

EXERCISE

1. Explain the conversion process.
2. How many conversion methods are there?
3. Can you spot the major difference in each of the conversion methods?



CHECK YOUR PROGRESS

Q1: What is the significance of conversion?

Q2: What are the conversion plans?

Q3: Explain the difference between Parallel conversion and Direct cutover conversion.

14.4 ACTIVITY NETWORK FOR CONVERSION

The activity network for conversion consists of the following sections: Training, File Conversion, Forms and Displays conversion, Conversion of physical facilities and Conversion of Administrative Procedures. Each of theseis discussed below:

14.4.1 Training

Even if a software product is well designed and technically superior, but if the people who are supposed to use it do not know how to use it, then the software may turn out to be a failure. So, the intended people who will actually use the software should know how to use it, what their specific roles will be, and what the system will do or will not do under different conditions.

Training Systems Operators:

As software can have different constraints and limitations, so the systems operators must be trained properly so that they can graciously handle all kinds of outcomes and issues. The operators should be acquainted with all the common types of errors or anomalies that might occur, they should know how to recognize

Fitness for use means that not only the essential requirements but the requirements of the customers are also taken into consideration. For example, customers may desire a certain product with some additional requirements, like a hatchback car with added space, comfort, performance, etc. As different customer may have different requirements, so there may be different models of the same products.

Ans. to QNo2:Functional requirements: These are the fundamental criteria that the software should essentially satisfy and are usually stated by the users / customers. Some of the examples of functional requirements are that the system should be able to accept and process valid input, produce the anticipated output, should be free of errors and bugs, etc. Here the user is mostly able to visualize them in the software.

Non-functional requirements: These are the requirements that are mostly specified by the software developer or project leader based on the requirements and restrictions of the system as a whole. Although it consists of many attributes like Complexity, Efficiency, Interoperability, Learnability, Modularity, Portability, Reliability, Resilience, Reusability, Robustness, Scalability, Security, Supportability, Testability and Usability, but not all of these attributes are obligatory to be fulfilled.

Ans. to QNo3: The purpose of Software Quality Assurance (SQA) is to ensure that the software meets the required specifications and standards. It provides a clear picture about the performance of the software, information of resource allocation, existence of risk, deficiencies, etc. To assure software quality, many reviews and audits are performed throughout the life cycle of system development process.

Ans. to Q No4: Software may successfully fulfill the basic requirement of essentially having no inherent bugs or errors, but it may not stand strong on other non-measurable attributes like Reusability, Robustness, etc. As a result, the overall quality of the software may be hampered. Thus software quality is assessed not only by

measureable attributes like bugs, logical errors, etc., but also depends on other non-functional attributes like Interoperability, Learnability, Maintainability, etc. McCall's Factor Model gives us an all-inclusive method which will cover all the attributes and aspects of software quality.

Ans. to Q No5: With the software requirements specification (SRS), the developers, the product owners, and various other stakeholders will have to meet, discuss and formally write down about the system's detailed description and purpose, its functionalities, its interfaces, performance criteria, functional and non-functional requirements, quality standards, security issues, financial as well as time constraints, resource allocation, risk involved, deficiencies, etc., all has to be noted down. Here, the developers will get to know what goals and vision the client exactly has in his or her mind, whereas the client will get to know about the technology aspects. Once the SRS has been written, developers can refer it at any point of time to clear any ambiguity, managers can use it to plan the system development and maintenance process, users / clients can use it to understand the software, test engineers can use it to create test cases, develop validations sets, etc. Thus the SRS will provide the system with a concrete and well-defined structure to follow.

Ans. to Q No6: While writing the Software Requirement Specification, the functional requirements of the software should be clearly defined, that is, here the following question should be asked: 'What has to be developed'? or 'What are supposed to be the essential functions of the system'? The internal logic like 'how to develop the functionalities' or 'when to develop them' are not dealt right now. Here primarily, the views of the users / customers / product owners are taken into consideration. For non-functional requirements we have handle the questions like: 'How each feature / function of the software should work? Given certain conditions and limitations, how it should behave? How to handle accessibility and security issues?' etc. Here we have to describe how the software has to work in terms of Efficiency,

- Verifying conversion schedules.

14.3.2 Conversion Methods

There are four methods of conversion which are: Parallel Conversion, Direct Cutover Conversion, Pilot Approach and Phase-In Method.

Parallel Conversion: In Parallel conversion, both old and new systems are used simultaneously. If in case the new system fails then the old system will provide a fallback, thus providing a greater sense of security.Here the cost may increase as we have to keep on running both the systems.

Direct Cutover Conversion: In Direct cutover conversion, the old system is completely replaced by the new system. Since there is no fallback, it requires meticulous planning as we have to ensure that the new system works at all cost. The advantage is that the benefit and utilities of the new system can be realized immediately.

