

Software Process Model

...

What Is Software ???

- Software is a set of programs written for a computer to perform a particular task.
- The **Software development process** defines a sequence of tasks that must be carried out to build new software.

Concept Of SDLC(Software/System Development Life Cycle)

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

Importance and the Necessity of SDLC

- It helps to determine the needs of the user.
- It supports constant communication between the developer and the user.
- It ensures that the software meets the needs of its users.
- It supports proper analysis and design of the software.
- It ensures proper development and testing.
- Proper documentation support for future upgrades and maintenance.

SDLC Phases

The different stages or 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.

SDLC phases



Explanation Of SDLC

- **1. System Study**

This is the first Step Of SDLC. This phase is also called as preliminary investigation or Need Analysis phase. This phase is basically concerned with determining whether or the new software/ system should be developed or not.

At the end of this phase, the team of system analyst comes to conclude with a new software to be adopted then the details plans and schedules are drafted for making new system through Feasibility study.

continue

- Feasibility Study : It is the determination of whether or not a software/system or project is worth doing or not. The process followed in making this determination is called a feasibility study.

Need of feasibility Study

- It determines whether the system meets the goal of the clients or not.
- It determines the strengths and limitations before starting to develop the system.
- It suggests new opportunities through the investigation process.
- It helps to increase investment in the system.

Types of feasibility Study

- **Technical Feasibility:** During the technical feasibility study, the system analyst identifies the existing computer systems of the concerned department and determines whether these technical resources are sufficient for the proposed system or not.
- It includes: It determines whether the technology needed for a system is available or not.
- It determines how the technology can be well used within the system.

Types of feasibility Study

- **Economical Feasibility:** It is the most important study that determines the cost and benefits of the proposed system.
- It includes: It concerns the returns of investment in the system.
- It determines whether the system's goals can be achieved within the allocated fund/budget.

Types of feasibility Study

- **Operational Feasibility:** It depends upon human resources for the development and implementation of the system.
- It includes: It determines the skills and capacity of the person who works for the project/system.
- It concerns with organizational structures.

Types of feasibility Study

- **Social Feasibility:** It is a determination of whether a proposed system will be acceptable for the people or not.
- There could be a cost to the environment, cost to society etc.

Types of feasibility Study

- **Management Feasibility:** It is a determination of whether a proposed system will be acceptable to the management or not. If it is not accepted by management then, the system is declared as non-feasible.

Types of feasibility Study

- **Legal and Contractual Feasibility:** It is a determination of whether a proposed system infringes(break) on known acts, or any legal areas.
- It includes : Legal effects like taxation, copyright, patent, trademarks, etc

Types of feasibility Study

- **Time(Schedule) Feasibility:** It is a determination of whether a proposed system can be implemented fully within estimated time.
- **Behavioral Feasibility:** It includes a study of organizational behavior.

Behavioral Feasibility includes:

It concerns the behavioral approach of the management staff and workers with in the organization.

It concerns fear of technology, fear of loss of job etc

Explanation Of SDLC

- **2. System Analysis :** System Analysis is a detailed study of various operations performed by a system and their relationships within and outside the system.
- During system analysis, data are collected from various sources like interviews, observation, questionnaires, sampling , research etc.
- It follows the feasibility study and involves a detailed study of the current system , leading to specifications of a new system.

Explanation Of SDLC

- **3.System Design** : Once the Analysis Phase is over, next comes the need to come up with the most accurate, robust, efficient and cost-effective architecture of the product that needs to be developed. Usually, more than one design is proposed in this phase, and the best one is selected based on different parameters such as robustness, durability, timeline, cost-effectiveness, and many more! The different design architecture is generally documented in Design Document Specification or DDS.

