UNIT 3 PROCESS OF SYSTEM DEVELOPMENT

Structure **Page No.**

3.0 INTRODUCTION

Information Systems Analysis and Design is a complex and stimulating process that is used to develop and maintain computer based information systems. The analysis and design of information systems are driven from an organizational point of view. An organization might consist of whole enterprise, specific departments or individual work groups. Information Systems Analysis and Design is, therefore, an organizational improvement process. Systems are built and rebuilt (enhanced) for organizational benefits. Benefits result by adding value during the process of creating, producing and supporting the organization's services and products. Thus, Information Systems Analysis and Design is based on the understanding of objectives, structure and processes of organization and the knowledge about the application of Information Technology for this purpose. Most organizations find it beneficial to use standard sets of steps, called a systems development methodology, to develop and support their information systems (IS). Like many processes, the development of Information Systems often follows a life cycle called Systems Development Life Cycle. For example, a product follows a life cycle when it is created, tested and introduced in the market. Its sale increases and goes to peak point and after that it declines and a new product or next version of the existing product is introduced in the market to replace it. SDLC is a common methodology for systems development in many organizations, consisting of various phases that mark the progress of system analysis and design effort.

3.1 OBJECTIVES

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

- define Information Systems Analysis and Design;
- describe the information systems development life cycle; and

- list alternatives to SDLC and compare the advantages and disadvantages of SDLC to its alternatives.

3.2 SYSTEMS DEVELOPMENT LIFE CYCLE

Most organizations find it beneficial to use a set of steps, called a systems development methodology, to develop and support their information system. Like many processes, the development of information system often follows a life cycle. The system development life cycle (SDLC) is a common methodology for system development in many organizations, featuring various phases that mark the progress of the system analysis and design effort.

Although any life cycle appears at first glance to be a sequentially ordered set of phases but actually it is not. The specific steps and their sequence are meant to be adapted as required for a project, consistent with management approach. For example, in any given SDLC phase, the project can return to an earlier phase, if necessary. If a commercial product does not perform well just after its introduction, it may be temporarily removed from the market and improved before being re-introduced. In the system development life cycle, it is also possible to complete some activities in one phase in parallel with some other activities of another phase. Sometimes, life cycle is iterative; that is, phases are repeated as required until a satisfactory and acceptable system is found. Such an iterative approach is special characteristic of rapid application development methods, such as prototyping. Some people consider life cycle to be spiral, in which we constantly cycle through the phases at different levels of detail. The life cycle can also be thought of a circular process in which the end of the useful life of one system leads to the beginning of another project that will develop a new version or replace an existing system altogether. However, the system development life cycle used in an organization is an orderly set of activities conducted and planned for each development project? The skills of a system analyst are required to be applied to the entire life cycle.

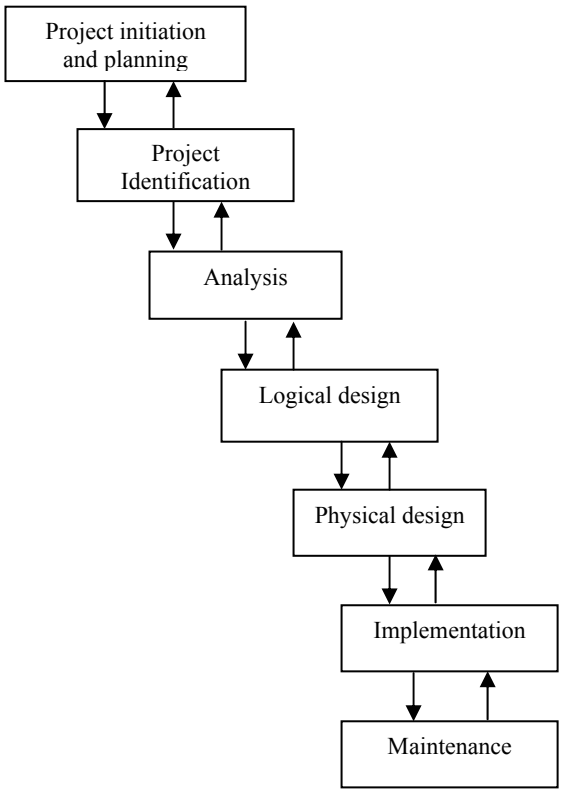

Figure: 3.1: Phases of System Development Life Cycle

Every custom software producer will have its own specific detailed life cycle or system development methodology. Even if a particular methodology does not look like cycle, you will discover that many of SDLC steps are performed, SDLC techniques and tools are used.

