UNIT 4 INTRODUCTION TO DOCUMENTATION OF SYSTEMS

Structure **Page No.**

4.0 INTRODUCTION

Documentation is a process to help users of software and other people to use and interact with system. Effective and accurate documentation is very necessary for the success of any system. Documentation becomes part of each step of system development through out the process of system development even before the documentation starts officially. During the process of system development, study reports and other necessary information are documented to help people involved in system development to understand the process. There are many kinds of documentation namely analysis documentation, design documentation, interface documentation, internal program documentation and user-oriented documentation. Documentation should not be seen as a overhead for system development. Rather, documentation has to be seen as an investment for the development of high quality software.

4.1 OBJECTIVES

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

- understand 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;
- know relation between the documentation and quality of software; and
- understand the good practices for documentation process.

4.2 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."

Why documentation?

Documentation is needed because it is

- 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. The figure below shows, how at the development process, documentation is done alongside each step. 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. It also defines and the target audience.

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.

Figure 4.1 depicts various stages of the process of documentation.

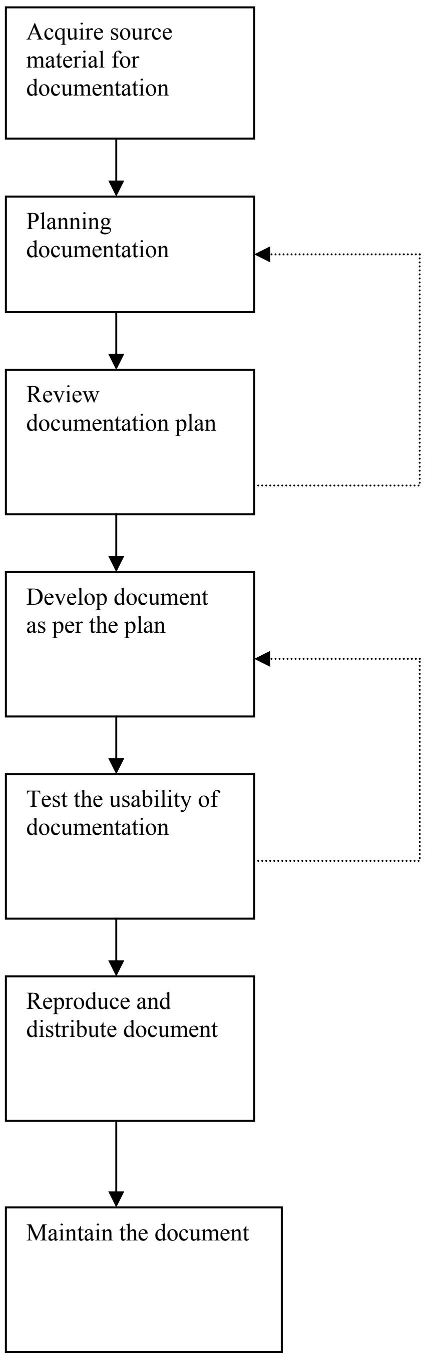
Acquire source material for documentation
Planning documentation
Review documentation plan
Develop document as per the plan
Test the usability of documentation
Reproduce and distribute document
Maintain the document

Figure 4.1: Stages of process of documentation

Check Your Progress 1

1. ISO standard ISO/IEC 12207:1995 describes documentation as a ____________ activity.

2. The documenter is responsible for preparation of a documentation plan (Yes/No).

4.3 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.

4.3.1 System Requirements Specification

System requirement 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.

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. So, all the requirements stated must be verifiable to know conformity to the requirements. 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.

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.

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.

4.3.2 System Design Specification

The system design specification or software design specification as referred to has a primary audience, the system implementer or coder. It is also an important source of information for the system verification and testing. The system design specification gives a complete understanding of the details of each component of the system, and its associated algorithms, etc.

The system design specification documents all as to how the requirements of the system are to be implemented. It consists of the final steps of describing the system in detail before the coding starts.

The system design specification is developed in a two stage process: In the first step, design specification generally describes the overall architecture of the system at a higher level. The second step provides the technical details of low-level design, which will guide the implementer. It describes exactly what the software must perform to meet the requirements of the system.

Tools for describing design

Various tools are used to describe the higher level and lower level aspects of system design. The following are some of the tools that can be used in the System Design Specification to describe various aspects of the design.

- **Data dictionary**

Definition of all of the data and control elements used in the software product or sub system. Complete definition of each data item and its synonyms are included in the data dictionary. A data dictionary may consist of description of data elements and definitions of tables.