Explanation Of SDLC

- **4.System Development:** This phase is where the actual implementation of programming languages and different frameworks are being utilized for the development of the product. In this phase, all developers are involved. Developers are expected to follow certain predefined coding standards and guidelines; they are expected to complete the project modules within the defined deadline for the project. This phase is also the longest and one of the most critical phases in the Software Development Life Cycle. This phase is documented as a Source Code Document (SCD).

Explanation Of SDLC

- **5 System Testing:** Once the Development phase is completed, the next step is to test the developed software. The developed software is sent to the testing team, where they conduct different types of testing thoroughly on the software and look for defects. If any defect is found, the testing team records and document which is again sent back to the development team for error removal. This role is taken care of by Software Testers and Quality Analysts of the company. The testing team has to make sure that each component of the software is error-free and it works as expected.

Types of system Testing

- System Testing
 - 1. unit testing
 - 2 Integration testing
 - 3. System Test
 - 4. User Acceptance testing

Explanation Of SDLC

- **6. System Implementation :** After the testing phase is over, the first version of the software is deployed and delivered to the customer for their use.
- **7.System Maintenance:** Once the customer starts using the developed software, there is the scope of bug fixing that was not detected during the testing phase as when a large group of end-users starts using the software; there could be some probability that few boundary cases might have been missed. There is also scope for upgrading the software with newer versions and the latest security patches and technologies. And finally, there is also scope for enhancement of the software by adding more features into the existing software.

System Analyst

- A **System Analyst** is a computer specialist person who is involved in analyzing, designing, implementing and designing , and evaluating computer-based information systems to support the decision making and operations of an organization.

Roles & Responsibilities / Duties of System Analyst

- Defining Requirement
- Prioritizing requirement
- Gathering Data, facts and opinions of users.
- Analysis and evaluation.
- Solving Problems.
- Making a specification for system
- Designing System
- Evaluating System.

Characteristics/Qualities/Attributes of System Analyst

- Knowledge of Organization
- Technical Knowledge
- Interpersonal Communication Skill
- Problem Solving Skills
- System Analysis and Design skills.

System Analyst VS Software Engineer

Requirement Collection Methods



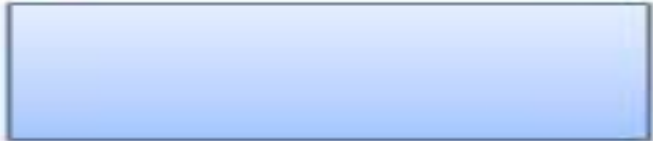




- Interview
- Survey/Questionnaire
- On-site Observation
- Brainstorming
- Document Analysis
- Prototyping
- Focus Group

