

COMPUTER SCIENCE

Grade: XII

SOFTWARE PROCESS MODEL



REFERENCE NOTE

Unit 1- Software Process Model (SPM)

Software Project Concept

Program:

The program is a sequence of instructions. It is the set or collection of instructions.

Instruction:

An instruction is a command given to the computer to perform a certain specified operation on given data.

Software:

A set of programs written for a computer to perform a particular task is called software or the logical components or set of procedures to routines or instructions are called software is the interface between the computer and the user. It is a group of commands that tells the computer what to do and how to do it.

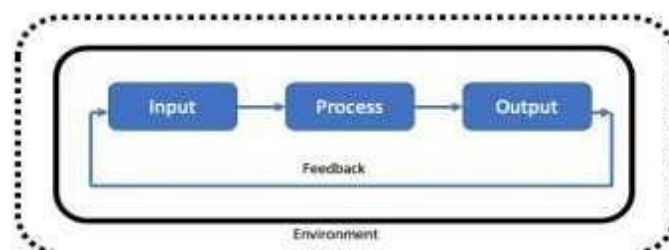
Project:

A project is a well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery).

Software Project Concept:

A software project is the complete procedure of software development from requirement identification to testing and maintenance, carried out according to the execution methodologies, in a specified period of time and budget in order to achieve intended software product.

Software Development Process



- Software development process defines a sequence of tasks that must be carried out to build new software.
- It groups the development activities into a sequence of phases.
- A phase in sequence can only commence on the previous phase has been completed.
- A report is produced at the end of each phase, describing what has been achieved and outlining the plan for the next phase.

Fundamental activities for the software development process are

1. **Software Specification:** The functionality or software and constraints on its operations must be defined.
2. **Software design and implementation:** The software to meet the specifications must be produced.
3. It must be validated to ensure that it does what the customer wants.
4. **Software evolution:** The software must evolve to meet changing customer needs.

System Development Life Cycle (SDLC)

System:

System is a set of interacting or interdependent components forming an integrated whole. A system can be described as a set of objects joined together for a common objective.

Development:

It is the process of step by step changing or growing of any program and system.

Information System:

Information system is a system which processes supplied/collected data and generates information that can be used for decision making at different levels.

SDLC

SDLC (Software/System Development Life Cycle) is an organized way to develop a software/system. System Development Phase or System Development Life Cycle or Software Development Life Cycle (SDLC) is a methodology used to develop, maintain, and replace software/information systems.

- *It is a systematic process of developing any software. It helps in establishing a system, or software or project, or plan. It gives an overall list of processes and sub-processes required for developing a system.*
- SDLC consists of a set of development activities that have a prescribed order. It is the development of software in chronological order.

System Development Life Cycle (SDLC), which is also known as Application Development Life Cycle, is a term used in system that describes the process of planning, creating, testing and deploying an information system.

