uniform practices on documentation style and process. The relation of documentation with software quality has also been discussed.

3.9 QUESTIONS FOR EXERCISE

Q1. Define Documentation and discuss its importance

Q2. Write the process of documentation

Q3. Define System Requirement Specification and discuss its characteristics

Q4. Discuss some of the Standards for the documentation.

Q5. What are the good practices for documentation.

3.10 FURTHER READINGS

- Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich; Modern Systems Analysis and Design; Pearson Education; Third Edition; 2002.
- ISO/IEC 12207: Software life cycle process
- IEEE 1063: Software user Documentation
- ISO/IEC: 18019: Guide lines for the design and preparation of user documentation for application software

Reference Websites

- <http://www.systemanalysisanddesigns.com/documentation/>
- <http://www.sce.carleton.ca/squall>
- <http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/documentation.html>
- <http://www.sei.cmu.edu/cmm/>