Description of data element:

- Name and aliases of data item (its formal name).
- Uses (which processes or modules use the data item; how and when the data item is used).
- Format (standard format for representing the data item).
- Additional information such as default values, initial value(s), limitations and constraints that are associated with the data elements.

Table definitions

- Table name and Aliases.
- Table owner or database name.
- Key order for all the tables, possible keys including primary key and foreign key.
- Information about indexes that exist on the table.

Database schema: Database schema is a graphical presentation of the whole database. Data retrieval is made possible by connecting various tables through keys. Schema can be viewed as a logical unit from programmer's point of view.

E-R model: Entity-relationship model is database analysis and design tool. It lists real-life application entities and defines the relationship between real life entities that are to be mapped to database. E-R model forms basis for database design.

Security model: The database security model associates users, groups of users or applications with database access rights.

Trade-off matrix

A matrix that is used to describe decision criteria and relative importance of each decision criterion. This allows comparison of each alternative in a quantifiable term.

Decision table

A decision table shows the way the system handles input conditions and subsequent actions on the event. A decision table is composed of rows and columns, separated into four separate quadrants.

Input Conditions	Condition Alternatives
Actions	Subsequent action Entries

Timing diagram

Describes the timing relationships among various functions and behaviours of each component. They are used to explore the behaviours of one or more objects throughout a given period of time. This diagram is specifically useful to describe logic circuits.

State machine diagram

State machine diagrams are good at exploring the detailed transitions between states as the result of events. A state machine diagram is shown in figure 4.2.

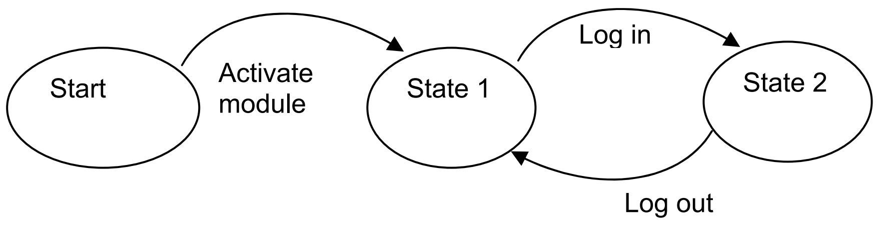

Figure 4.2 : A state machine diagram

Object Interaction Diagram

It illustrates the interaction between various objects of an object-oriented system. Please refer to figure 4.3. This diagram consists of directed lines between clients and servers. Each box contains the name of the object. This diagram also shows conditions for messages to be sent to other objects. The vertical distance is used to show the life of an object.

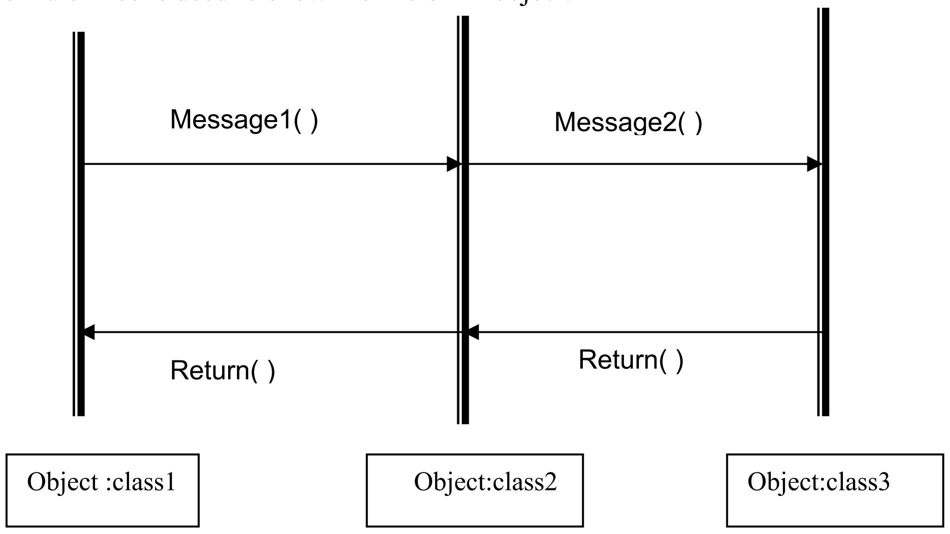

Figure 4.3: An object interaction diagram

A Flow chart shows the flow of processing control as the program executes. Please refer to figure 4.4.