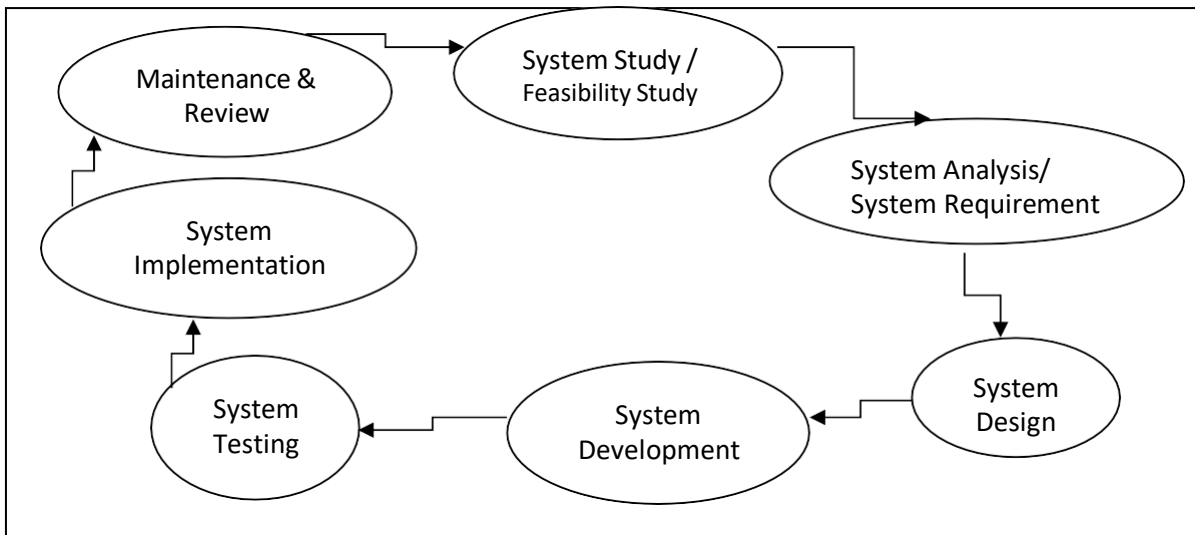
Importance and the necessity of SDLC

1. It helps to determine the needs of the user.
2. It supports constant communication between the developer and the user.
3. SDLC helps for easy identification of missing requirements.
4. It ensures that the software meets the needs of its users.
5. It supports proper analysis and design of the software.
6. It ensures proper development and testing.
7. Proper documentation support for future upgrade and maintenance.
8. It provides flexibility for adding features even after the software is developed.

SDLC Phases

The different **phases of SDLC** are as follows:

1. System Study or Preliminary Investigation and Feasibility study.
2. System analysis or Determination of system requirements.
3. System design.
4. System development or development of software.
5. System Testing.
6. System Implementation.
7. System Maintenance and Reviews or Evaluation.



1. **System study:**

A system is intended to meet the needs of an organization. Thus the first step in the design is to specify these needs or requirements. The top manager of the organization takes the basic decision to use a computer based (information) system for managing the organization.

During this phase, the development team focuses on completing three tasks:

- Survey the system by collecting the inputs from various sources.
- Analyzing the current system (manual or automated) in depth and developing possible solutions to the problem.
- Selecting to the best solution and defining its function with a feasibility study.

2. **System analysis:**

System analysis is the dissection of a system into its component pieces to study how those component pieces interact and work.

- System analysis is a term that collectively describes the early phases of development.
- It is defined as those phases and activities that focus on the business problem, independent of technology.

In this stage, the development team once again goes to the organization and studies very minutely to collect all the drawbacks and details of information from the users, management and data processing personnel.

Then the system analyst analyzes the information and proposes the following specifications.

- i. Goals and objectives of the proposed system.
- ii. Fundamental actions that must take place in the software.
- iii. Outputs to be produced.
- iv. Inputs to be used.
- v. Processes to be performed.
- vi. Interfaces to be provided.

- vii. Organizational and other constraints' to be met.
- viii. Performance requirements to be met.

Feasibility study:

Feasibility study is the most important activity in the system analysis phase. It analyses the proposed system from different aspects so that it makes us clear that how practical or beneficial the system will be to the organization. So it tells us whether the system is feasible to design or not.

Need of feasibility study

- It determines whether the system meets the goal of the clients or not.
- It determines the strength and limitations before starting to develop the system.
- It focuses on the boundary of the system's outline.
- It suggests new opportunities through the investigations process.
- It provides quality information for decision making.
- To provide documentation of the investigated system.

The different levels of feasibility study are as:

- 1) **Economic feasibility:** it concerns with cost effectiveness of the system. The main objective of economic feasibility is to calculate approximate cost—both the development cost and the operational cost and the benefits from the system.
- 2) **Technical feasibility:** it concerns with the availability of the hardware, software and the support equipment for the complete development of the system.
- 3) **Operational feasibility:** it concerns with smooth operation of the system. It is all about the problems that may occur during operation of the system after its development.
- 4) **Behavior feasibility:** it concerns with behavior of the users and the society towards the new system. Generally, most of the traditional employees are not easily ready to upgrade them with the new system.
- 5) **Social Feasibility:** It is a determination of whether a proposed system will be acceptable to the people or not.
- 6) **Management Feasibility:** It is a determination of whether a proposed system will be acceptable to management or not.
- 7) **Schedule(Time) feasibility:** it is the process of splitting project into tasks and estimate time and resources required to complete each task. It determines the deadline to complete a system and schedule the task accordingly.
- 8) **Legal feasibility:** it concerns with legal issue of the system. If the system is illegal then the system designing is meaningless. Everything is measured whether it is legal or illegal. It considers copyright law, foreign law, foreign trade, tax, etc.