In order to make this unit generic, we follow a rather general life cycle model, as described in figure 3.1.

This model resembles a staircase with arrows connecting each step to the step before and to the step after it. This representation of the system development life cycle (SDLC) is sometimes referred to as the "waterfall model". We use this SDLC as one example of methodology but more as a way to arrange the steps of systems analysis and design. Each phase has specific outputs and deliverables that feed important information to other phases. At the end of each phase, system development project reaches a milestone and, as deliverables are produced, parties outside the project team often review them.

3.3 PHASES OF SDLC

SDLC consists of mainly seven steps. These are:

1. Project Identification and Selection
2. Project Initiation and Planning
3. Analysis
4. Logical Design
5. Physical Design
6. Implementation
7. Testing.

3.3.1 Project Identification and Selection

The first phase in the SDLC is called project identification and selection. In this phase, the user identifies the need for a new or improved system. In large organizations, this identification may be part of a systems planning process. Information requirements of the organization as a whole are examined, and projects to meet these requirements are proactively identified. The organization's information system requirements may result from requests to deal with problem in current system's procedures, from the desire to perform additional tasks, or from the realization that information technology could be used to capitalize on an existing opportunity. These needs can then be prioritised and translated into a plan for the Information System department including a schedule for developing new major systems. In smaller organizations, determination of which systems to develop may be affected by user request submitted as the need for new or enhanced systems arises as well as from a formal information planning process. In either case, during project identification and selection, an organization determines whether or not resources should be devoted to the development or enhancement of each information system under consideration. The outcome of the project identification and selection process is a determination of which systems development projects should be undertaken by the organization at least in terms of an initial study.

3.3.2 Project Initiation and Planning

The second phase is project initiation and planning. The problems that are identified should be investigated and a decision to implement the information system or not for the organization should be taken. A critical step at this point is determining the scope of the proposed system. The project leader and initial team of system analysts also produce a specific plan for the proposed project, which the team will follow using the

remaining SDLC steps. Now, this baseline project plan customizes the standardized SDLC and specifies the time and resources needed for its execution.

The formal definition of a project is based on the likelihood that the organization's information system department is able to develop a system that will solve the problem or use the opportunity and determine whether the costs of developing the system outweigh the benefits it could provide. The final presentation with the subsequent project phases is usually made by the project leader and other team members to someone in management or to a special management committee with the job of deciding which projects the organization will undertake.

3.3.3 Analysis

Analysis is the next phase. During this phase, the analysis has several sub-phases. The first is requirements determination. In this sub-phase, analysts work with users to determine the expectations of users from the proposed system. This sub-phase usually involves a careful study of current systems, manual or computerized that might be replaced or enhanced as part of this project. Next, the requirements are studied and structured in accordance with their inter-relationships and eliminate any redundancies. Third, alternative initial design is generated to match the requirements. Then, these alternatives are compared to determine which alternative best meets the requirement in terms of cost and labour to commit to development process.

In this phase, feasibility study of the proposed system is also performed. Various types of feasibilities are:

- Technical feasibility
- Economic feasibility
- Behavioural feasibility
- Operational feasibility
- Legal feasibility
- Time feasibility.

If the proposed system is not feasible to develop, it is rejected at this very step. The output of the analysis phase is a description of (but not are detailed design for) the alternative solution recommended by the analysis team. Once, the recommendation is accepted by those with funding authority, you can begin to make plans to acquire any hardware and system software necessary to build or operate the system proposed.

System Design: After analysis phase is complete, design of the system begins. The design consists of logical and physical design of the system. The fourth and fifth phases are devoted to design of the new and enhanced system. During design, you and the other analysts convert the description of the recommended alternative solution into logical and then physical system specifications. You must design all aspects of the system from input and output screens to reports, databases, and computer processes. Design occurs in two phases, viz., logical design and physical design.

