



Chapter

2

SOFTWARE DEVELOPMENT PROCESS AND ITS STAGES

2.1 INTRODUCTION

Software development, a process consisting of two major steps of Software Engineering design, starts when management or sometimes Software development personnel feels that a new Software or an improvement in the existing Software is required. The Software development life cycle is thought of as the set of activities which analysts, designers and users carry out to develop and implement an information Software. To develop a successful information Software, the development process of an information Software within an organisation involves sequence of steps or events. An information Software is a Software that collects and processes data that can be further used by Software analyst for planning and decision making.

The stages involved in the Software development process are :

- Understanding a problem
- Deciding a plan for a solution
- Adding code to the planned solution
- Testing the program to which a code is added

An information Software can be designed efficiently when the development process is splitted into small phases.

2.2 VARIOUS PHASES OF THE SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

Software life cycle is an organisational process of developing and maintaining Softwares. It helps in establishing a Software project plan because it gives overall list of processes and sub-processes required for developing a Software.

Software development life cycle means combination of various activities. In other words, the various activities put together are referred to as Software development life cycle.

Software development life cycle includes :

- (a) Preliminary Investigation
- (b) Feasibility Study
- (c) Software Investigation
- (d) Software Analysis
- (e) Software design
- (f) Development of Software
- (g) Software testing
- (h) Implementation
- (i) Evaluation and Maintenance

Following are the phases of Software development life cycle :

2.2.1. Preliminary Investigation or Software Investigation or Problem Formulation

The Software investigation is the introductory step in a Software development project. In the Software investigation stage the request made by the user is handled. The request made by the user can be a request for changing, improving or enhancing the user's request. A user invites a Software analyst so that the problem can be easily defined and can be resolved later. This stage is not responsible for design study and details of the Software.

The substages involved in the Software investigation are:

2.2.1.1. Defining a Problem : A user calls a Software analyst so that he can help the user in defining and resolving the problem in the Software development process. The Software analyst identifies the problem and prepares a written statement of the objectives and scope of the problem. The problem can also be defined on the basis of interview or meetings with the user that helps the Software analyst to understand the problem more distinctively.

After the problem is defined, the written statement of the problem is sent to the user and the user gives response to these statements. On the basis of the response given by the user, the errors and misunderstandings regarding the problem are being resolved. That is why the proper understanding and definition of the problem is essential to understand the cause of the problem.

The Software problem may occur because of the following reasons :

- The Software is working slowly.
- The existing Software is unable to manage the workload.
- The existing Software may not be cost effective.
- The accuracy and reliability of problem may also arise.
- There can be a security problem.
- The information required in the Software development storage is not produced by the existing Software.

Many requests from employees and users in the organizations are not clearly defined. Therefore it becomes necessary that project request must be examined and classified properly before considering Softwares investigation.

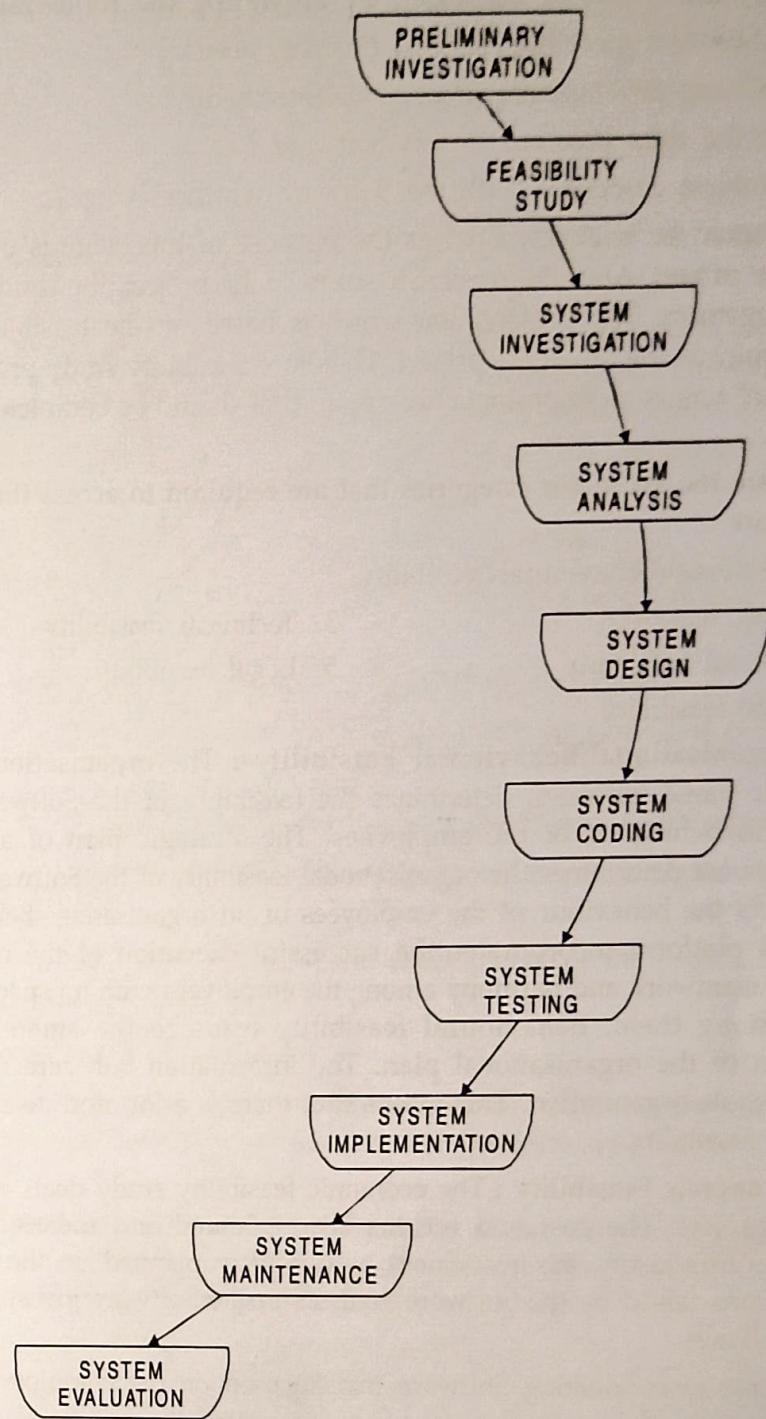


Fig. 2.1. Phases of System Development Life Cycle (SDLC)

2.2.2 Feasibility Study

An important outcome of the preliminary investigation is the determination that the Software requested is feasible. The feasibility of a project for the Software development process is thoroughly examined during the Software investigation stage. The objective of the feasibility study is to assure alternative Softwares so that the most feasible Software for the development process can be proposed.