System design:

The next step is to develop the logical design of the system. During this phase, the logic of the system, namely, the information requirement of users, and use this to find the necessary database.

System design is concerned with the design of new system. It involves designing of various things such as output design, input design, files design, processing and general program design etc.

Logical Design: Theoretically designing of the system is called logical design. The system could be designed on the basis of the requirements.

Physical Design: The conversion of logical design into designing tools and techniques is called physical design. It is more detail and complex jobs describing the solution of the problem. It uses

algorithms, flowcharts, pseudo codes, decision table, decision tree, E-R diagram, Data flow diagram etc.

Theoretically designing of the system is called **logical design**. The system could be designed on the basis of the requirements.

The conversion of logical design into designing tools and techniques is called **physical design**. It is more detail and complex jobs describing the solution of the problem.

To create the logical design different kinds of tools are used.

- | | |
|----------------------|---------------------|
| ➤ Algorithm | ➤ Decision Table |
| ➤ Flowchart | ➤ Decision Tree |
| ➤ Pseudo codes | ➤ Data flow diagram |
| ➤ Structured English | ➤ E-R diagram |

3. **System development:** after designing a logical diagram of a system then next step is to convert into program. This process is called system development. Flowchart, algorithm, Pseudo code, etc. are the outlines the procedures for taking the input data and processing it into usable output.
4. **System testing:** it is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. System testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.
5. **Implementation:** implementation involves testing the installed system, converting from the old system to the new one and training the users. This phase consists of implementation of the system into a production environment, and resolution of the problem identified in testing phase.
6. **Maintenance and review:** it begins after the system is implemented. Like any system, there is an ageing process that requires periodic maintenance of hardware and software. The content of the review will include objectives met, cost, performance, standards and recommendation.

System Analyst:

System analyst is person who is involved in analyzing, designing, implementing and evaluating computer-based information systems to support the decision making activities and operations of an organization.

A good system analyst is:

1. Understanding and commitment to the organization
 2. People skills
 3. Conceptual skills and
 4. Technical skills
- *A system analyst is information specialist. To be a system analyst, one must be knowledgeable about the technical aspects of analyzing, designing and implementing computer-based systems.*
 - *A system analyst is a person who conducts a study, identifies activities and objectives and determines a procedure to achieve the objectives.*

Designing and implementing systems to suit organizational needs are the functions of the systems analyst. One plays a major role in seeing the business benefits from computer technology.

An analyst is a person with unique skills. One uses these skills to coordinate the efforts of different types of persons in an organization to achieve business goals.

The characteristics (attributes) of system analyst are as follows:

1. Knowledge of organization.
2. Technical Knowledge.
3. Interpersonal Communication Skill.
4. Character and Ethics.
5. Problem-Solving Skill.
 1. Defining the problem
 2. Analyzing the problem

3. Considering Many alternatives

4. Evaluating many alternatives
5. Choosing the best alternatives
6. System analysis and Design skills

The roles of system analyst area as follows:

1. Change event
2. Investigator and event
3. Architect
4. Psychologist
5. Motivator
6. Intermediary and diplomat

Duties and Responsibilities of System Analyst

1. Defining Requirements
2. Prioritizing Requirements
3. Analysis and Evaluation
4. Solving Problems
5. Drawing up functional specification
6. Designing System
7. Evaluating System

SDLC (System Development Life Cycle):

Describes in Details:

1. System study or Preliminary Investigation :

In this stage, the development team studies the present and identifies the drawbacks. They interact with the users and gathers information from different sources to recognize the problems of present system.

2. System Analysis:

In this stage, the development team once again goes to the organization and studies very minutely to collect all the drawbacks and details of information from the users, management and data processing personnel.