3.3.4 Logical Design

Logical design is not tied to any specific hardware and systems software platform. Theoretically, the system could be implemented on any hardware and systems software. The idea is to make sure that the system functions as intended. Logical design concentrates on the business aspects of the system.

3.3.5 Physical Design

In physical design, the logical design is turned into physical or technical specifications. For example, you must convert diagrams that map the origin, flow, and

processing of data in a system into a structured systems design that can then be broken down into smaller and smaller units known as modules for conversion to instruction written in a programming language. You design various parts of the system to perform the physical operations necessary to facilitate data capture, processing, and information output. During the physical design, the analyst team decides the programming language in which the computer instructions will be written in, which database system and file structure will be used for the data, the platform that will be used and the network environment under which the system will be run. These decisions finalize the hardware and software plans initiated at the end of the analysis phase. Now, proceedings can be made with respect to acquisition of any new technology not already present in the organization. The final product of the design phase is the physical system specification in a form ready to be turned over to programmers and other system builders for construction. The physical system specifications are turned over to programmers as the first part of the implementation phase.

Check Your Progress 1

1. What is the difference between Project Identification and Project Initiation?
   ……
   ……
   ……
2. What is the difference between analysis and design?
   ……
   ……
   ……
3. Why do most of the projects die after feasibility phase?
   ……
   ……
   ……

3.3.6 Implementation

During implementation, you turn system specification into working system that is tested and put into use. Implementation includes **coding, testing and installation**. During coding, programmers write programs that make up the system. During testing, programmers and analysts tests the individual programs and the entire system in order to find and correct errors. During installation, the new system becomes a part of the daily activities of the organization. Application is installed or loaded, on existing or new hardware and users are introduced to new system and trained. The analysts begin planning for testing and installation as early as the project initiation and planning phase, since testing and installation require extensive analysis in order to develop the right approach.

Installation of the system can be done in the following three ways:

- **Direct conversion:** In this type of conversion, the software is directly installed at user's site.
- **Parallel conversion:** In this type of conversion, both the old and new systems are run in parallel for some time. After monitoring the new system for a reasonable period of time and if it is performing well, then, the new system is implemented replacing the old one.

- **Phased conversion:** In this type of conversion, the system is installed module by module.

Implementation activities also include initial user support such as the finalization of *documentation, training programs, and ongoing user assistance*. Note that documentation and training programs are finalized during implementation. Documentation is produced throughout the lifecycle. Implementation can continue for as long as the system exists since ongoing user support is also part of implementation. Despite the best efforts of analysts, managers, and programmers, however, installation is not always a simple process. Many well-designed systems can fail if implementation is not well managed. The management of implementation is usually done by the project team.

3.3.7 Maintenance

The final phase is maintenance. When a system is operating in an organization, users sometimes find problems with how it works and often think of better ways to perform its functions. Also, the organization's requirements with respect to the system change with time. During maintenance, programmers make the changes that users ask for and modify the system to reflect and support changing business conditions. These changes are necessary to keep the system running and useful. Maintenance is not separate phase but a repetition of the other lifecycle phases required to study and implement the needed changes. Thus, maintenance is an overlay to the life cycle rather than a separate phase. The amount of time and effort devoted to maintenance depends a great deal on the performance of the previous phase of life cycle. There comes a time, however, when an information system is no longer performing as desired, when maintenance cost becomes prohibitive, or when the organization's needs has changed substantially. Such problems are an indication that it is the time to begin designing the system's replacement, therefore, completing the loop and starting the life cycle over again. Often, the distinction between the major maintenance and new development is not clear, which is another reason why maintenance often resembles the lifecycle itself.

Maintenance is of three types:

Corrective maintenance: In this type, the errors that creep into the system are removed. Hence the name *corrective maintenance*.

Adaptive maintenance: It is done to adapt with the changing external factors. For example, if the government rules change regarding the Dearness Allowance from 52% to 58%, then the changes have to be made in the Information System to adapt with the changing scenario**.**

Perfective maintenance: This is done to satisfy the users' requirements to make the system more and more perfect.