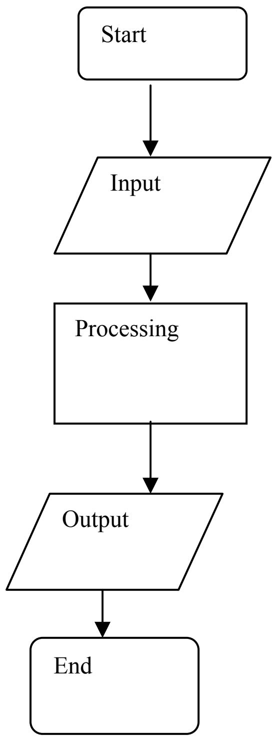

Figure 4.4 : Flow chart

Inheritance Diagram

It is a design diagram work product that primarily documents the inheritance relationships between classes and interfaces in object-oriented modeling. The standard notation consists of one box for each class. The boxes are arranged in a hierarchical tree according to their inheritance characteristics. Each class box includes the class name, its attributes, and its operations. Figure 4.5 shows a typical inheritance diagram.

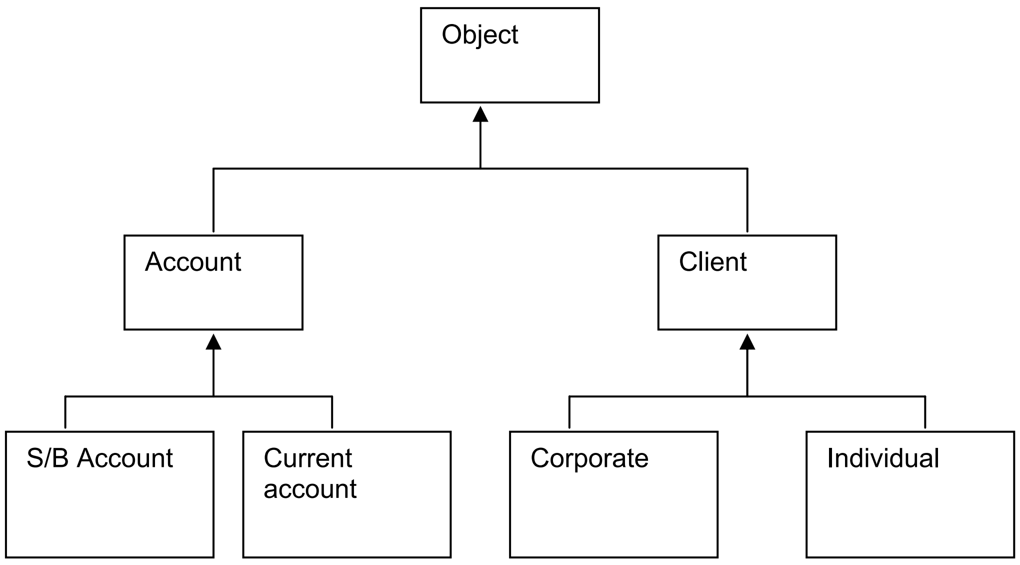

Figure 4.5: A typical inheritance diagram

Aggregation Diagram

The E-R model cannot express relationships among relationships. An aggregation diagram shows relationships among objects. When a class is formed as a collection of other classes, it is called an aggregation relationship between these classes. Each module will be represented by its name. The relationship will be indicated by a directed line from container to container. The directed line is labelled with the cardinality of the relationship. It describes "has a" relationship. Figure 4.6 shows an aggregation between two classes (circle *has a* shape).

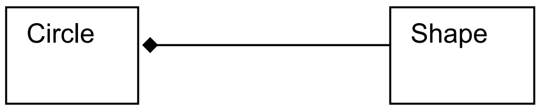

Figure 4.6 : Aggregation

Structure Chart

A structure chart is a tree of sub-routines in a program (Refer to figure 4.7). It indicates the interconnections among the sub-routines. The sub-routines should be labelled with the same name used in the pseudo code.