2.4

- The feasibility study can be addressed by answering the following questions:
- Can this Software meet the required business needs ?
 - Is the Software development process suitable/useful ?
 - What are the risks involved in this Software ?
 - Is the problem associated with the Software worth solving ?

The study should be relatively brief as the purpose of this stage is only to get an idea of the scope of the project. After the feasibility study of the project, the result can be presented to the user management. The presentation which is based on the feasibility study marks a crucial decision point in the life of the project. Therefore feasibility study provides an overview of the problem and acts as an important check point that should be completed before executing other resources.

Following are the six major categories that are required to access the feasibility of the proposed Software :

1. Organisational/Behavioural feasibility.
2. Economic feasibility
3. Technical feasibility
4. Operational feasibility
5. Legal feasibility
6. Schedule feasibility

2.2.2.1. Organisational/Behavioural Feasibility : The organisational or behavioural feasibility as the name indicates, determines the feasibility of the Software in terms of the organisation and behaviour of the employees. The strategic plan of an organisation for information Software determines the organisational feasibility of the Software. The behavioural feasibility reflects the behaviour of the employees of an organisation. Behavioural feasibility on a borrowed platform incorporates the successful execution of the organisational plan, which involves teamwork and harmony among the employees with no space for discrimination and hatred among them. Behavioural feasibility leads to the smooth functioning and implementation of the organisational plan. The information Software must be taken as a subset of the whole organisation. This means that there is a lot more to study other than the organisational feasibility.

2.2.2.2. Economic Feasibility : The economic feasibility study deals with the economy of the Software project. The cost and returns are evaluated and therefore, it is determined whether the returns justify the investment, which were planned, in the Software project or not. The questions raised by the Software analysts in the Software investigation stage resolve the following issues :

- The cost of conducting Software investigation on the complete Software.
- The cost of hardware and software involved in the application of the project.
- The benefits such as reduced costs, improved customer service or improved resource utilisation.

2.2.2.3. Technical Feasibility : The major concern of technical feasibility is to observe whether the reliable hardware and software of the organisation meets the needs of the proposed Software or not. It also determines the requirements that can be developed in the required time.

In this study the following points are taken into consideration :

- Does the necessary technology acquire the proposed suggestions ?

SOFTWARE DEVELOPMENT PROCESS AND ITS STAGES

- Is the proposed technology capable of managing the data acquired by the new Software ?
- Does the proposed Software provide sufficient responses to the queries irrespective of the number of locations and users ?
- Can the Software be expanded or is it flexible ?
- Does the proposed Software provide technical security such as accuracy, reliability, accessibility and data security ?

2.2.2.4. Operational Feasibility : The operational feasibility is responsible for the operations of management, employees, customers and suppliers involved in a project it determines. The use and support of the proposed Software is that the operational feasibility examines the Software's operation while developing and installing the Software. The operational feasibility includes the following questions.

- Will the implementation of the project be done smoothly ?
- Will the management, employees, customers and suppliers provide the adequate support of the project ?
- Will the existing business methods be acceptable to the users ?
- Have the users been involved in the planning and development of the Software project ?

The proposed Software is assessed on the basis of the following categories :

1. Is Software being developed as per the rules, regulations, law, organisational culture and union agreements ?
2. Are the users actively participating in the development of the project ?
3. Is the Software legally feasible ?
4. Is the Software schedule feasible ?

2.2.2.5. Legal Feasibility : Legal feasibility of the Software refers to the viability of the Software. In other words, legal feasibility verifies whether the Software abides by all the laws and regulations or not.

2.2.2.6. Schedule Feasibility : The scheduled feasibility of the Software evaluates whether the Software finishes its task within the provided time of development or not. It is recommended for a Software to complete all the tasks well before the requirements.

2.2.2.7. Social Feasibility : The fact that social feasibility is a determination of whether a proposed project will be acceptable to the people or not. The determination typically examines the probability of a project being accepted by a group or directly affected by the group or directly affected by the proposed Software change.

2.2.2.8. Management Feasibility: It is a determination of whether a proposed project will be acceptable to the management or not. If the management does not accept a project or gives a negligible support to it, the analyst will tend to view the project as a non feasible one.

A project can be considered feasible only if the project proposal passes all the tests. In this stage, the infeasible projects are discarded unless these projects are resubmitted as new proposals.

2.2.2.9. Handling Infeasible Projects: It is not necessary that all the projects which are submitted for evaluation and review will be accepted. In general, requests that do not pass all the feasibility tests are not pursued further, unless they are modified and re-submitted as

new proposals. In some cases, it so happens that a part of a newly developed Software is unworkable and the selection committee may decide to combine the workable part of the project with another feasible proposal. In still other cases, preliminary investigation produces enough new information to suggest that improvements in management and supervision, not the development of information Softwares, is the actual solution to the reported problems.

2.2.3 Software Investigation

There are certain strategies for requirement determination that allows a Software analyst to understand the existing Software and also to determine the information requirement. Various strategies that solve the purpose are described below :-

The first step in the Software development life cycle is the preliminary investigation to determine the feasibility of the Software. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details, it rather, describes the business Software. In all respects, it is collecting of information that helps committee members to evaluate the merits of project request and make an informed judgement about the feasibility of the proposed project.

Analysts working on Software Investigation should accomplish the following objectives:

1. Clarify and understand the project requests.
2. Determine the size of the project.
3. Assess costs and benefits of alternatives approaches.
4. Determine the technical and operational feasibility of alternative approach.
5. Report the findings to management, with recommendations outlining the acceptance as rejection of the proposal.

The Software investigation can be performed primarily by the following methods:

2.2.3.1 Conducting the Investigation : The data that analysts collect during preliminary investigations are gathered through three primary methods :

- (a) Reviewing organisation documents
- (b) On site observations
- (c) Conducting Interviews

2.2.3.2 Reviewing Organisation Documents: It is essential for a Software analyst to learn about the organisation before involving in the project. It is also required to know the operations and management of the organisation. This can be examined by studying the organisation charts and the written procedures of the organisation. These procedures describe the ways in which the processes of the organisation operate. In addition, it identifies the steps involved in the operations performed in the processes.