The SDLC is a highly linked set of phases where output of one phase serves as input to the subsequent phase. Throughout the systems development life cycle, the systems development project needs to be carefully planned and managed. Therefore, the larger the project, the greater is the need for project management.

3.4 PRODUCTS OF SDLC PHASES

- *Project Identification and Selection*: Priorities for systems and project, architecture for data, networks, hardware and Information System Management are the result of the associated system.

- *Project Initiation and Planning*: Detailed work plan for project, specification of system scope and high level system requirements, assignment of team members and other resources.
- *Analysis*: Description of current system, need to enhance or replace current system, explanation of alternative systems and justification of alternatives.
- *Logical Design:* Functional and detailed specification of all system elements (data, process, input and output).
- *Physical design*: Technical, detailed specifications of all system elements, i.e., programs, files, network, system software, etc. and acquisition plan for new technology.
- *Implementation*: Code, documentation, training programs and support capabilities.
- *Maintenance*: New version of software with associated updates of documents, training and support.

3.5 APPROACHES TO DEVELOPMENT

In the continuing effort to improve the systems analysis and design process, several approaches have been developed. Attempts to make system development less of an art and more of a science usually referred to as engineering techniques, are applied to system development. We will discuss prototyping, followed by introduction to joint application design and participatory design.

3.5.1 Prototyping

Designing and building a scaled down but fundamental version of a desired system is known as prototyping. A prototype can be built with any computer language or development tool to simplify the process. A prototype can be developed with some fourth generation languages (4GLs) such as query, screen and report design tools of a data base management system (DBMS), and with tools called computer aided software engineering (CASE) tools.

Using prototyping as a development technique, the analyst works with user to determine the initial or basic requirements of the system. The analyst then builds a prototype. When the prototype is completed, the user works with it and tells the analyst what they like and do not like about it .The analyst uses this feedback to improve the prototype and take the new version back to the user. This process is iterated until the users are satisfied. Two key advantages of the prototyping technique are the large extent to which proto typing involves the user in analysis and design and its ability to capture requirements in concrete rather than verbal or abstract form. In addition to being used stand-alone, prototyping can also be used to augment the SDLC.

Prototyping is a form of rapid application development or RAD. The fundamental principle of any RAD methodology is to delay producing detailed system design document until the user requirements are clear. The prototype serves as the working description of needs. RAD methodologies emphasize gaining user acceptance of the human system interface and developing core capabilities as quickly as possible.

3.5.2 Joint Application Design

In the late 1970s, systems development personnel at IBM developed a new process for collecting information system requirements and reviewing systems designs. The process is called Joint Application Design (JAD). The basic idea behind JAD is to bring structure to the requirements determination phase of analysis and to the reviews that occur as part of design. Users, managers, and system developers are brought together for a series of intensive structured meetings run by a JAD session leader who

maintains the structure and sticks to the agenda. By gathering the people directly affected by an Information System in one room at the same time to work together to agree on system requirements and design details, time and organizational resources are better put to use. An added advantage is that, group members are more likely to develop a shared understanding of what the information system is supposed to do.

3.5.3 Participatory Design (PD)

Participatory Design (PD) represents a useful alternative approach to the SDLCPD emphasizes the role of the user much more than other techniques do. In some cases, PD may involve the entire user community in the development process. Each user has an equal share in determining system requirements and in approving system design. In other cases, an elected group of users control the process. These users represent the larger community. Under PD, systems analysts work for the users. The organization's management and outside consultants provide advice rather than control. PD is partly a result of the role of labour and management in the workplace where labour is more organized and is more intimately involved with technological changes.

Check Your Progress 2

1. Designing and building a scaled down but fundamental version of a desired system is the process known as _____

2. Prototyping is a form of _____

3. Implementation includes _____

3.6 CASE STUDY

The problem is to computerize Library of XYZ College. In this library, all transactions are handled manually. Registers are maintained to record the details of the books, information about the members of the library and to manipulate the issue and return of books. The data entry and recovery procedures are all manual and this takes a lot of time and energy to browse through the pages of the register for locating the relevant information.