Then the system analyst analyzes the information and proposes the following specifications.

- i. Goals and objectives of the proposed system.
- ii. Fundamental actions that must take place in the software.
- iii. Outputs to be produced.
- iv. Inputs to be used.
- v. Processes to be performed.
- vi. Interfaces to be provided.
- vii. Performance requirements to be met.
- viii. Organizational and other constraints' to be met.

3. Feasibility study:

Feasibility study is the most important activity in the system analysis phase. It analyses the proposed system from different aspects so that it makes us clear that how practical or beneficial the system will be to the organization. So it tells us whether the system is feasible to design nor not. Thus it is necessary before system design.

The different levels of feasibility study are as:

1. **Economic feasibility:** it concerns with cost effectiveness of the system. The main objective of economic feasibility is to calculate approximate cost-both the development cost and the operational cost and the benefits from the system.
2. **Technical feasibility:** it concerns with the availability of the hardware, software and the support equipment for the complete development of the system.
3. **Operational feasibility:** it concerns with smooth operation of the system. It is all about the problems that may occur during operation of the system after its development.
4. **Behavior feasibility:** it concerns with behavior of the users and the society towards the new system. Generally, most of the traditional employees are not easily ready to upgrade them with the new system.
5. **Schedule feasibility:** it is the process of splitting project into tasks and estimate time and resources required to complete each task. It determines the deadline to complete a system and schedule the task accordingly.
6. **Legal feasibility:** it concerns with legal issue of the system. If the system is illegal then the system designing is meaningless. Everything is measured whether it is legal or illegal. It considers copyright law, foreign law, foreign trade, tax, etc.

4. System Design:



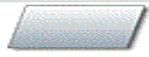


System design is concerned with the design of new system. It involves designing of various things such as output design, input design, files design, processing and general program design etc. This state consists of logical design and physical design of the system.

7. **Logical Design:** Theoretically designing of the system is called logical design. The system could be designed on the basis of the requirements.
8. **Physical Design:** The conversion of logical design into designing tools and techniques is called physical design. It is more detail and complex jobs describing the solution of the problem. It uses algorithms, flowcharts, pseudo codes, decision table, decision tree, E-R diagram, Data flow diagram etc.

System Design Tools:

The tools which are used to design the system in known as system design tools. They are used during system analysis and design phase of the system development.

1. **Algorithm:** An algorithm is defined as the finite sequences of instructions for solving a problem.
2. **Flowchart:** A flowchart is the pictorial representation of an algorithm which is classified into two types' system flowchart and program flowchart. The different symbols used in system flowchart are defined below:

Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

I) System flowchart:

System flowchart describes the internal architecture of a system that describes how data are moved inside the internal components of a system.

II) Program flowchart:

Program flowchart describes to solve the application types of real world problem.

3. DFD (Data flow diagram):

DFD is the logical diagram to describe the flow of data inside the components of system. It is easier to understand or grasp when being explained and most important to all, it is much more precise and less ambiguous than a narrative one. The main components are: process, data store, data flow, external entities.



4. Context Diagram:

It is combination of many other DFD. It is the highest level of DFD. It contains only one process, representing the entire system, the process is given the symbol circle. The external entities are denoted by rectangle. The flow of data is described by arrow.