2.2.3.3 Conducting Interviews : The written procedures do not provide the Software analyst with the views of the users for the current operations. The Software analyst needs to conduct interviews of the selected persons. These interviews allow the Software analyst to learn more about the nature of the Software project request. The Software analyst must be certain while addressing the problems of the users so that the purpose of the interview can be accomplished. This method provides details that can further help the Software analyst to understand the project economically, operationally and also technically.

The following format is recommended for the Software investigation process:-

1. Project title

2. Problem statement : It includes the statement of the problem in a concrete manner, possibly in new lines.
3. Project Objectives : It states objectives of the project defined by the problem.
4. Preliminary ideas : It provides possible solutions, if any occurring to a user or a Software analyst.
5. Project scope : It gives overall cost estimate.
6. Feasibility study : It indicates time and cost required in next step.

2.2.3.4. On Site Observations : Another important technique to collect data on site observation. In this method, the analysts observe the activities of the Software directly. The purpose of on-site observation is to get as close as possible to the real Software being studied. During on site observation, the analyst can see the office environment, workload of the Software and the users, methods of work and the facilities provided by the organization to the users.

2.2.4 Software Analysis

Assuming that a new Software is to be developed, the next phase is Software analysis. Analysis involves a detailed study of current Software, leading to specifications of a new Software. Analysis is a detailed study of various operations performed by a Software and the relationships within and outside the Software. During analysis, datas are collected on the available files, decision points and transactions are handled by the present Software. Interviews based on the observation and questionnaire are the tools used for Software analysis. Using the following steps, it becomes easy to draw the exact boundary of the new Software under consideration.

- Keeping in view the problems and new requirements.
- Workout the pros and cons including new areas of the Software.

All the procedures, requirements must be analysed and documented in the form of detailed data flow diagrams (DFD), data dictionary, logical data structures and adminiature specifications. Software Analysis also includes sub-division of complex process involving the entire Software, identification of data store and manual processes.

The main points to be discussed in a Software analysis are :

1. Specification of what the new Software is to accomplish based on the user requirements.
2. Functional hierarchy showing the functions to be performed by the new Software and their relationship with each other.
3. Function networks which are similar to function hierarchy but they highlight those functions which are common to more than one procedure.
4. List of attributes of the entities-these are the data items which need to be held about each entity (record).

2.2.5 Software Design

Based on the user requirement and detailed analysis of a new Software, the new Software must be designed. This is the phase of Software designing. The Software analysis phase answers the *What* question in the Software development process. The Software design also answers the *How* question. In other words, it specifies how the objectives of a Software project can be accomplished.

While designing a project, the Software mainly emphasizes on the following activities:

1. Preliminary or General design
2. User interface design
3. Data design
4. Process design
5. Structure or detailed design

2.2.5.1. Preliminary or General Design: In the preliminary or general design, the features of a new Software are specified. The cost of implementing these features and the benefits to be derived are estimated. If the project is still considered to be feasible one can move to the next stage.

2.2.5.2. User Interface Design : This activity focuses on designing the interactions between the users and the computer Softwares.

2.2.5.3. Data Desgin : This activity focusses on the design of the logical structure of the data base. It also emphasises on the files which are used in the proposed information Software.

2.2.5.4. Process Design : This activity focusses on the design of the software resources such as programs and procedures which are required in the proposed Software.

2.2.5.5. Structure or Detailed Design : In the detailed design stage, computer oriented works begins at its earliest. At this stage the design of the Software becomes more structured. Structure design is a blue print of a computer Software solution to a given problem being the same. Components and inter-relationship among the same components is the original problem. Input/Output and processing specifications are drawn up in detail. In this design stage, the programming language and the platform in which the new Software will run are also decided.

2.2.6 Development of a Software or Software Coding

After designing the new Software, the whole Software is required to be converted into a computer understanding language. It is done by Coding the new Software into computer programming language. It is an important stage in which the defined procedures are transformed into control specification with the help of a computer language. This is also called as the programming phase in which the programmer converts the program specifications into computer instruction, which is referred to as programs. The program coordinates the data movements and controls the entire process in a Software. It is generally felt that the program must be modular in nature. This helps in fast development, maintenance and future change if required.

2.2.7 Software Testing

Before actually implementing the new Software into operations, a best run of the Software is done by removing all the bugs, if any; it's an important phase of a successful Software. After codifying the whole programmes of a Software, a test plan should be developed and run on a given set of test data. The output of test run should match the expected results.

Using the test data following test runs are carried out :

- (a) Unit testing
- (b) Software testing

2.2.7.1. Unit testing : When the programs have been coded and compiled and brought to working conditions, they must be individually tested with the prepared test data. Any undesirable happening must be noted and debugged (error corrections).

2.2.7.2. Software testing : After carrying out the unit testing for each of the programs of the Software, when errors are removed, then Software test is done. At this stage the test is done on actual data. The complete Software is executed on the actual data. At each stage of the execution, the results or output of the Software are analysed. During the result analysis, it may be found that the outputs do not match the expected output of the Software. In such case the errors in the particular programs are identified and are further tested for the expected output. When it is ensured that the Software is running error free, the users are called with their own actual data so that the Software can be shown running as per the requirements.

2.2.8 Software Implementation

After the user accepts the new Software developed, the implementation phase begins. Implementation is the stage of a project during which theory is turned into practice. During this phase, all the programs of the Software are loaded into user's computer. After loading the Software, training of the users starts.

Main topics of such type of training are :

- How to execute the package ?
 - How to enter the data ?
 - How to process the data (processing details) ?
 - How to take out the reports ?

After the users are trained about the computerised Software, manual working has to be shifted to computerised working.

The following two strategies are followed for running the Software :

(a) Parallel run

(b) Pilot run

2.2.8.1. Parallel run : In such type of run, both the Softwares run parallel for a certain defined period for ex. computerised and manual are executed parallel. This strategy is helpful because of the following:

- Manual result can be compared with the results of the computerised Software.
 - Failure of the computerised Software at the earlier stage, does not affect the working of the organisation, because the manual Software continues to work, as it used to do.

2.2.8.2. Pilot run : In this type of run, the new Software is installed in parts. Some parts of the new Software are installed first and executed successfully for considerable time period. When the results are found satisfactory then only other parts are implemented. This strategy builds the confidence and the errors are traced easily.