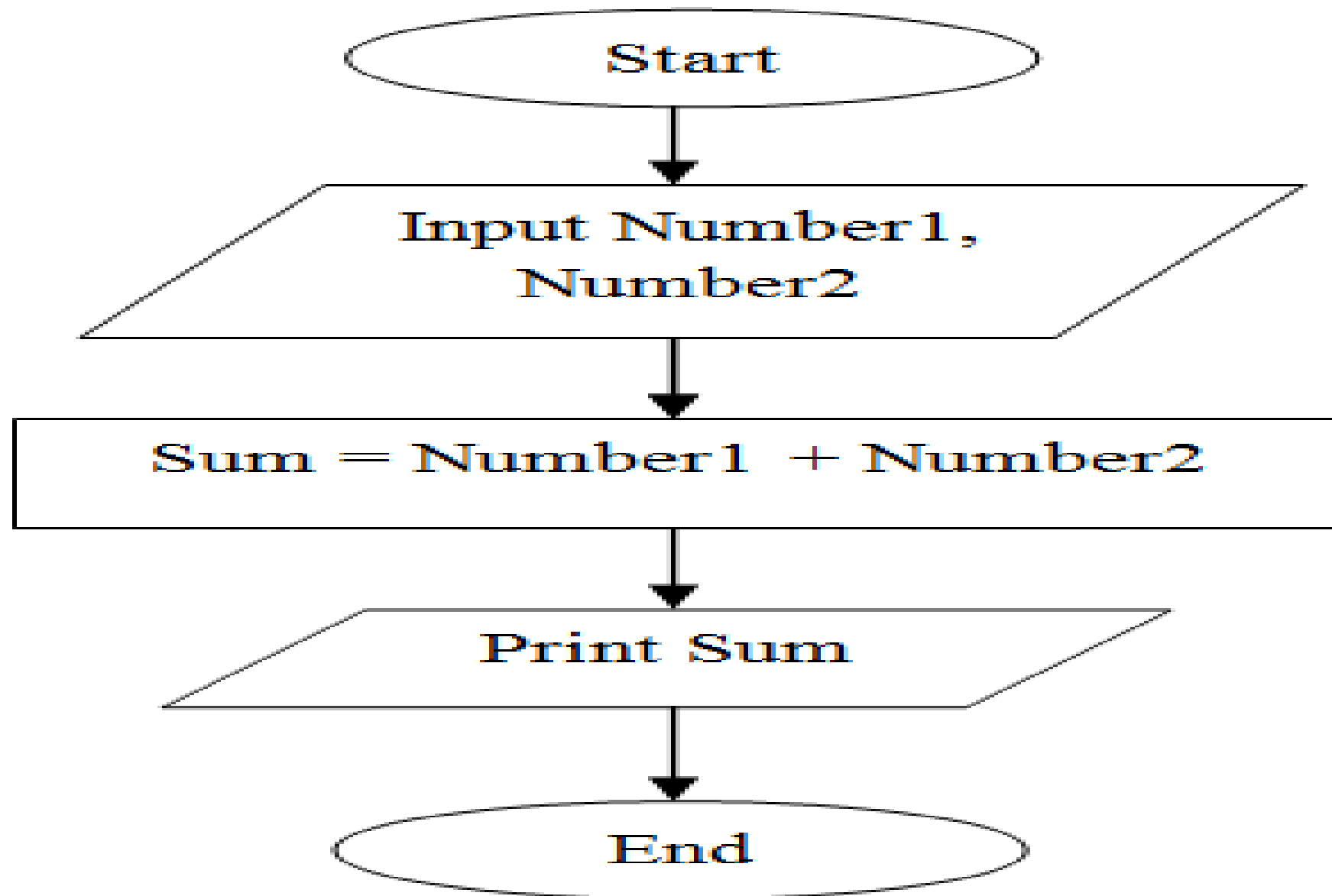
System Design (continue)

- **Flow Chart:** A Flowchart is a diagrammatic representation that illustrates the sequence of operations to be performed to arrive at the solution.
- There are two types of flowchart:
- They are
- **1. Program Flowchart**
- **2. System Flowchart**