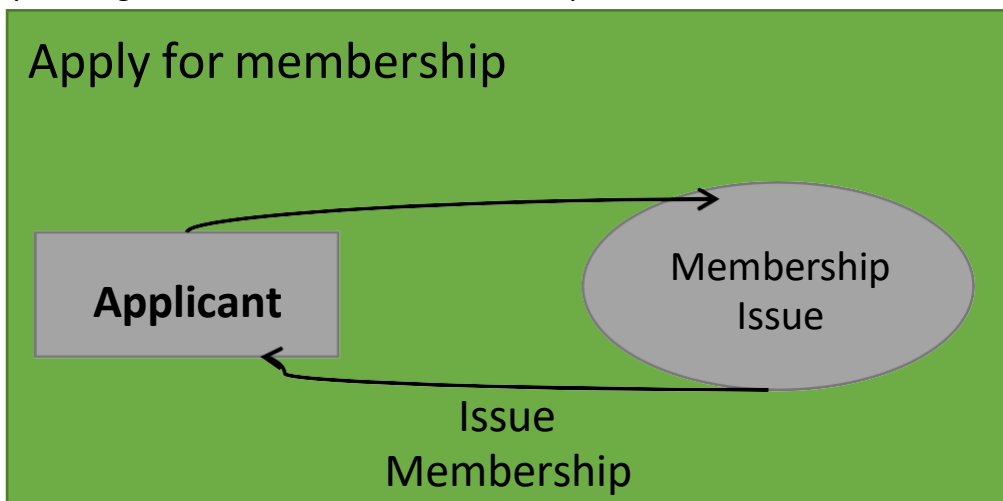


Fig.: The Context diagram of library membership

5. ER (Entity Relationship) diagram:

The E-R diagram is an overall logical structure of a database that can be expressed graphically. It was developed to facilitate database design and the simplicity and pictorial clarity of this diagramming technique have done great help in the designing part of database. The main components are attributes, entities and relationship.

The diagrammatic representation of entities attributes and their relationship is described by E-R diagram.

• **The main components of ERD are**

• Entity:



• Attributes:



• Relationship:



• Lines:

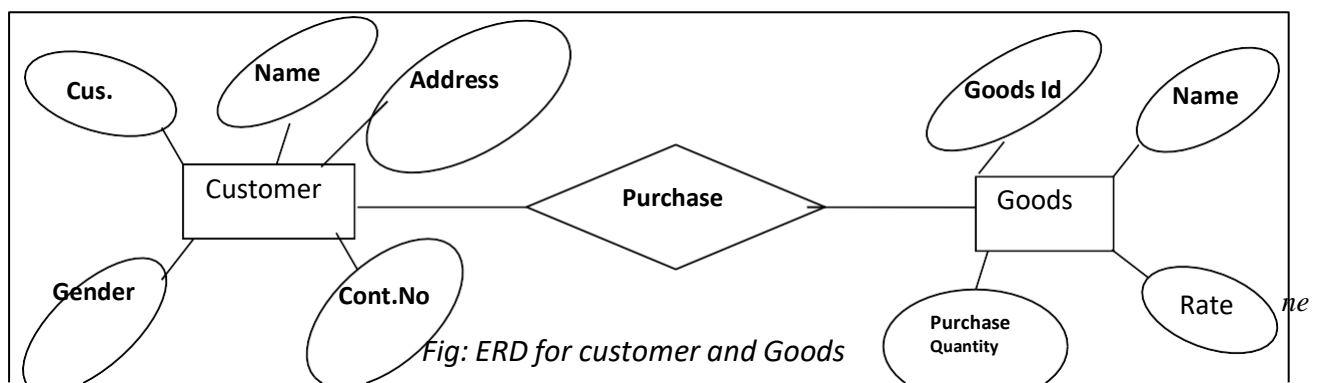


Fig: ERD for customer and Goods

6. Case diagram

Computer aided software engineering tool is automatic computer based program that helps for software engineering and SDLC process. It is very fast and effective tools for the development of big scale software. It helps in analysis, design, implementation, testing and maintenance.

7. UML

Unified Modeling Language is a standardized general purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the object management group. UML includes a set of graphic notation techniques to create visual models of object-oriented software.

8. Decision Table:

A table allows us to identify the exact course of actions for given conditions in tabular form. Decision table is a tabular representation of the logic of a decision, which specifies the possible conditions for the decision and the resulting actions.

Parts of Decision Table.

Condition Stub	Condition Entry
Action Stub	Action Entry

9. Decision Tree:

Decision tree is also a technique to represent condition and actions in a diagrammatic form in computer. A decision tree allows us to identify the exact course of actions for given conditions in tree structures.

10. Pseudo Code:

It is a kind algorithm for solving a problem and the instructions of pseudo code are written by using English phrase and mathematical expressions.

5. System Development:

Programmers begin to develop the program by using a suitable High Level Language. In System developments following processes are done.