2.2.9 Software Maintenance

Maintenance is necessary to eliminate errors in the Software during its working life and to tune the Software to any variations in its working environment. It has been seen that there are always some errors found in the Software that must be noted and corrected; it also means the review of the Software from time to time.

The review of the Software is due to :

- Knowing the full capabilities of the Software.
- Knowing the required changes or additional requirements.
- Studying the performance.

If a major change to a Software is needed, a new project has to be set up to carry out the change.

2.2.10 Software Evaluation

Evaluation of the Software is performed to identify its strengths and weaknesses. The actual evaluation can occur along any of the following dimensions :

- (a) Operational Evaluation
- (b) Organisational Impact
- (c) User Manager Assessment
- (d) Development performance

2.2.10.1. Operational Evaluation : It is the assessment of the manner in which the Software functions including ease of use, responsive, overall reliability and level of utilization.

2.2.10.2. Organisational Impact : It is the identification and measurement of benefits to the organisation in such areas as financial concerns, operational efficiency and competitive impact.

2.2.10.3. User-manager Assessment : It is the evaluation of the attributes of senior and user manager within the organisation, as well as end users.

2.2.10.4. Development Performance : Evaluation of the development process in accordance with such yardsticks as overall development, the aid effort, conformance to budgets and standards and other project management criteria.

The new project will then proceed through all the above life cycle phases.

2.3. A PRACTICAL EXAMPLE WHICH IMPLEMENTS THE SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) : STUDENT ADMISSION AND EXAMINATION SOFTWARE

Example. Anand School of Mathematics and Information Technology wants to automate its admission and examination Software for two years course of Masters in Business Administration (MBA). The main objective of developing this Software is to help the University to manage the database of students efficiently. This Software will maintain the electronic record related to personal and academic data of each student.

2.3.1 Problem Statement

The Problem Statement provides an outline of the Software from user's perspective. ASMIT offers IV-Semester MBA programme. This statement has three modules, namely, registration module, examination module, and result generation module.

2.3.1.1. Registration Module : To be a part of the University, an applicant must be registered, for which the applicant should pay the required registration fee. The fee can be paid through demand draft or cheque drawn from a nationalized bank. After successful registration an enrollment number is allotted to each student, which makes the student eligible to appear in the examination.

2.3.1.2. Examination Module : The examination of the MBA programme comprises of assignments, theory papers, practical papers and a project.

2.3.1.3. Assignments : Each subject has an associated assignment, which is compulsory and should be submitted by the student before a specific date. Each assignment carries 30 marks where student obtaining 40% or more (7-8 marks) is said to have passed.

2.3.1.4. Theory Papers : The Theory Papers can be core or elective. Core papers are mandatory papers, while in elective papers, students have a choice to select two out of three papers. Note that in first three semesters there are four core papers and three elective papers out of which two papers are to be chosen. Also the student is required to prepare a project in the IVth semester. Each theory paper carries 50 marks where student obtaining 40% or more (7-20 marks) is said to have passed.

2.3.1.5. Practical Papers : The Practical Papers are mandatory and every semester has three of them. Each practical paper carries 30 marks where student obtaining 40% or more ($>= 12$ marks) is said to have passed.

2.3.1.6. Project : Students need to submit a project in the IVth semester. The project carries 100 marks ($>= 50$ marks) is said to have passed. Also, students are required to appear for viva-voce session, which will be related to the project.

2.3.1.7. Result Generation Module : The result is declared on the University's website. This website contains mark sheets of the students who have appeared in the examination of the said semester (for which registration fee has been paid) Note that to view the result student can use enrollment number as password.