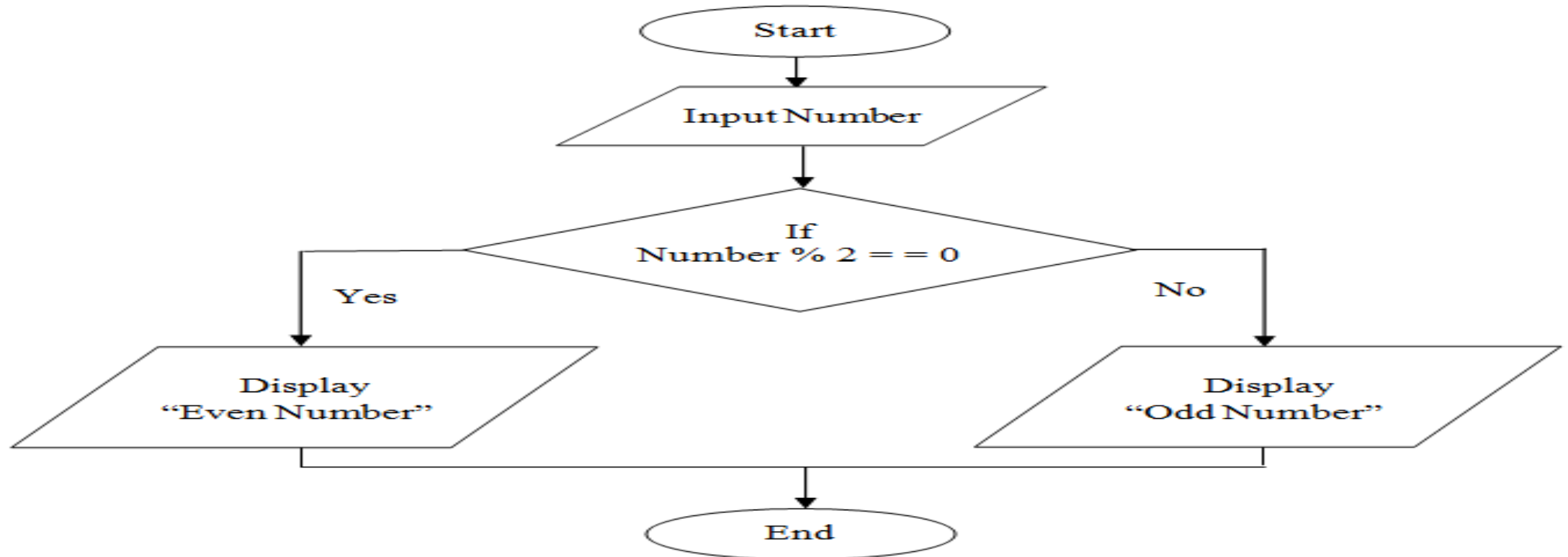
Program Flowchart

- A program flowchart is a pictorial representation of steps involved in solving a particular procedure or program. It is a design tool to check the flow of logic running in the program.

| <u>Symbol</u> | <u>Symbol Name</u> | <u>Description</u> |
|-------------------------------------------------------------------------------------|---------------------|-----------------------------------------------------------------------------------------------------------------|
|  | start/stop Terminal | This symbol is used to represent start and stop of the flow chart. |
|  | Input/output | This symbol is used to represent the input and output of the flow chart. |
|  | Processing | This symbol is used to represent the processing like arithmetic operations ,data assignments and movements etc. |
|  | Decision | This symbol is used to check whether condition is true(yes) or false(no). |
|  | Flow Lines(arrows) | This symbol is used to connect the symbols.it indicates the direction of the flow. |
|  | Connector | This symbol is used to connect the flow lines. |
|  | Off-page Connector | This symbol is used to continue the flow chart on Next Page (New page) |