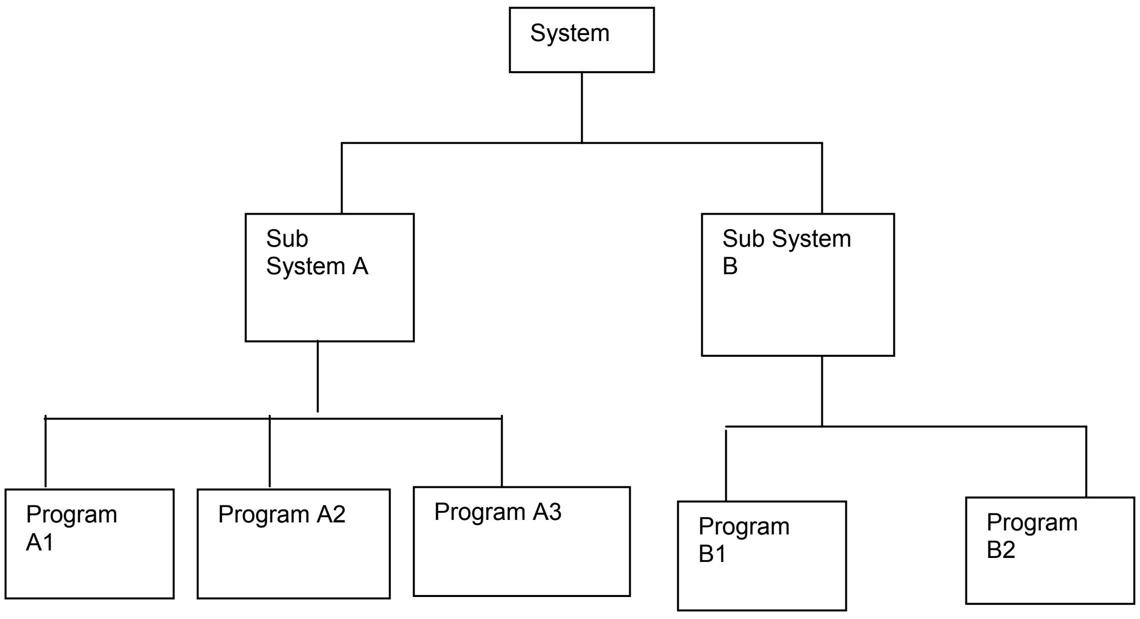

Figure 4.7 : A structures chart

Pseudocode

Pseudocode is a kind of structured English for describing algorithms in an easily readable and modular form. It allows the designer to focus on the logic of the algorithm without being distracted by details of language syntax in which the code is going to be written. The pseudocode needs to be complete. It describes the entire logic of the algorithm so that implementation becomes a routine mechanical task of translating line by line into source code of the target language. Thus, it must include flow of control.

This helps in describing how the system will solve the given problem. "Pseudocode" does not refer to a precise form of expression. It refers to the simple use of Standard English. Pseudocode must use a restricted subset of English in such a way that it resembles a good high level programming language. Figure 4.8 shows an example of pseudo code.

```
IF HoursWorked > MaxWorkHour THEN
    Display overtime message
ELSE
    Display regular time message
ENDIF
```

Figure 4.8 : Example of a Pseudocode

Contents of a typical System Design Specification document content

1. Introduction

   1.1 Purpose and scope of this document:
   Full description of the main objectives and scope of the SDS document is specified.

   1.2 Definitions, acronyms, abbreviations and references
   Definitions and abbreviations used are narrated in alphabetic order. This section will include technical books and documents related to design issues. It must refer to the SRS as one of the reference book.

2. System architecture description

   2.1 Overview of modules, components of the system and sub-systems

   2.2 Structure and relationships

   - Interrelationships and dependencies among various components are described.
   - Here, the use of structure charts can be useful.

3. Detailed description of components

   - Name of the component
   - Purpose and function
   - Sub-routine and constituents of the component
   - Dependencies, processing logic including pseudocode
   - Data elements used in the component.

4. Appendices.

4.3.3 Test Design Document

During system development, this document provides the information needed for adequate testing. It also lists approaches, procedures and standards to ensure that a quality product that meets the requirement of the user is produced. This document is generally supplemented by documents like schedules, assignments and results. A record of the final result of the testing should be kept externally.

This document provides valuable input for the maintenance phase.

The following IEEE standards describe the standard practices on software test and documentation:

1. 829-1998 IEEE Standard for Software Test Documentation
2. 1008-1987 (R1993) IEEE Standard for Software Unit Testing
3. 1012-1998 IEEE Standard for Software Verification and Validation

The following is the typical content of Test Design Document:

1. **Introduction**

Purpose

The purpose of this *document* and its intended audience are clearly stated.

Scope

Give an overview of testing process and major phases of the testing process. Specify what is not covered in the scope of the testing such as, supporting or not third party software.