2.3.2 Data Flow Diagrams

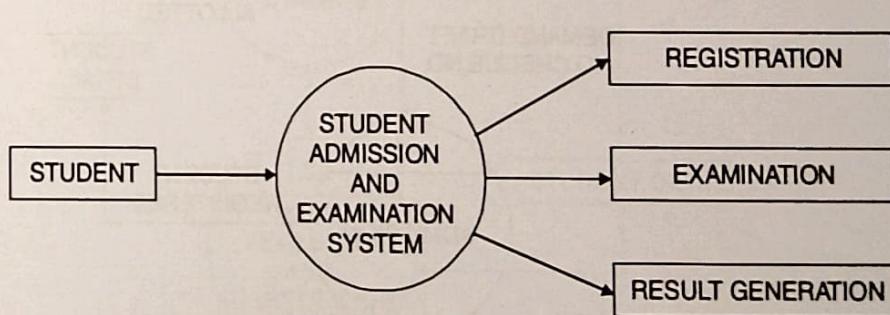


Fig. 2.2. Level 0 DFD

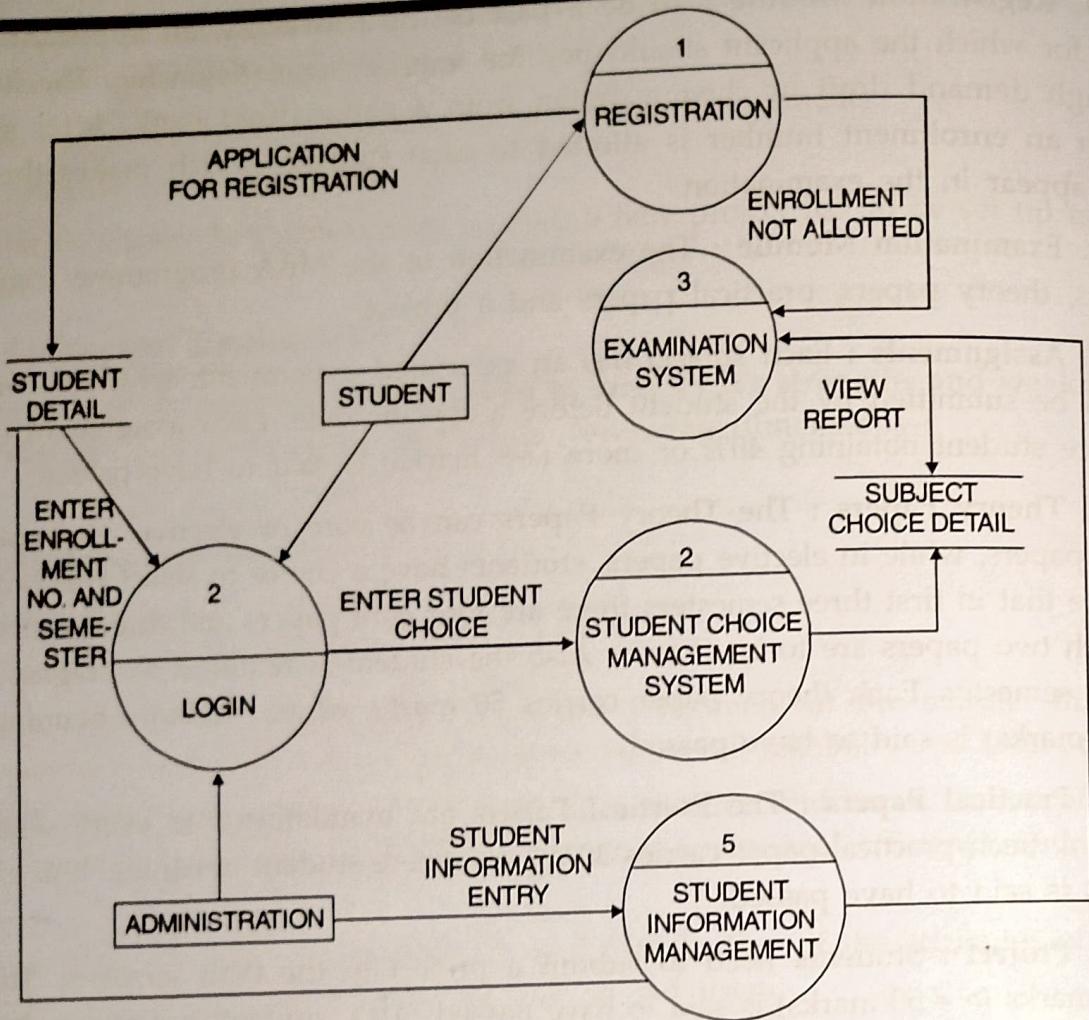


Fig. 2.3. Level 1 DFD of Student Admission and Examination System

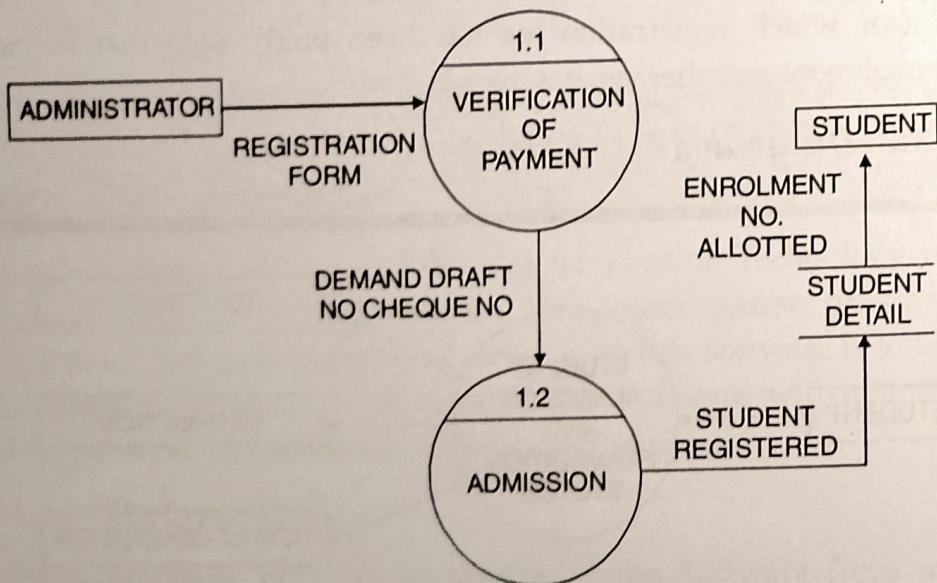


Fig. 2.4. Level 2 DFD of Registration

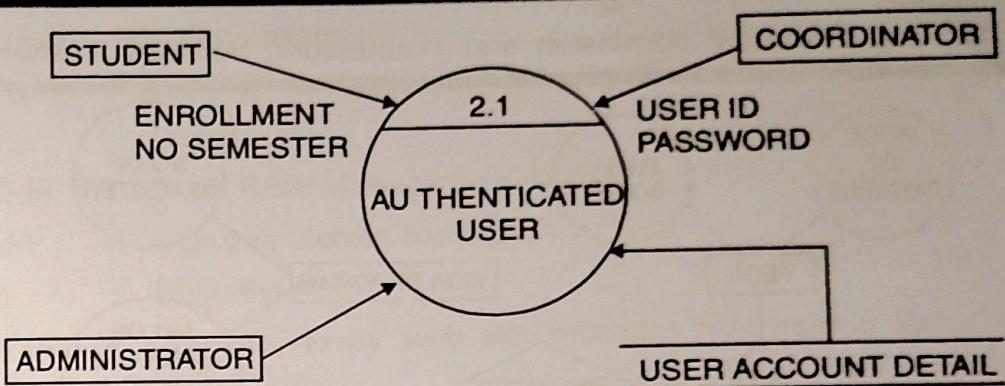


Fig. 2.5. Level 2 DFD of marks information system

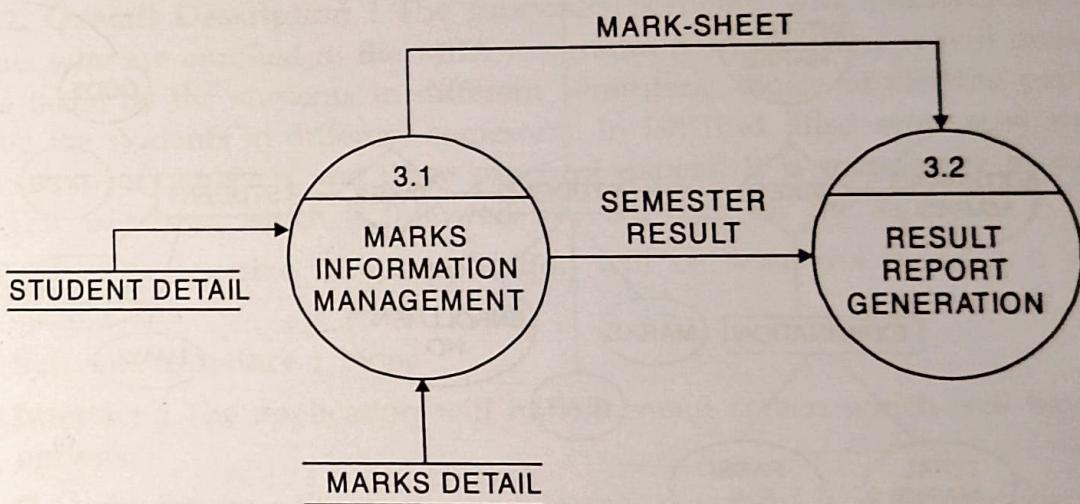


Fig. 2.6. Level 2 DFD of Examination

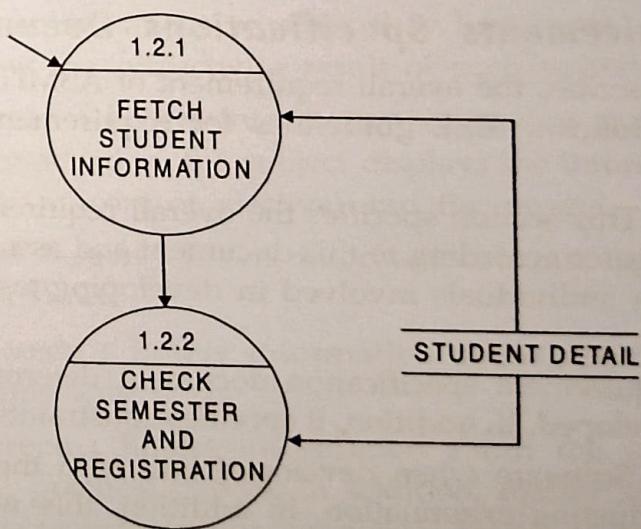


Fig. 2.7. Level 3 DFD Registration

2.3.3 Entity Relationship Diagram

The ER Diagram of student admission and examination Software is shown below:

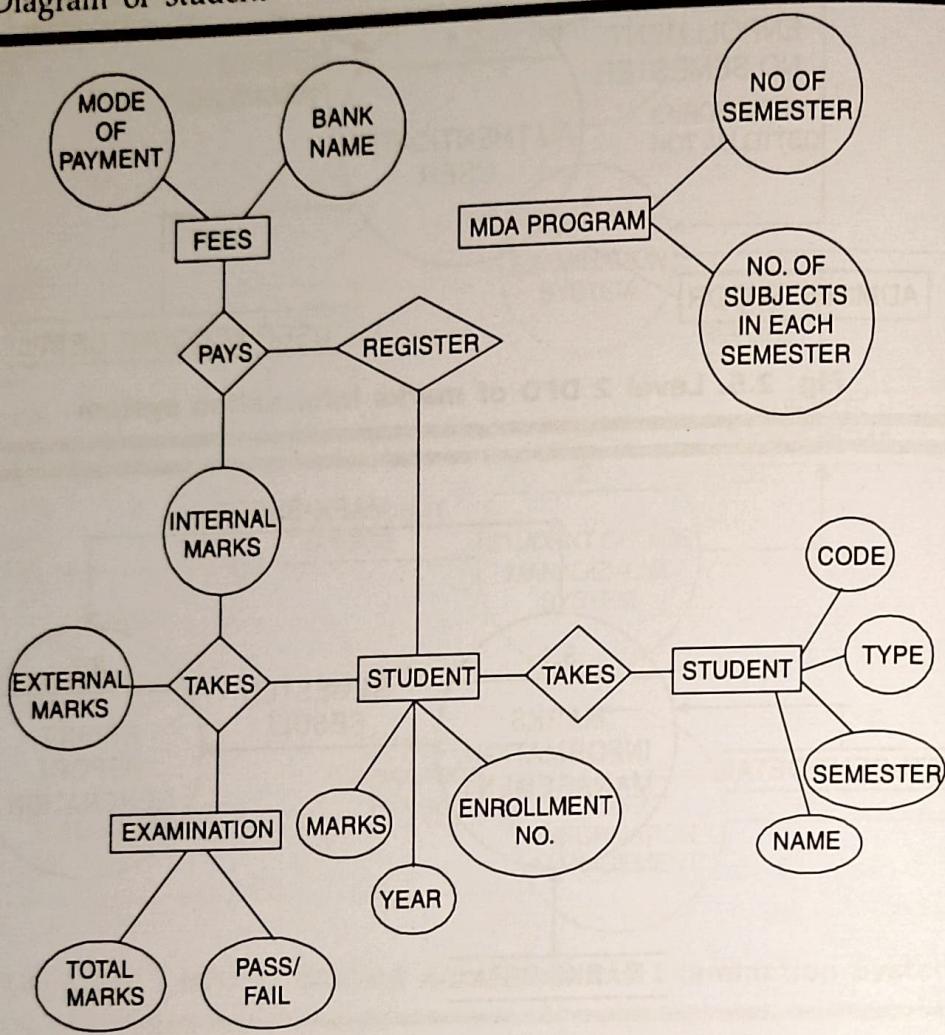


Fig. 2.8. ER Diagram of Student Management Software

2.3.4 Software Requirements Specifications Document

The SRS document describes the overall requirement of ASMIT to automate the proposed Software. This document follows IEEE guidelines for requirements specification document with some variations.

2.3.4.1. Introduction : This section specifies the overall requirements of the Software. The final Software will have features according to this document and assumptions for any additional features should be made by individuals involved in developing/testing/implementing/using this product.

- Purpose :** The requirement specification document determines the capability of the Software to be developed. In addition, it specifies constraints required by the Software.
- Scope :** The final Software when developed will help the university in registering students and conducting examination. In addition, this will manage the record of the subjects offered in different semesters, choice of elective papers and the marks obtained by them in different subjects in various semesters.

- (c) Definitions, acronyms and abbreviations : Following abbreviations are used in the entire specification document.

MBA : Masters in Business Administration
 DB : DataBase
 DBMS : Data Base Management Software
 RAM : Random Access Memory
 MB : MegaByte

- (d) References : The university web site provides information about the course, result and other information.

- (e) Overview : The SRS document provides description about the Software requirements, interfaces, features and functionalities.

2.3.4.2. Overall Description : The proposed Software will maintain information about the students who are enrolled in the MBA Programme. In addition, it will manage the record of subjects taken by the students in different semesters, choice of elective papers and marks obtained by the students in different semesters. In Ist, IInd, IIIrd semesters, students have to appear in six theory subjects and three practical papers. It is mandatory to submit a project report in IVth semester, which is followed by viva-voce for the same.