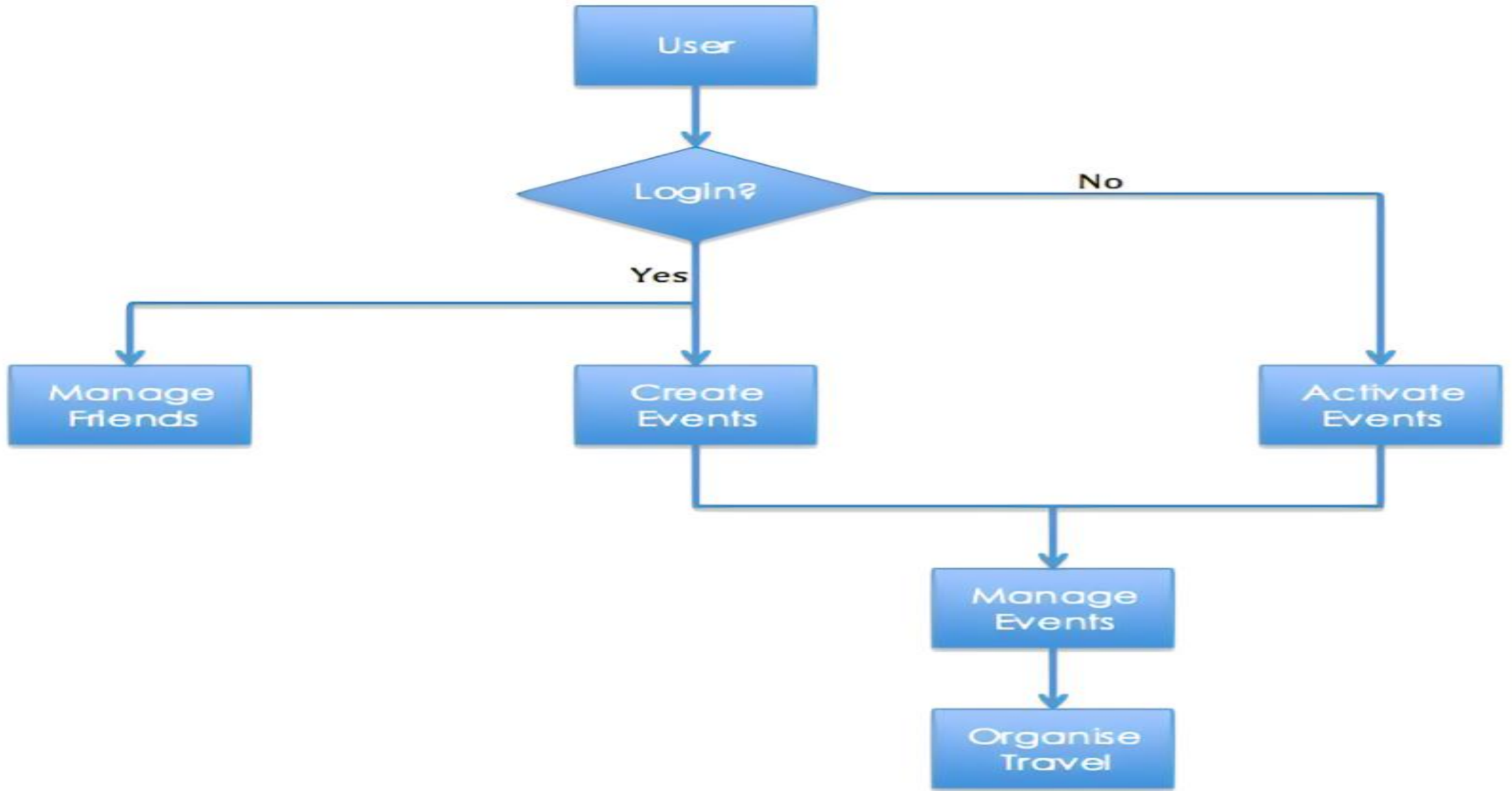


Flow Chart for odd or even



System Flow Chart

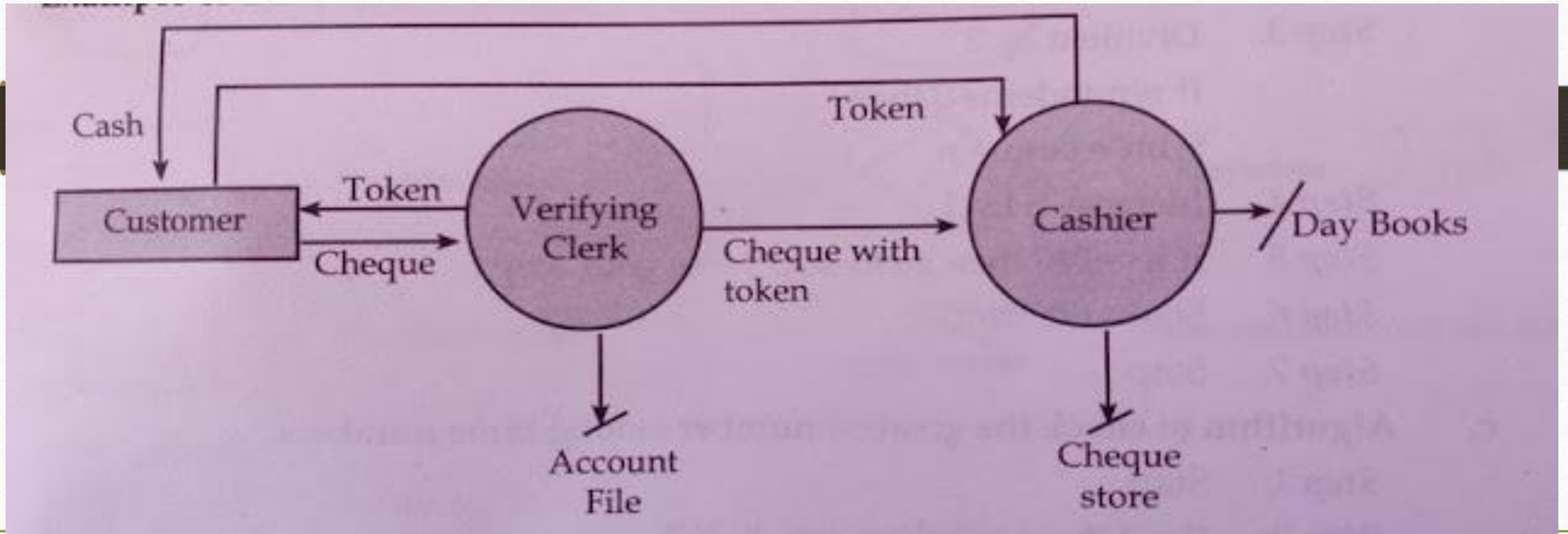
- A system Flow chart is a diagram that shows a broad overview of the data - flow and sequence of operations in a system.
- In next slide, System Flow Diagram for Event management system is shown.



Data Flow Diagram (DFD)

- A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the virtualization of data processing (structured design). A data flow diagram (DFD) illustrates how data processing (structured design). A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs.

DFD



ER-Diagram

Decision Table

- A decision table is a tabular method for describing the logic of the decisions to be taken. The decision table accompanies the flowcharts defining the possible outcomes. A decision table is an excellent tool to use in both testing and requirements management. Essentially it is a structures exercise to formulate requirements when dealing with complex business rules. Decision tables are used to model complicated logic.

Decision Table

| Conditions | TC1 | TC2 | TC3 | TC4 |
|-----------------------------|------------|------------|------------|------------|
| Request login | 0 | 1 | 1 | 1 |
| Valid user name entered | X | 0 | 1 | 1 |
| Valid password entered | X | X | 0 | 1 |
| Actions | | | | |
| Offer recovery credentials | 0 | 1 | 1 | 0 |
| Activate entrybox user name | 0 | 1 | 1 | 0 |
| Activate entrybox password | 0 | 0 | 1 | 0 |
| Enter privileged area | 0 | 0 | 0 | 1 |