This current manual system of the library is very tough and time-consuming and chances of getting errors gets very high. This method is not trustworthy.
This problem can be solved in the following steps:

Project Initiation and Planning

The specific services provided by our project will, of course, differ from other projects. Understanding the reasons behind the development of this project gives an appreciation of what our project does.

The main objective behind this project is:

- To provide the user with an easy and fast interface.
- To see that information handling is very easy and fast.
- Easy updation and modification of data.
- The basic aim of the project is to automate the basic functions of the library:
    - To handle the Book Details.
    - To record and handle the Member Details.
    - To handle issue, return of books and to keep details about the books given for reading.

Analysis

Is it feasible to automate the system? The three major areas to determine the feasibility of project are given below:

- **Technical Feasibility:** The current level of technology can support the proposed system. The proposed software is enabled to meet all the objectives of the system and the output received would be more efficient. So, the project is technically feasible.

- **Economic Feasibility:** The proposed system needs to get hardware and software installed. The short-term costs are overshadowed by the long-term gains. The management in question can invest in the system and is in condition to pay for the cost of system's study, cost of employee's time involved in the study and the cost of development of software. Thus, project is economically feasible.

- **Operational Feasibility**: The current system faces a lot of problems which would be removed in the proposed system. The employees of the system will be free from the burden of the paper work and a lot of confusion. The employees are themselves interested in getting the manual system replaced by the automated one. The proposed system is user friendly. So, even a layman can use it. Thus, it is operationally feasible.

Design

Once it is found that the project development is feasible, Design has to be developed for the requirements listed in the analysis phase.

Data Dictionary

- A Data Dictionary is a catalogue of all elements in a system. It consists of data about data.
- It is a document that collects co-ordinates and confirms what specific data terms mean to different people in the team.
- It is important for the following reasons:
    - to manage the details,
    - communicate meaning,
    - document system features,
    - facilitate analysis, and
    - locate errors and omissions.

Consider a Library Information System. Our Data Dictionary record stores the following descriptions:

- ***Book_Details***

Stores information about books in the library. The table contains the following attributes:

Attributes	Stores
Book Id (primary key)	ISBN Number
Name	Name of the book.
Category	Category of the book.
Subject	Subject of the book.
Author	Author of the book.
Price	Price of the book.

Date	Date on which the book is added in the library.
Edition	Edition of the book.
Copies	Total no. of copies of the book present in the library.

- ***Book_Copy***

Stores information about various copies of a book available in the library.

Attributes	Stores
Book Id	ISBN Number.
Book Idg	Indent no. of the book whose multiple copies are present.

- ***Book_State***

Stores information regarding current status of the book.

Attributes	Stores
Book Id (primary key)	ISBN Number.
Status	It gives information about current status of the book. Status of a book can be –
	• I => Book is issued.
	• L => Book is lost.
	• P => Book is present.
	• M => Book is lost by member.
	• D => Book is deleted.

- ***Book_IR***

It gives information about the state of the book which is issued to a member.

Attributes	Stores
TID (primary key)	Identification no of issue / return transaction.
Book ID	ISBN Number.
MemID	Stores indent of member who issued the book.
Mode	Mode can be –
	• I => book is issued for home.
	• R => book is issued for reading in library.
State	State can be –
	• 0 => book is with the member.
	• 1 => member returned the book.
	• 2 => member lost the book.

- ***Book_Irdetail***

It gives information about issue/return and date/time of the issue/return of the book.

Attributes	Stores
TID	Identification number of the transaction of issue/return.
Issue_Date	Stores the date on which the book is issued.
Issue_Time	Stores the time of issue of the book to the member.
Return_Date	Stores the date on which a member returns the book.
Return_Time	Stores the time at which a member returns the book.

- ***Book_Delete***

It stores the information about the books that are deleted and the date on which it was deleted.

Attributes	Stores
BookId	ISBN Number.
Date	Stores the date on which the book was deleted.

- ***Book_Renewed***

It stores the information about the book, when it is brought back to the library and the date on which it was renewed.

Attributes	Stores
BookId	Stores the bookId of the book being deleted.
Date	Stores the date on which the book was resumed.

