is said to functionally independent of other modules if it possess *high cohesion* and *low coupling.* Funcionally independent means that the module has minimum or no interaction with other modules. A design is good when the modules are functionally independent of each other. It also helps in isolating errors, increase the scope for reuse, and makes easy to understand.



CHECK YOUR PROGRESS

Q 1: Choose the correct answer :

(i) The first step of the SDLC is

(a) System design (b) Preliminary investigation

(c) Development of software (d) None of these

(ii) The detailed study of the present system is referred to as

(a) System planning (b) Feasibility study

(c) System analysis (d) None of these

(iii) The person responsible for the analysis of the system is often called

(a) client (b) programmer

(c) user (d) analyst

(iv) Interviews, questionnaires, forms are required in .....of SDLC.

(a) System Design phase (b) Analysis phase

(c) Preliminary investigation phase (d) None of these

Q 2: Fill in the blanks:

(i) High cohesion and low coupling indicated modules are funcionally.....

(ii) Two important characteristics of a proper module decomposion are _____ and _____.

(iii) In _____ phase, SRS document is transformed into logical structure which contains the specifications that can be implemented in a progrmming language.

- Requirement Analysis
- Design
- Development
- System Testing
- Implementation
- Maintenance

In this section, we will be learning about the various stages that make the system's life cycle.

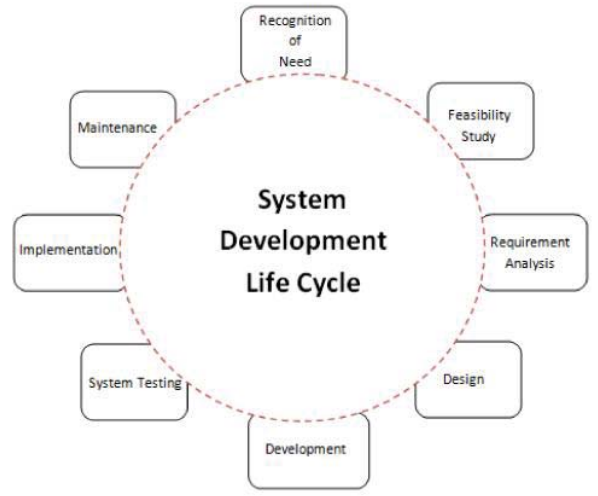


Figure 2.1: Various phases of System Development Life Cycle

2.4.1 Recognition of Need

The recognition of need is basically a survey or assessment of the existing system at the initial stage. This survey is used to determine whether a new alternative system is required or not. Recognition of need is sometime referred to as *preliminary investigation*. The main aim of recognition of need is to identify the problem. The idea of developing a new system arises when some problem appears. One must know what the problem is before going to solve it though it is difficult to identify the real problem. Without preliminary survey and clear understanding of the problem in the system, any further work done will lead to wastage of effort at a later stage. First, the need for the new or the enhanced system is

established. Only after the recognition of need for the proposed system is accomplished then only further analysis is possible. This may lead to a replacement of the existing system by developing a new computer based system. That means, if there is an issue for solving a particular task in the existing system, then it is to determine an alternative solution to that problem by developing a new system.

Detailed investigation is not actually perfomed at this stage. It is the role of a system analyst to obtain more crucial information by interviewing and meeting concerned people who are already working in that situation. This will clarify how the problem was identified, how often it occurs, how it affects the overall functioning etc. A proper understanding of the problem is necessary for the system analyst which enables to separate symptoms from the real causes. It also helps the analyst to consider it as a systems problem rather than other problem and thereby increasing the chances of success in proposing solutions. It will be more clear if we explain this with an example.

Let us assume, in an organization, all leave applications are processed manually. Now, this organization is recruiting many new people time to time. As the number of employees in the organization is increasing, so manual processing of leave application has also become very difficult. And the management is considering the option of automating the leave processing system. If this is the case, then the system analyst would need to investigate the existing system, find the limitations present, and finally evaluate whether automating the system would help the organization.

Once the need for new or improved system is recognised, all possible alternate solutions are chalked out. All these systems are known as ***candidate systems***. From all the candidate systems, the best alternative solution is selected, which is termed as the ***proposed system***. The proposed system is evaluated for its feasibility.

language.Thus, the operational phases are handled through program construction and testing, including a list of the programs needed to meet the system's objectives. The final report prior to the implementation phase is also known as *design document.* It includes flowcharts, layouts of record and reports separately, with a workable plan for implementing the candidate system. Information regarding manpower, hardware facililities, and their estimated cost are also included in this report.

A good software design should possess some characteristics like understandibility, correctness, efficiency, and maintainability. A design which is easily understandable is also easy to develop, implement, maintain, and change if required. Software design becomes clear when the design is modular, names of various design components are consistent and meaningful, modules are in a hierarchy. ***Modularity*** and ***decomposition*** should be properly maintained in a good software design. Decomposition of a problem into modules facilitates the design. The complexity of the design reduces if different modules perform different task independent of each other.

In the figure 2.2, a large program P is decomposed into set of three different modules M_1, M_2 and M_3. Specific tasks are performed by each of these three modules. A proper decomposition of the problem into module is very necessary for a good software design. There are two important terms associated with module decomposion : cohesion and coupling.

