UNIT 3: ROLE OF SYSTEM ANALYST

UNIT STRUCTURE

3.1 Learning Objectives
3.2 Introduction
3.3 Role of System Analyst
3.4 Interpersonal Skill
3.5 Academic and Professional Qualification
3.6 Multifaceted Role of System Analyst
3.7 User Interface Analyst
3.8 Conflict Resolution
3.9 Place of Analyst in The MIS Organization
3.10 Let Us Sum Up
3.11 Further Reading
3.12 Answers to Check Your Progress
3.13 Model Questions

3.1 LEARNING OBJECTIVES

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

- define the role and responsibility of a system analyst
- understand interpersonal skill and able to apply it
- learn about professional and academic qualification needed for system analyst
- define conflict and apply conflict resolution strategy
- compare and contrast system analyst and MIS analyst

3.2 INTRODUCTION

In the previous unit we learned about the system development life cycle. In this unit, we will learn about the role of system analyst. The unit covers topics like interpersonal skill, academic and professional qualification, user interface analyst and conflict resolution in detail. The next unit explores the topics related to system planning and initial investigation.

2.4.5 Development

Once the system design is completed, the system is actually programmed. The goal of this phase is to translate the design of the system into code in a selected programming language. A well written code reduces the testing and maintenance effort. Since the testing and maintenance cost of software are much higher than the coding cost, the goal of coding should be to reduce the testing and maintenance effort. Hence, during coding the focus should be on developing programs that are easy to write. The analyst should decide whether to buy a commercial software or to develop new customized programs with the help of programmers. This decision depends on the cost of the software and the cost of programming such software. Documentation of the programs while programming the software is also necessary to provide explaination of the procedure so that other programmers can work on the programs if required .

In the system development life cycle, after the completion of the development phase the product becomes ready to be delivered to the client.

2.4.6 System Testing

With the help of system testing, quality of the product is tested by applying various quality control measure after software development. It is the process to ensure that the developed system perform the intended tasks which are already mentioned in the SRS document. Its basic function is to detect errors in the software if any.

System testing is usually carried out in a planned manner. During system testing, the system is used as an experiment to ensure that the software does not fail, i.e., it will run according to its specification. The system is tested with some test data and the results are examined for their validity. For testing to be successful, proper selection of test cases is essential. Some of the users may

be permitted to operate on the system so that the analyst can ascertain that the system can work in the specified environment.

There are three different categories in testing activities, which are ***α - testing, β-testing*** and *acceptance testing*.

Generally, the ***α-testing*** (alpha-testing) is performed by the development team or testers who are usually internal employees of the organization. It is performed before releasing the product to the real user of the software.

β-testing (beta-testing) is performed to access the product by some clients in their environment who are not part of the organization. We can say that beta testing is performed at end-user level. The main focus of beta testing is to access the quality of the product. This test helps in collecting feedback from the users on the product. It also ensures that the product is ready for real time users after fixing the defects if any. This also helps in enhancing the product.

After the product delivery, ***acceptane testing*** is performed to determine whether to accept or reject the delivered product. It is a method of software testing where a system is tested for acceptability.

Testing phase continues until the software becomes stable, error-free and working according to the requirements.

2.4.7 Implementation

In the implementation phase, the organization configures and enables the system security features, tests the functionality of these features, installs or implements the system, and obtains a formal authorization to operate the system. This phase consists of some activities like, installation of required hardware, programs, collection of data, organization of users to interact with and run the system etc.

Necessary training for the users to use the system is also provided in implementation phase. When the new system is linked

2.8 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1:

(i) (b) Preliminary investigation (ii) (c) System analysis

(iii) (d) analyst (iv) (b) Analysis phase

Ans to Q No 2: (i) independent (ii) high cohesion, low coupling

(iii) detailed design

Ans to Q No 3: (a) adaptation (b) evaluation (c) documentation

Ans to Q No 4: (a) True (b)True (c) False (d) False

2.9 MODEL QUESTIONS

Q 1: What is SDLC? What are phases associated with SDLC?

Q 2: Define the terms cohesion and coupling in the context of software design.

Q 3: When is a system design said to be good design?

Q 4: What do you mean by feasibility study? In preliminary investigation three types of feasibilities are usually studied. Name them.

Q 5: Why is maintenance of a system necessary?

Q 6: Discuss briefly the various phases of SDLC.

Q 7: What is documentation? Explain the uses of documentation.