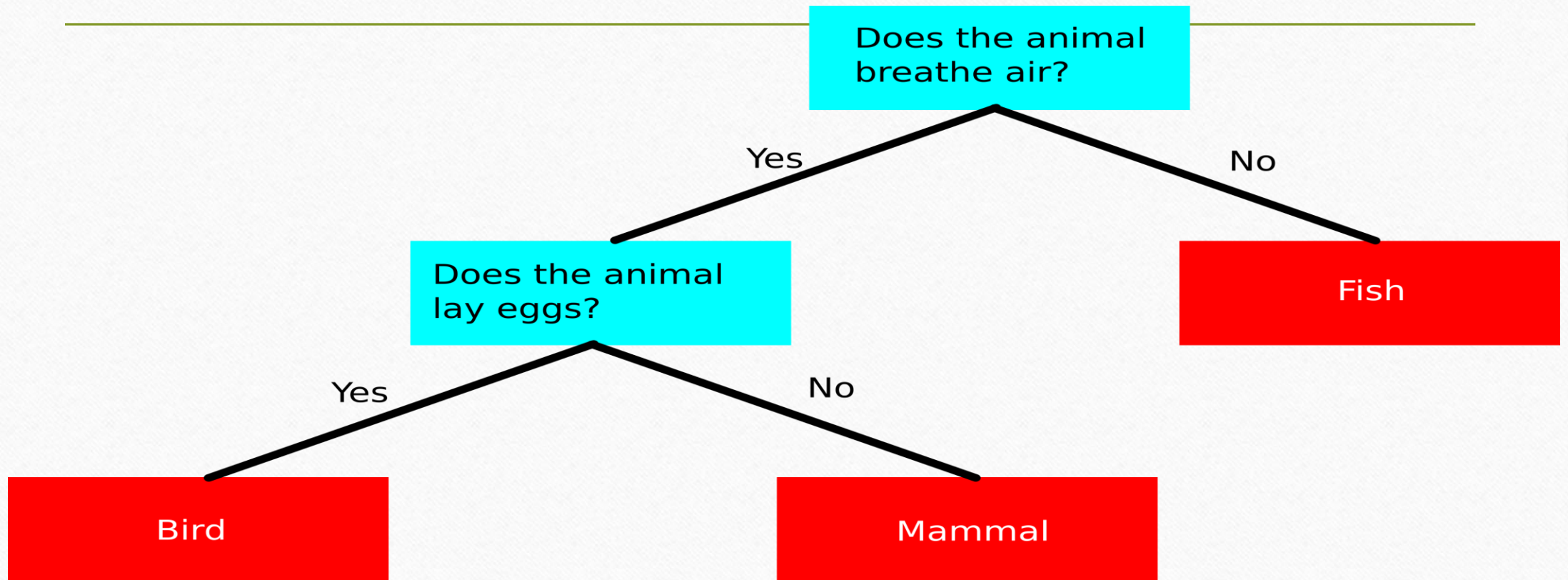
Decision Table

| Conditions | Rules | | |
|------------------|-------|---|---|
| Sun is shining | ✓ | | |
| Car is repaired | ✓ | ✓ | x |
| Music is playing | | ✓ | |
| Actions | | | |
| Driving | ✓ | | |
| Staying at home | | | ✓ |
| Dancing | | ✓ | |
| Repairing a car | | | ✓ |

Decision Tree

- Decision tree provides a graphical representation of decision logic that helps non-computer people find it easy to understand. The principles for the development of decision trees are relative forward. A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm.

Decision Tree



Software and Quality

- Two main approaches to software quality
- **1.Defect Management:** A software defect can be regarded as any failure to address end user requirements. It is based on counting and managing defects. Common defects includes misunderstood requirements, errors in designs, logic etc and coding errors.
- **2.Quality Approaches:** It is based on fixed quality models like ISO/IEC 25010:2011

2. Quality Approaches(continue)

- Its characteristics are as follows(There are 8 characteristics.)
- Functional suitability
- Reliability
- Operability
- Performance efficiency
- Security , Compatibility, Maintainability, and Transferability.