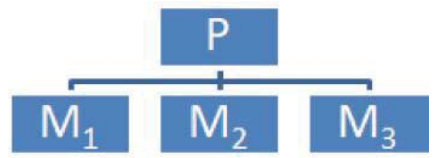


Figure: 2.2: A modular design

Cohesion is a measure of the functional strength of a module, on the other hand **coupling** is a measure of the degree of interdependence or interaction between the two modules. A module

2.4.4 Design

The purpose of the design phase is to plan a solution of the problem specified in the SRS document. It can be accomplished by transforming the customer requirements mentioned in the SRS document into a form that is implementable using a programming language. During the design phase different modules required to implement the design are identified, relationship among the identified modules are also determined. Data structures and algorithms for individual modules are also designed in this phase.

The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance. The design activities can be grouped into two categories: ***system design*** and ***detailed design***.

In ***system design***, the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules.

The first step in system design is to determine how the output is to be produced and what will be the format of this output. The second step is to design the input data as well as the database to meet the requirements of the proposed output. Thus, system design includes the design of user interface, system interface, network, databases, as well as the major modules in the system and their specifications. The output of this phase is the *design document*. This document is similar to a blue print or a plan for the solution, and it is used later during the implementation, testing and maintenance phase.

During ***detailed design*** further details of the data structures and algorithmic design of each of the modules is specified. The internal logic of each of the modules are decided. In this phase, the SRS document is transformed into logical structure which contains the specifications that can be implemented in a progrmming



Module
Module is a small part of the software that is responsible for performing any individual task. Sometimes, the term sub-program is also used to refer to the term module.

2.4.2 Feasibility Study

Feasibility study is a study, usually carried out by software engineers to determine whether conditions are right to implement a particular project. Feasibility study is an important stage in any system development process. Feasibility for a system means whether it is practical and beneficial to build that system. Feasibility is evaluated from the developer and customer's point of view. The developer sees whether they have the required technology or manpower to build the new system. It is also necessary to check that developing a new system really going to benefit the customer. Does the customer have the required money to build that type of a system? All these issues are covered in the feasibility study of the system.

The feasibility of the system is evaluated on the three main issues: ***technical feasibility***, ***economical feasibility***, and ***operational feasibility***. Another issue in this regard is the ***legal feasibility*** of the project.

Technical Feasibility :

Under technical feasibility, it is necessary to know whether the correct technology exists to support the software to be developed. In technical feasibility, availibility of both hardware and software resources are assessed so that the new product can be developed. Technical skills along with capabilities of technical team for the development of the new product is also analysed at this feasibilty study.

Economic Feasibility :

Under economic feasibility, costs and benefits are determined. Initially a detailed analysis is carried out for determining the overall cost in design and development of the new software product. It is also determined whether the product will be benificial for the organisation or not.

Legal Feasibility :

In this study, the feasibility is assessed in legal point of view. Privacy issues, data protection, authorization, copyright issues, social media laws and ethics etc. come under legal feasibilty study. All these issues and barriers are checked prior to development and implementation of the product.

There is one more issue of feasibility which is known as **operational feasibility** where it is observed that how a system can be maintained after being built.

Feasibility study helps software engineers whether to go ahead with proposed project if the project seems practically feasible or to stop the project if it is not feasible to develop. The outcome of feasibility study is usually a formal document or a report which contains the detail nature and scope of the proposed solution.

2.4.3 Requirement Analysis

The requirement analysis begins once the feasibility study has been completed and the proposed project is found to be finantially and technically feasible. In this phase, the emphasis is on identifying what is needed from the system and not how the system will achieve its goals. The main purpose of requirement analysis phase is to formulate the system's reruirements in a document format. This can be accomplished by analysing the expectations and requirements of the clients who will ultimately use the product.

In this phase, the relationships of the various system components among themselves and with environment are studied and understood. This task is complicated because there are often at least two parties involved in software development - *client* and *developer*.

Usually the developer initially does not understand the client's expectation from the product to be developed, and the client often does not understand the issues involved in software systems. This may lead to a communication gap, which has to be adequately

bridged during the analysis phase. Therefore, a few members of the developer team usually visit the client or customer site. The engineers who gather and analyze client requirements and prepare the requirement specification document are often known as **system analyst**. The system analyst communicate with different categories of persons to collect information about the process and their views for changing the existing process. This requires data collection from a variety of sources for which interviews, questionnaires, forms, study of existing document, records *etc. are necessary. Data flow diagrams(DFD), on-site observations* are also used in this phase.

After being analyzing the user requirements, the system analyst tries to sort out the anomalies, inconsistencies, and incompleteness. He then prepares the *Software Requirements Specification* (SRS) in a document format which is the final output of the requirement analysis phase.

The SRS document is reviewed by a internal project team to ensure that it is easily understanble, complete, consistent, and possess no ambiguity. The document is then given to the client for their approval after review process. After being approved by the client or customer, the proposed project goes for design and development. The SRS document sometime serves as a contract document between customer and the development organization.

Effort is required for writing the SRS document because the this document is needed by different people for various purposes. For example, users, customers, and marketing personnel may ask the SRS document to ensure the system to be developed described in the SRS will meet their needs. Again software developers may also refer to SRS document to make sure that the software that they are developing is exacly same what is required by the client. Sometime the engineers who will maintain the software may also require the SRS document to understand the functionality of the system.