- ***Member_Details*** – It stores the information about members of the library

Attributes	Stores
MemId	Stores the ID of the member.
First	Stores the first name of the member.
Middle	Stores the middle name of the member.
Last	Stores the last name of the member.
Sex	Stores the gender of the member.
Address	Stores the address of the member.
City	Stores the city of the member.
State	Stores the state of the member.
Country	Stores the country of the member.
PIN	Stores the pin code of the member.
Age	Stores the age of the member.
Phone	Stores the phone number of the member.
Issue_limit	Stores the maximum number of books that can be issued to the member.
Date_of_Joining	Stores the date on which member enrols in the library.
Books_Issued	Stores the number of books issued to the member.

- ***Member_State***

Storing information about state of the member in the library.

Attributes	Stores
MemID	Stores the ID of the member.
State	It can have two values. • P => Member can access the library. • D => Member is deleted for library access.

- ***Member_Deleted***

It stores information about the member deleted and the date on which the member was deleted.

Attributes	Stores
MemID	Stores identification no of the member being deleted.
Date	Stores date on which member was deleted.

The Zero level DFD is depicted in Figure 3.2.

Figure 3.2: 0- level DFD of Library Information System (refer to Fig. 3.2 in unit-3.jpg)

Input Design

The input design of this project is as follows. Points considered for the design of 'easy to fill out' form are given below which conforms to the design of the project:

- Designing form with proper flow.
- Logical grouping of information.
- Labels holding suitable captions & textboxes to accept the data.
- Usage of other tools, such as radio buttons, checkboxes, combo boxes, etc. also serve purpose for the better recording, processing, storing and retrieval of information.
- The appearance of the form has been tried to be kept as attractive as possible to help in better and logical organization of details.
- Since we know good screen design like good form design is an important instrument for steering the course of work, our design of input is guided by the following six objectives:
    - Effectiveness
    - Accuracy
    - Ease to use
    - Consistency
    - Simplicity
    - Attractiveness.
- Our screens show only that data which is necessary for the particular action being undertaken.
- Screens are kept consistent by locating information in the same area each time a new screen is accessed.
- We have made it easy to move from one screen to another through the use of icons, which channels the way to other screens apart from direct access to screens through the main menu.
- Rather than jamming all data into one screen and cluttering up the screen, we have made use of multiple screens which add to the user appeal, thus are more productive and are prone to less errors.

Data Capture Information

- **Identification of data**

The identifying data item in each transaction record is called a "KEY". Therefore, details of member is identified by the unique code, the details of the books through a unique book code. Similarly, the issue / return transactions through the transaction code.

- **Details of the retrieval system**

This in with reference to the stored data that can be quickly retrieved from the system files. This is done when we perform search on a particular criterion to draw the records or details of the search parameters.

- **Output Design**

Users generally merit the system by its output. Thus, in order to create the most useful output, system analyst works closely with the user through the interactive process, until the result is considered to be satisfactory.

The objectives of the output design are:

- Serve the intended purpose.
- Output should satisfy the user.
- Assured output where it is needed.
- Output on time.
- Choose appropriate output methods.
    - Reports
    - Messages (on screen)
    - Document on help

Depending on the circumstances and the contents, the output may be displayed or printed. Output contents may originate from these sources:

- Retrieval from data stores.
- Transmission from a process or system activity.
- Directly From input source.

Keeping the above points in mind, we have taken best care to present our information with the most clear and readable output. Our details are convincing enough to make the decisions fast and accurate.

Our reports represent one feature of output to present the various details in discrete categories. These reports can be viewed on screen as well as can be kept as a hardcopy in the printed layouts. Our system produces following reports:

1. List of books
2. List of members
3. List of books issued
4. List of books returned.

- **Database Design**

The following are various entities along with attributes for the project:

- ***Book_Details*** – (BookId, Name, Author, Category, Subject, Price, Date, Copies, Edition).
- ***Book_Copy*** – (BookId, BookIDC).
- ***Book_IR*** – (TID, BookId, MemId, Mode, State).
- ***Book_TRDetail*** – (TID, Issue_Date, Issue_Time, Return_Date, Return_Time).
- ***Book_Delete*** – (BookId, Date).
- ***Book_Resumed*** – (Book_Id, Date).
- ***Book_State*** – (BookId, Status).