- i. Convert logical structure in to programs in programming language.
- ii. Database is created.
- iii. User operational documents are written.
- iv. Users are trained.
- v. The internal documentation of a system is prepared.

6. System testing:

It is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. System testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

1. **White box testing:** white box testing of software is predicted on close examination of procedural details. Logical path through the software and collaborations between components are tested by providing test case that exercises specific sets of conditions or loops. It is used when the tester has access it the internal data structures and algorithms including the code that implement these.
2. **Black box testing:** black box testing treats the software as a black box without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary

value analysis, specification based testing, etc. it is also called functional testing because it tests whether a system is functioning or not.

7. Implementation:

Implementation involves testing the installed system, converting from the old system to the new one and training the users. This phase consists of implementation of the system into a production environment, and resolution of the problem identified in testing phase.

Types of Implementation:

1. **Direct Conversion:** All users stop using old system at the same time and then begin using the new. This option is very fast, less costly but more risky.
2. **Parallel conversion:** Users continue to use old system while an increasing amount of data is processed through the new system. Both the systems operate at the same time until the new system works smoothly. This option is costly but safe approach.
3. **Phased conversion:** Users start using the new system component by component. This option works only for systems that can be compartmentalized. This option is safe and conservative approach.
4. **Pilot conversion:** Personnel in a single pilot-site use the new system, and then the entire organization makes the switch. Although this option may take more time, it is very useful in big organizations where a large number of people make the conversion.

8. Maintenance and review:

It begins after the system is implemented. Like any system, there is an ageing process that requires periodic maintenance of hardware and software. The content of the review will include objectives met, cost, performance, standards and recommendation.

Types of Maintenance

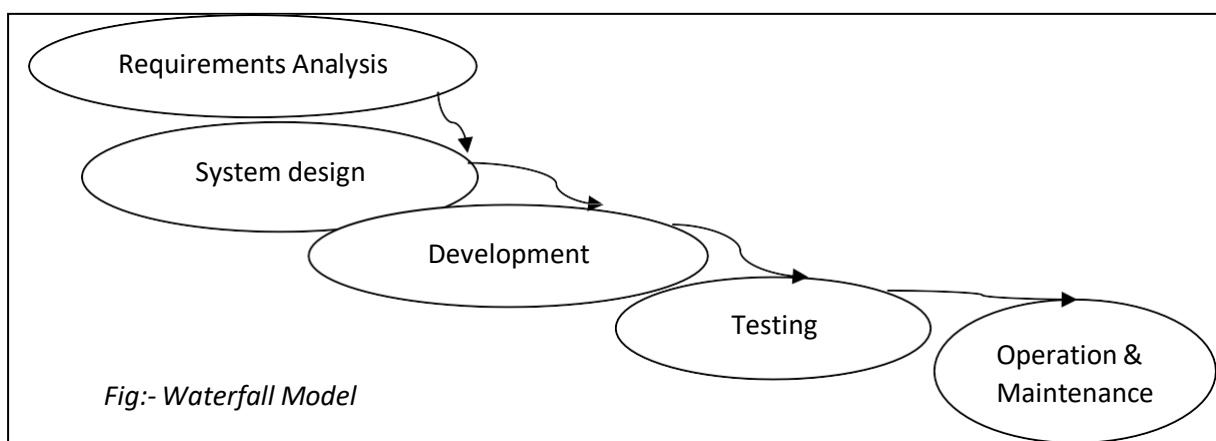
1. **Corrective Maintenance:** it corrects the run time errors during the operation.
2. **Adaptive Maintenance:** It modifies or adds new features in the system.
3. **Perfective Maintenance:** It makes the system perfect, up-to-date and improve the life of the system.