Q 8: Write short notes on any one of the following:

(a) System Design

(b) Recognition of need

(c) Analysis

(d) System Testing

*** ***** ***

- During design phase, different modules, interactions among them, algorithms, data structures are identified.
- Based on the design document, programmer build the software product in the development phase. The goal of this phase is to translate the design of the system into code in some programming language.
- Once the developers build the software product, then the testing team tests the functionality of the entire newly developed system. It ensures that the newly developed product works according to the customer requirements.
- After testing process is over the product is released to the customer for implementation. Users are provided training so that they can operate and handle the software product in the implementation phase. Documentation part is also provided for better understanding of the product.
- In the maintenance phase, the developed product is taken care of. As per change of environment and technology, the software product is updated and maintained time to time.

At the end of this system documentation is described. With the help of documentation, maintenance of software becomes easier. All the aspects of software can be understood if proper software documentation is done.



2.7 FURTHER READING

1. Awad Elias M. (1996). System Analysis and Design.
2. Mall, R. (2018). Fundamentals of software engineering. PHI Learning Pvt. Ltd..
3. Kendall, K. E., & Kendall, J. E. (2011). Systems analysis and design (Vol. 2013). Upper Saddle River, NJ: Pearson Prentice Hall.

to terminals or remote machines/sites, then the test of the network along with the system is also included under implementation.

2.4.8 Maintenance

Once the system is implemented, it should be evaluated. In this phase, systems and products are in place and operating. The organization should continuously monitor performance of the system to ensure that it is consistent with pre-established user and security requirements and that the needed system modifications are incorporated.Thus, *evaluation* is the process of veryfying the capability of a system, after it is put in operation, to see whether it is meeting the objectives or not.

Maintenance includes all the activities after the installation of the software that is performed to keep the system operational. As we have mentioned earlier, software often has design faults. For large systems, removing all the faults before delivery of the software is extremely difficult and faults will be discovered long after the system is installed. As these faults are detected, they have to be removed. Maintenance includes fixing of errors as well as removing of errors.

Maintenance also needed due to a change in the environment or the requirements of the system. Activities like enhancement, adaptation are also included in maintainance phase. Adding new functions or additional capabilities come under *enhancement* and customizing the software to run in the new environment is termed as *adaptation*.

2.5 SYSTEM DOCUMENTATION

System documentation is the collection of documents that describes the requirements, capabilities, limitations, design, operation, and maintenance of a system. It is basically a written document that is often accompanied with a software product. Documentation are very important

because they represent the formal information flow of the present system. It also provided the performance criteria of a system and an explaination of how the system can be used. The documentation needs to be kept up-to-date throughout the project. For instance, a user manual can be a good example of documentation. Usually, a user manual is used to refer for installing, using, and providing maintenance to the software product or application.

The objective of system documentation is to keep the system running in optimum conditions. Some other uses of documentations are:

- To ensure that the system can be maintained after completion even though programmers involved in the original design or programming of the system may not be available.
- To help new programmer for making necessary corrections, adaptation of new features etc.
- To ensure what data should be inputed and how and what information will be available from the system.

There are different types of documentation. Some of them with their usefulness are mentioned here:

- ***Design documentation*** describes the overall system design and includes system flowcharts, all input/output formats, file description etc.
- ***Program documentation*** consists of programming specification like programming logic, control flow, graphics used etc.
- ***Training documentation*** includes manual for user training.
- ***Operation documentation*** describes how to handle the software.
- ***User reference documentation*** provide quick and clear answer just like dictionary.



CHECK YOUR PROGRESS

Q 3: Fill in the blanks:

(a) Customizing the software to run in the new environment is termed as .....

(b) ..... is the process of verifying the capability of a system after it is put in operation.

(c) ..... needs to be kept up-to-date throughout the project

Q 4: State True or False:

(a) Maintenance is the process of incorporating changes in the existing system.

(b) Selection of hardware is an implementation activity.

(c) User interaction is not necessary in analysis phase of SDLC.

(d) Design documentation includes programming secifications.



2.6 LET US SUM UP

The unit discusses the complete development process of a software product with the various stages of System Development Life Cycle(SDLC). The various phases are: *Recognition of need, Feasibility Study, Requirement Analysis, System Design, Development, Testing, Implementation, and Maintenance.* The activities discussed in different phases of SDLC are summarized in the following points:

- Recognition of need is basically a preliminary assessment of the existing system to determine whether an alternative system can solve the problem.
- A feasibility study, as the name implies is a study to determine the necessity for developing a new system or product. It is a process of ensuring a product to be developed is technically. economically as well as legally justifiable.
- In requirement analysis phase, users expectation from the new software product being built are identified. The main purpose of requirement analysis phase is to formulate the system's reruirements in a document format.
- Design phase involves the process of tranforming user requirements into some suitable form which is useful for the programmer or developer to develop the new system.