Pilot Approach: In Pilot approach, phased approach is used that gradually implements the system across all users. Since it is implemented in a phased manner, large risk that comes from implementing the whole system at once can be avoided. Also,installation and training of personals can also be easily implemented. But phase-wise implementation has a disadvantage that it might take very long time to completely install and implement the whole system.

Phase-In Method: This method is similar to Pilot approach in that the system is implemented in one part of the organization and feedback is collected from the users. Based on the feedback, the system can further be improved to give a better user experience and then the implementation moves on to the next part of the organization. The feedback given by the user is used to show that the old system had problems and hence had to be replaced. This will help in smooth transition of the new system to the rest of the organization.

Software Requirement Specification and Software Design Specification; and also various methods to formulate good requirements and design specifications. In the unit, we will discuss about how to implement thesoftware that has been developed in the actual environment, what are the processes involved, how to train the users so that they can efficiently handle the system and how to plan for a smooth conversion. In the next unit, we will discuss about project management.

14.3 CONVERSION

Conversion is the part of the implementation plan where the process of transferring the softwarefrom the old system to the new one occurs. If we have a proper conversion plan then it will provide understandable and structured approach to improve the communication between management and project team.The users, project managers, software developers and other engineers meet and review the project plan, the system test documentation and the implementation plan. Once the conversion plan is approved, files are converted, parallel processing between the existing and the new systems is initialized and results of computer runs from the new system are saved. After conversion is completed, parallel processing is discontinued, implementation results are documented and the new system becomes operational.

14.3.1 Conversion Plans

Conversion plan contains description of all the activities that must occur during implementation of the new system. It anticipates possible problems and solutions to deal with them. It includes the following activities:

- Naming all files for conversions.
- Identifying the data requirements to develop new files during conversion.
- Listing all the new documents and procedures that are required.
- Identifying the controls to be used in each activity.
- Identifying the responsibility of person for each activity.

Interoperability, Modularity, Scalability, etc. As the non-functional attributes are very difficult to describe because different person will have a different point of view, so we can assign score or point to each attribute.

Ans. to Q No7: The characteristics of a good SRS are (i) Correctness, (ii) Completeness, (iii) Consistency, (iv) Unambiguousness, (v) Ranking for importance and stability, (vi) Modifiability, (vii) Verifiability, (viii) Traceable, (ix) Design Independence, (x) Testability, (xi) Understandable and (xii) Right level of abstraction.

Ans. to Q No8: Software Requirement Specification (SRS) describes what the various stakeholders (like customers, users, project managers, software developers, engineers, invertors, test engineers, etc.) want in terms of features, functions, performance, constraints, quality, security, etc. Software Design Specification (SDS) describes the technical details of how the features and functions have to be implemented. It will contain information about the software architecture and describe how each component will contribute to meeting the requirements. It will help the programmer during coding and implementation.

Ans. to Q No9: A good design specification should have the following characteristics:

- Correctness: Software design should be correct as per requirement.
- Completeness: The design should have all components like data structures, modules, and external interfaces, etc.
- Efficiency: It should allocate resources very efficiently.
- Flexibility: In case of occurrence of changes, it should be modifiable.
- Consistency: There should not be any inconsistency in the design.
- Maintainability: The design should be as simple as possible so that it can be easily maintained by other designers.



13.11 MODEL QUESTIONS

Q1. "When we say quality of a product, different people may visualize it differently." - Elaborate. In software industry, describe the issues related to software quality and software quality assurance.

Q2. Explain in detail McCall's Factor Model.

Q3. Explain in detail the structure of a Software Requirement Specification (SRS).

Q4. Describe the characteristics of a good Software Requirement Specification?

Q5. Describe the Software design principles.

*** ***** ***

UNIT 14: IMPLEMENTATION

UNIT STRUCTURE

14.1 Learning Objectives
14.2 Introduction
14.3 Conversion
14.3.1 Conversion Plans
14.3.2 Conversion Methods
14.4 Activity Network for Conversion
14.4.1 Training
14.4.2 File Conversion
14.4.3 Forms and Displays Conversion
14.4.4 Conversion of Physical Facilities
14.4.5 Conversion of Administrative Procedures
14.5 Let Us Sum Up
14.6 Further Readings
14.7 Answers To Check Your Progress
14.8 ModelQuestions

14.1 LEARNING OBJECTIVES

After going through this unit, you will be able to –

- learn about the conversion process of migrating from old to new system
- understand different conversion plans and methods
- learn about the different activities of the conversion process
- visualize that success of software depends on proper training of all the people who will be using the system

14.2 INTRODUCTION

In the previous unit, we learned about the various aspects, issues and factors of software quality. We also learned about the significance of