System Development Model:

During software development or system development for organizations, a common process framework is established, defining a small number of framework activities that are applicable to all software projects, regardless of their size complexity. For a better paradigm of a software process, several models are designed and implemented. It is the choice of system analyst which model is used to achieve the goal. The different models are:

1. Waterfall model:

Waterfall model is a systematic and sequential model to develop software that begins with requirements analysis to operation and maintenance. It describes a development method that is liner and sequential. It is an oldest type of model for software engineering. The fundamental processes of waterfall model are as follows:



1. **Requirements analysis and definition:** it is the first stage of waterfall model. In this stage, the developer should identify the actual requirements of the given problem.
2. **System design:** in this stage the systems design process partition the requirements to either hardware or software systems.
3. **System Development:** During this stage, the system design is converting into development.
4. **Integration and system Testing:** The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met.
5. **Operation and maintenance:** in this stage, the system is installed to the desire location. The maintenance involves correcting errors which were not discovered in earlier stages of the life cycle, improving the implementation of system units and enhancing the system's service as new requirements are discovered.

Advantages

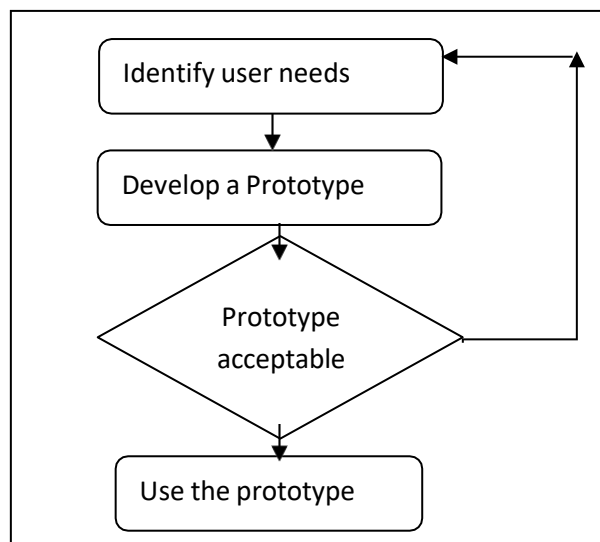
1. It is simple model suitable for small size project.
2. It is less expensive.

Disadvantages

1. It has no back track mechanism.
2. It is not suitable for large size project.
3. It has lack of proper documentation.

2. Prototyping model:

It is the iterative process of system development which is more appropriate for developing new system where there is no clear idea of requirements, inputs and outputs. These systems are then continuously modified until the user is satisfied.



1. **Identify the user needs:** the system analyst interviews the user to obtain an idea of what is required from the system.
2. **Develop a prototype:** the system analyst, working uses one or more prototyping tools to develop a prototype.
3. **Determine if prototype is acceptable:** the analyst educates the user in prototype use and provides an opportunity from becoming familiar with the system.
4. **Use the prototype:** the prototype becomes the operational system.

Advantages

1. The users get a better understanding of the system being developed.
2. Errors can be detected much earlier as the system is made side by side.
3. Quicker user feedback is an available leading to better solutions of the system.

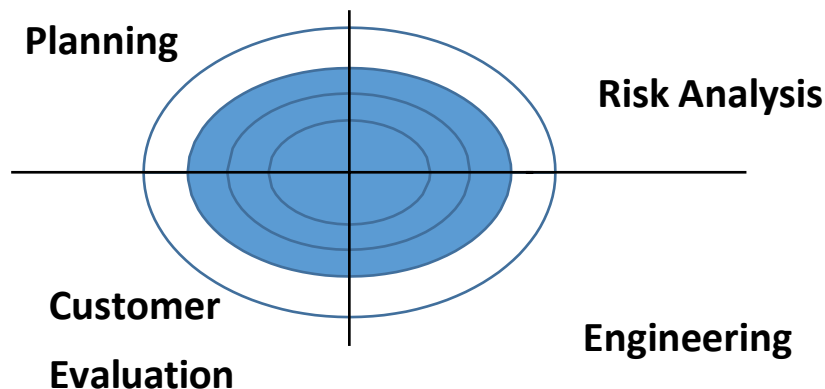
Disadvantages

1. It leads to implementing and repairing way of building systems.
2. It may increase the complexity of the system as scope of the system may expand beyond original plans.

3. **Spiral system:**

In this model, process is represented as a spiral rather than as a sequence of activities with backtracking. It is a software development process combining the elements of both waterfall and prototyping model. The spiral model is intended for large, expensive and complicated projects.