- ***Member_Details*** – (MemID, First, Middle, Last, Sex, Address, City, State, Country, Pin, Age, Phone, Issue Limit, Date_of_Joining, Books_Issued).
- ***Member_State*** – (MemID, State).
- ***Member_Delete*** – (MemID, Date).
- ***Member_Resumed*** – (MemID, Date).

After these steps, coding in any programming languages can be done and then the system will be tested against the requirements of the user. The tested system will be implemented either by direct conversion or by parallel conversion.

Check Your Progress 3

1. What are the phases involved in the analysis and development of the system?
   ……
   ……
   ……
2. What are the basic types of feasibilities used in system analysis and design?
   ……
   ……
   ……
3. "Implementation of system can be done in three different ways". Do you agree with this statement? Justify your answer.
   ……
   ……
   ……
4. What are various types of maintenance.
   ……
   ……
   ……

3.7 SUMMARY

In this unit, you learned about the basic framework that guides systems analysis and design, the systems development life cycle, with its seven major phases: project identification and selection, project initiation and planning, analysis, logical design, physical design, implementation, and maintenance. The life cycle has had its share of criticism, which you read about, and other frameworks have been developed to address the life cycle's problems. These alternative frameworks include: Prototyping (Rapid Application Development approach), Joint Application Design and Participatory Design.

3.8 SOLUTIONS/ ANSWERS

Check Your Progress 1

1. The first phase in the SDLC is called project identification. In this phase, the user identifies the need for a new or improved system. Information requirements of the organization as a whole are examined, and projects to meet these requirements are proactively identified. In project initiation, the formal, but

preliminary investigation of the system problem or opportunity at hand and the presentation of reasons as of why the system should or should not be developed by the organization is done. A critical step at this point is determining the scope of the proposed system.

2. During analysis phase, the requirements are determined. In this phase, analysts work with users to determine what the users want from a proposed system. This phase usually involves a careful study of any existing systems, manual or computerized that might be replaced or enhanced as part of the project. Next, the requirements are studied and structured according to their inter-relationship and eliminate any redundancies. After this alternative initial design is generated to match the requirements then these alternatives are compared to determine which alternative best meets the requirements.

   After analysis phase is complete, design of the system begins. The design consists of logical and physical design of the system. During design, the analysis converts the description of the recommended requirements into logical and then physical system specifications. Design occurs in two phases, viz., logical design and physical design.
   Logical design concentrates on the business aspects of the system.
   In physical design, the logical design is turned into physical or technical specifications.

3. If the proposed system is not feasible to develop, it is rejected at this very step. In this phase, feasibility study of the proposed system is performed. The Cost Benefit Analysis of the proposed system is prepared by the system analyst in this phase and if the cost of the proposed system in terms of time, money and resources outweigh the benefits of the system, then the proposed system is rejected.

Check Your Progress 2

1. Prototyping
2. Rapid Application Design
3. Coding, testing and installation

Check Your Progress 3

1. The following phases are involved in the analysis and development of the system:

   - Project identification and selection
   - Project initiation and planning
   - Analysis
   - Logical design
   - Physical design
   - Implementation
   - Maintenance.

2. The six basic types of feasibilities used in system analysis and design are:

   - Technical feasibility
   - Economic feasibility
   - Behavioural feasibility
   - Operational feasibility
   - Legal feasibility
   - Time feasibility.

3. Yes. The implementation of system can be done in three different ways. They are:

- Direct conversion
- Parallel conversion
- Phased conversion.

4. Types of maintenance are:

   - Corrective maintenance
   - Adaptive maintenance
   - Perfective maintenance

3.9 FURTHER READINGS

- Kendall & Kendall; *Systems Analysis and Design*; PHI; Fifth Edition.
- Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman; *System analysis and design methods*; Tata McGraw-Hill; Fifth Edition; 2001.

Reference Websites

- http://www.rspa.com
- http://www.ieee.org