Glossary

It gives definition of the technical terms used in this document.
References

Any references to other external documents stated in this document including references to related project documents. They usually refer the System Requirement Specification and the System Design Specification documents.

Overview of Document

Describe the contents and organization of the document.

2. **Test Plan**

A test plan is a document that describes the scope, approach, resources and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, and the person who will do each task, and any risks that require contingency planning.

2.1. Schedules and Resources

An overview of the testing schedule in phases along with resources required for testing is specified.

2.2. Recording of Tests

Specify the format to be used to record test results. It should very specifically name the item to be tested, the person who did the testing, reference of the test process/data and the results expected by the test, the date tested. If a test fails, the person with the responsibility to correct and retest is also documented. The filled out format would be kept with the specific testing schedule. A database could be used to keep track of testing.

2.3. Reporting test results

The summary of what has been tested successfully and the errors that still exist which are to be rectified is specified.

3. **Verification Testing**

3.1. Unit Testing

For each unit/component, there must be a test which will enable tester to know about the accurate functioning of that unit.

3.2. Integration testing

Integration test is done on modules or sub-systems.

4. **Validation Testing**

4.1. System Testing

This is the top level of integration testing. At this level, requirements are validated as described in the SRS.

4.2. Acceptance and Beta Testing

List test plans for acceptance testing or beta testing. During this test, real data is used for testing by the development team (acceptance testing/alpha testing) or the customer (beta testing). It describes how the results of such testing will be reported back and handled by the developers.

4.3.4 User Manual

This document is complete at the end of the software development process.

Different Types of User Documentation

Users of the system are not of the same category and their requirements vary widely. In order to cater to the need of different class of user, different types of user documentation are required. The following are various categories of manuals:

- Introductory manual: How to get started with the system?
- Functional description: Describes functionality of the system.
- Reference manual: Details about the system facility.
- System administrator guide: How to operate and maintain the system?
- Installation document: How to install the system?

The following is the typical content of User Manual

1. **Introduction**

1.1 Purpose

The purpose of this *document* and its intended audience is stated. If there is more than one intended audience, provide information in this section and direct the reader to the correct section(s) for his/her interest.

1.2 Scope of Project

Overview the product . Explain who could use the product. Overview the services that it provides. Describe limitation of the system. Describe any restrictions on using or copying the software and any warranties or contractual obligations or disclaimers.

1.3 Glossary

Define the technical terms used in this document. Do not assume that the reader is expert.

1.4 References

References to other documents cited anywhere in this document including references to related project documents. This is usually the only bibiliography in the document.

1.5 Overview of Document

The contents and organization of the rest of this document are described.

2. **Instructional Manual**

This section should be divided in the manner that will make it user friendly.

2.1 System Usage

Provide examples of normal usage. Images of the screendumps are very useful to provide a look and feel of the product. Provide any necessary background information. On-line help system is very common for systems today. Information on how to use the on-line help system, how to access it may be provided here.

3. **User Reference Manual**

3.1 List of Services

Provides an *alphabetical* listing of services provided by the system with references to page numbers in this document where the concerned service is described.

3.2 Error Messages and Recovery

Provides an *alphabetical* list of all error messages generated by the system and how the user can recover from each of these errors.

4. **Installation Information**

Installation information is provided including the operating environment.

Maintenance Manual

The supplier/developer of the software is sometimes different from the software maintenance agency. In such cases, the criticality of maintenance manual assumes a bigger role. Maintenance manuals provide precise information to keep your product operating at peak performance. Similar to installation manuals, these documents may range from a single sheet to several hundred of pages.

Check Your Progress 2

1. The system design specification is developed in ____________ process.
2. Pseudo code is used to describe _______________.
3. _________________ provides information as to how to operate and maintain the system.

4.4 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 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.

Check Your Progress 3

1. ____________________ is used in the organization for uniform practices for documentation.
2. International Standard ISO/IEC 12207 is documentation standard (Yes/No).

4.5 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.

4.6 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.

Check Your Progress 4

1. SEI CMM is a ________ model.
2. One of the basic goals of software engineering is _________.

4.7 SUMMARY

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. The documentation process itself requires proper planning like the software development process. 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.

4.8 SOLUTIONS/ ANSWERS

Check Your Progress 1

1. Support
2. Yes

Check Your Progress 2

1. Two stage.
2. Algorithms.
3. System administration guide.

Check Your Progress 3

1. Software documentation standard.
2. No.

Check Your Progress 4

1. Capability Maturity.
2. To produce accurately working software along with the best possible supporting documentation.

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