This is the most realistic model because it uses multidimensional approach for software development. The activities in SDLC are organized in a spiral structure that has many cycles which starts from the center of the spiral and goes out as it program and becomes matured. Each of the complete spiral segment is divided into four different attributes knows as



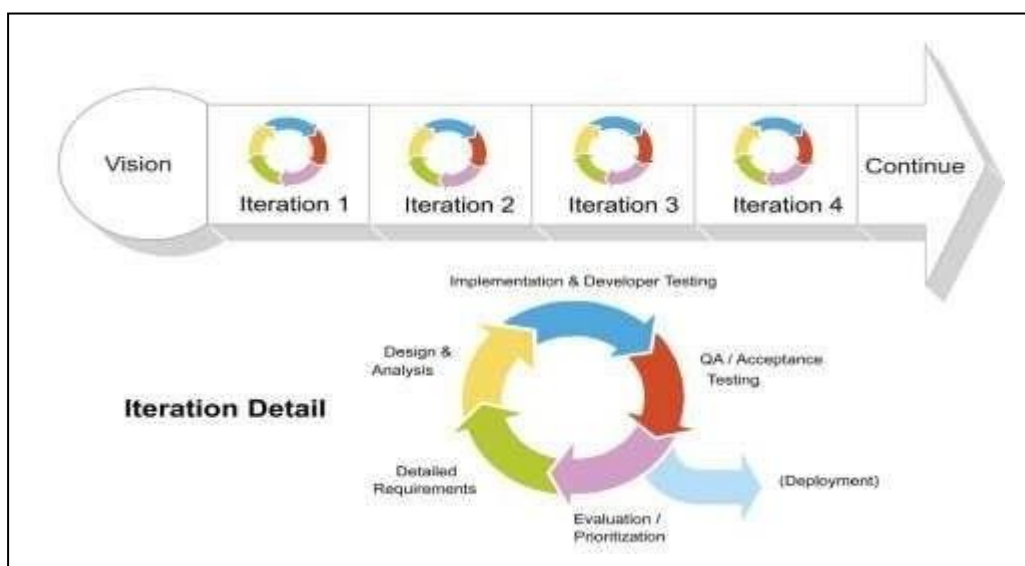
1. **Planning:** the project is reviewed and a decision made whether to continue with a further loop of the spiral. If it is decided to continue, plans are drawn up for the next phase of the project.
2. **Risk analysis:** for each of the identified project risks, a detailed analysis is carried out. Steps are taken to reduce the risk. For example, if there is a risk that the requirements are inappropriate, a prototype system may be developed.
3. **Software development (Engineering):** after risk evaluation, a development model for the system is chosen.
4. **User evaluation:** specific objectives for the phase of the project are defined by the evaluation of users. Constraints on the process and the product are identified. And a detailed management plan is drawn up. **Advantages**
 - It emphasizes quality.
 - It is effective for regular updating the system.
 - It emphasizes risk reduction techniques.

Disadvantages

- Full scale risk analysis requires training, skill so it may appropriate only for large projects.
- This model is relatively untested.

4. **Agile Software Development**

It is a software development method based on iterative and incremental development in which requirement and solutions evolve though collaboration between self-organizing, cross functional teams.



Documentation:

Documentation is the process of collecting, organizing, storing and maintaining a complete record of system and other documents used or prepared during the different phases of the life cycle of the system. It consists of the detail description about software requirements specification, feasibility report, and software designing report, description about input-output and processing mechanism, source code, comments, manuals, guides and effective help desk.

Types of Documentation

- 1. Internal Documentation:** It is used by the system analyst and the programmer during development process. It is very useful for the development period and also useful in future for the modification and maintenance for the software.
- 2. External Documentation:** It is used by the user during the running time of the software. It includes the detail description in terms of manuals, guide and help files. It is mainly deals with how to effective use of software.

Web References:

- <https://www.javatpoint.com>
- <https://www.w3schools.com>
- <https://www.tutorialspoint.com>
- <https://www.google.com>
- <https://www.wikipedia.org>