- (a) Product perspective : The application will be windows based and an Independent Software application.

(i) Software Interface : None

(ii) Interface : The application will have a menu screen which will have the following options:

- Login Screen** : Enters the user name, password and role (student, administrator and coordinator) Note that the role is defined to know the information about the individual (s) accessing the Software. This is essential to prevent the students from modifying the result in the database. Hence, the students will have access in the information about whether they have been successfully registered or not and can view the subjectwise result of each semester of year.
- Subject Screen** : Enters information regarding the subjects offered in different semesters. In addition, the subject displays the information about the assignments, subjects (that is core or elective) and the project.
- Examination Screen** : Enters information about registered students who seek to take examination.
- Student Screen** : Enters information about the student enrolled for MBA in different semesters.
- Master Screen** : Enters information about the marks of assignments, theory papers and practical papers. In addition, master screen displays the information of the subjects successfully completed. Marks of the student will be displayed in the form of printable marksheet which includes total marks and percentage of the student.

(iii) Hardware Interface : Screen resolutions with maximum 800×600 pixels should be used. It should also support output devices like printer.

(iv) Software Interface : The Software Interfaces that will be used for the proposed Software are listed below :

- Windows - based operating Software (such as window 95/98/XP/NT/2003/VISTA)
- Oracle 8i as the database management Software (DBMS) to store files and other related information.
- Crystal reports 8 to generate and view reports
- Visual Basic 6.0 as a front end tool for coding and designing the software.
- Internet Explorer 7 or higher, to view results of the examination on the internet.

(v) Communication Interface : None

(vi) Memory Constraints : Intel Pentium III processor or higher with a minimum of 256 MB RAM and 600 MB of hard disk space will be required so that Software performs its functions in an optimum manner.

(vii) Operations : The Software release will not include automated and maintenance of database. The university is responsible for manually deleting old/outdated, data and managing backup and recovery of data.

(viii) Site adaptations requirements : The terminals at the user's end will have to support the interfaces (both hardware or software) as mentioned above.

(b) Product Functions : The Software will allow access only to authorized users like student, administrator and coordinator depending upon the role. Some of the functions that will be performed by software are listed below :-

- Login facility for authorised users.
- Perform modification (by administration only) such as adding or deleting the marks obtained by the students.
- Provide a printable version of mark sheet (result) of the student.
- Use of 'clear' function to delete the existing information in the database.

(c) User Characteristics. None

(d) Constraints :

- As oracle 8i is a powerful database, it can store a large number of records.
- The university should have a security policy to maintain information related to marks, which are to be modified by the administrator.

(e) Assumptions and Dependencies :

- The subjects taken by the students in the semester will not change.
- The number of semester and elective subjects offered by the University will not change.

(f) Apportioning of requirements. Not required

2.3.4.3. Specific requirements : This section provides the information required by the developers of the Software.

(a) **External Interface :** This contains complete description of inputs and outputs from the software Software.

(b) **Functions :** None

(c) **Performance requirements :** None

(d) **Logical database requirements :** The information that will be stored in the database listed below :

- Student detail.** Stores information about student's enrolment number, student name, the year of enrollment, and fees details according to the semester.
- Subject choice detail.** Stores information about subject name, code and semester. In addition, it stores information about enrollment number, semester, and the subject chosen by the student.
- Marks detail.** Stores information about student's enrolment and the subject wise marks secured by the student.
- User account details** stores information about user name, password and role.

(e) **Design Constraints :** None

(f) **Software Software attributes :**

- (i) **Security :** The application will be password protected and hence will require users to enter their login ID (Username) and password.
- (ii) **Maintainability :** The application will be designed in a manner that is easy to modify the software Software later, when required and to incorporate new requirements in the individual modules such as subject information, marks information and user accounts.
- (iii) **Portability :** The application will be portable of any windows - based Software that has oracle 8i installed on it.

2.3.4.4. Change Management Process : In case the university desires to modify the criteria to select the elective papers as change the numbers of practical papers in each semester, then the changes will be updated and reflected in SRS document accordingly.

2.3.4.5. Document Approvals : When the requirement is gathered according to the user, SRS is then finally reviewed, approved, and signed by the developer and user (university). The SRS serves as a contract for software development activities.

2.3.4.6. Supporting Information : None.

2.4 SUCCESSFUL SOFTWARES

When the new Software you've worked on is implemented and running regularly, and you're assigned to another project, how will you know if you did a good job? How will you know if you've helped to produce a successful Software? A typical question often used in analysis and design examinations asks:

You're called in to evaluate the effectiveness of a recently implemented Software. What criteria would you use? Leaving aside project management considerations such as implementation to time, cost and quality you could ask the following questions:

- Does the Software achieve the goals set for it? Some of these will be operational running goals concerned with performance, some will be Software goals concerned with the production of outputs, and some will be business goals addressing the purpose of the Software development.
- How well does the Software fit the structure of the business for which it was developed? The new Software will no doubt have been developed based on an understanding of the then present structure of the organisation and some appreciation of how it might change in the future. However, it must not be an 'albatross Software' that hangs around the organisation's neck limiting its movement and freedom to reorganise. Softwares should be designed in a flexible way so that they can be changed to meet changing business conditions.
- Is the new Software accurate, secure and reliable? There will be basic requirements for financial control and auditing, but the Software should also be robust so as to continue in operation with degraded performance during partial failure. Security from unauthorised access has also now become increasingly important with the growth in the development of tactical and strategic information Softwares.
- Is the Software well documented and easy to understand? Increasingly large proportions of the budget of Software development departments are being used in the maintenance and updating of existing Softwares. The biggest single way of limiting these expenses in the future is to take account of it when we design today the Softwares of tomorrow.

This 'single-Software' view may help to identify the characteristics of successful Softwares, but it doesn't give a sufficiently wide framework for our analysis. We need to begin with an overview of the organisation as a whole. We can, for example, see the organisation in Softwares terms, operating within its environment and made up of a series of subSoftwares. A representation of this that has been widely used is shown in Figure 2.9.

It shows an industrial organisation with subSoftwares for:

- Marketing and purchasing: these are the main links with the environment as represented by customers and suppliers. It's important to recognise, however, that the environment also interacts with the organisation through legislation, social pressures, competitive forces, the education Software and political decisions.
- The production Software: this is concerned with transforming raw materials into finished products. It applies just as much in service organisations as in traditional manufacturing industry: an architectural drawing office is the equivalent of a motor engine assembly line.
- Support Softwares: these are shown as the accounting, personnel and management control subSoftwares.

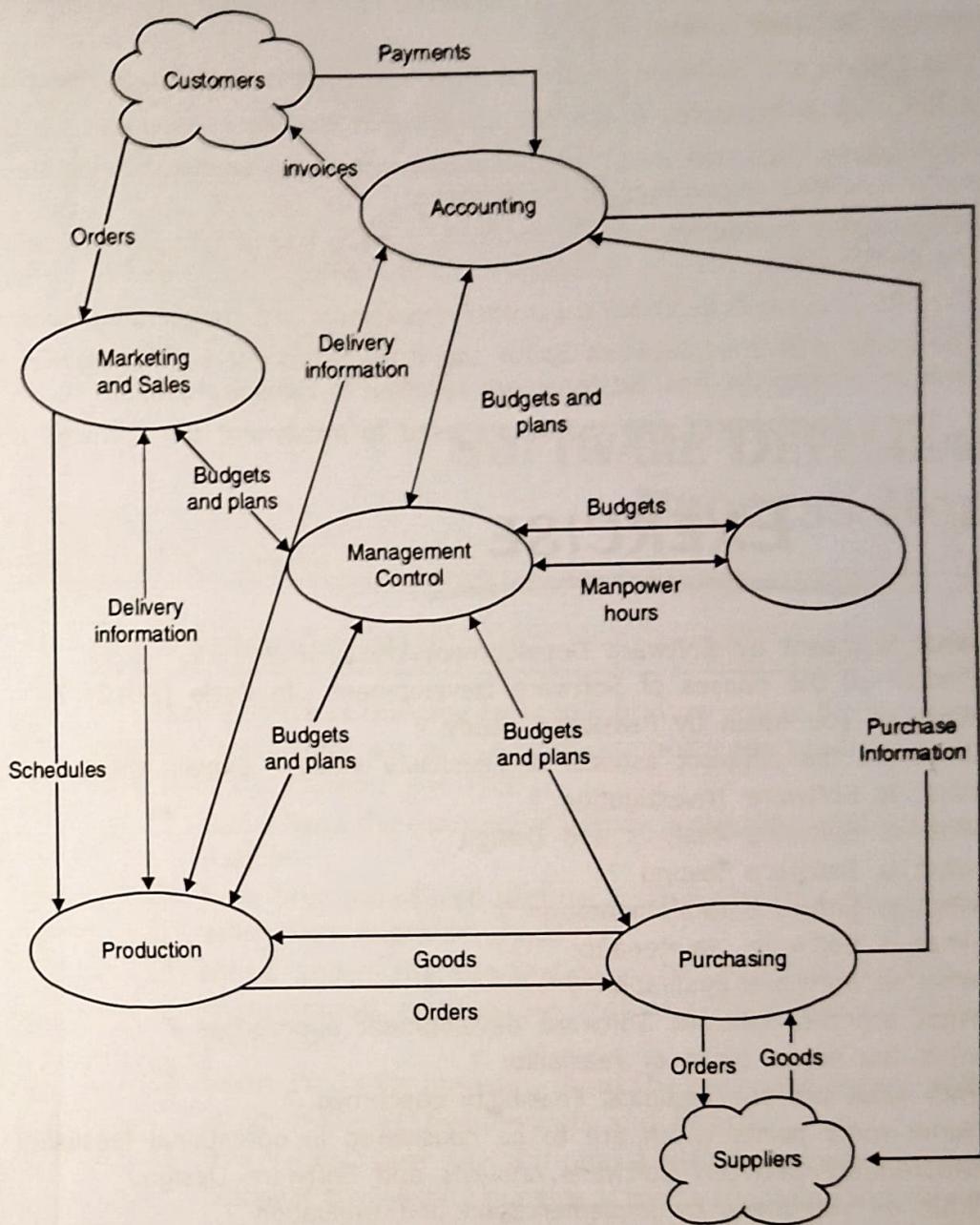


Fig. 2.9 The organisation as a system

POINTS TO REMEMBER

1. Software development includes Software analysis and Software design.
2. Software Life cycle is a process of developing and Maintaining Softwares.
3. Software Investigation to determine the feasibility of the Software is the primary step in a Software development project.
4. An analyst working on Software Investigation must clarify and understand the project request and determine the technical and operational feasibility of alternative approach.

2.20

5. After a Software is designed, it is converted into a computer understanding language through Software coding.
6. Unit Testing and Software Testing are the test runs carried out to test the data.
7. A Software is evaluated to identify its strength and weaknesses.
8. An Iterative Enhancement Model, the Software is development in increments to add some functional capabilities to the Software.
9. FGT specifies characteristics of software at high level and then automatically generates the source code.
10. The SRS Document describes the overall requirement to automate the proposed Software.
11. The major difference between Spiral and Waterfall model is that major risks involved with developing the final Software are resolved in the spiral model.
12. Software development approaches are used to implement the Software development process.

EXERCISE

1. What is meant by Software Development life cycle ?
2. Explain all the phases of Software Development Life Cycle (SDLC) ?
3. What do you mean by Feasibility Study ?
4. What are the different aspects of Feasibility study ? Explain them.
5. What is Software Investigation ?
6. What is Software Analysis and Design ?
7. What is Software Testing ?
8. What is Software Implementation ?
9. What is Software Maintenance ?
10. What is Software Evaluation ?
11. What are the different Software development approaches ?
12. What are seven types of Feasibility ?
13. With what are the technical Feasibility concerned ?
14. Name some points which are to be considered in operational feasibility.
15. Differentiate between Software Analysis and Software Design.
16. What do you mean by implementation and evaluation ?
17. Write a short note on :
 - (a) Problem formulation
 - (b) Software testing
18. What are the different stages involved in Software development process ?
19. What are the substages involved in Software Investigation ?
20. What questions are to be answered during feasibility study ?
21. What are the different methods of performing Software Investigation ?
22. What is the function of SRS document in SDLC ?
23. Draw an Entity Relationship Diagram of student admission and examination Softwares.
24. Differentiate between Unit Testing and Software Testing ?