Qualities of a Good software

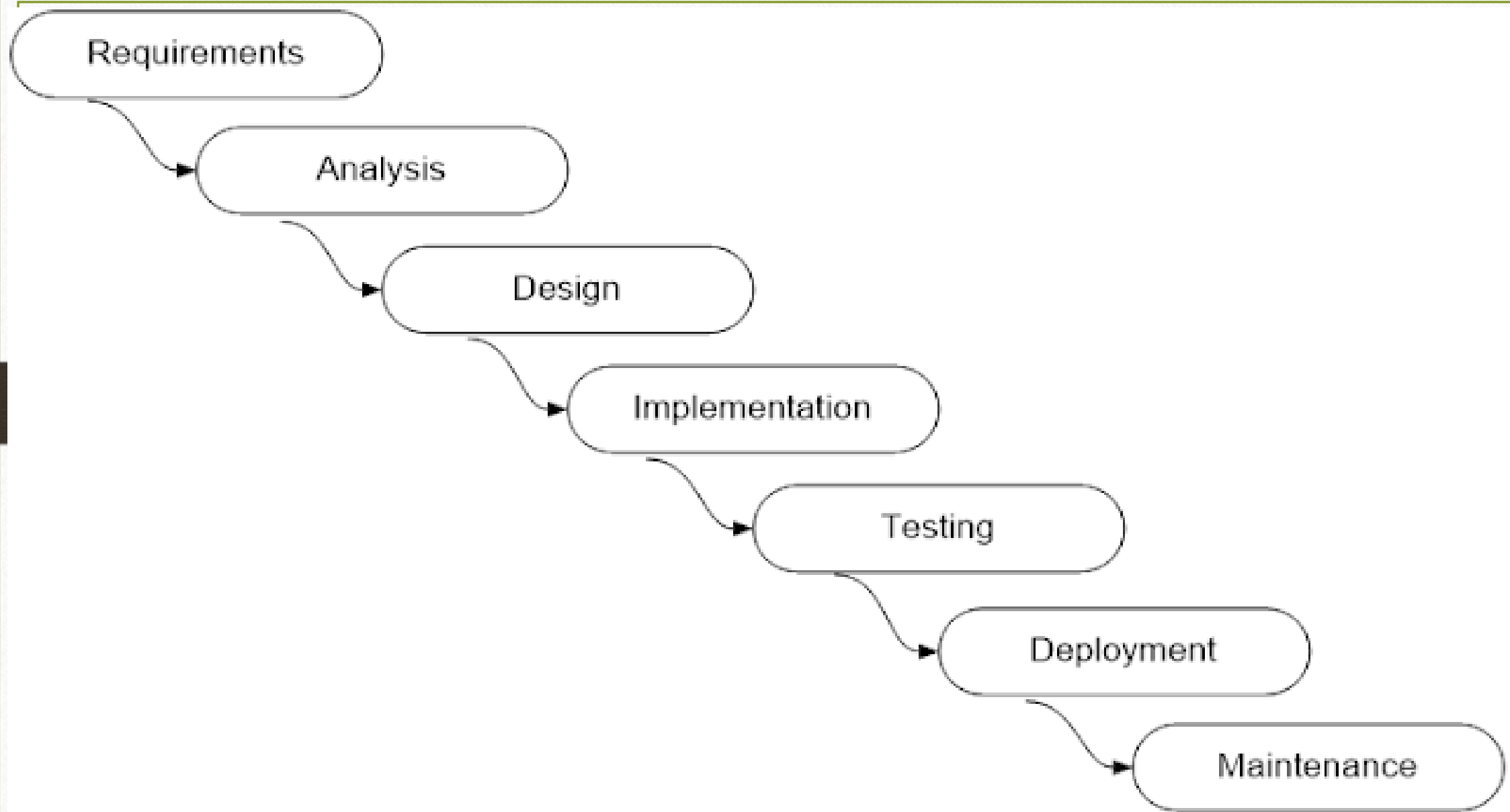
- The software should be correct. It should not contain errors.
- It should be easy to understand.
- It should be easy to maintain and update
- It should be reliable
- It should be flexible
- It should have an easier user interface.

Software Development Models

- **1. Waterfall Model**
- The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/ implementation, and maintenance. This is the simplest approach to develop a new system. It is an old fashion but reasonable approach which is adopted when the requirements are completely or mostly known. This approach consists of the following phases:

Waterfall model (continue)

- Requirements
- Analysis
- Design
- Implementation
- Testing
- Deployment
- Maintenance



Advantages of Waterfall model

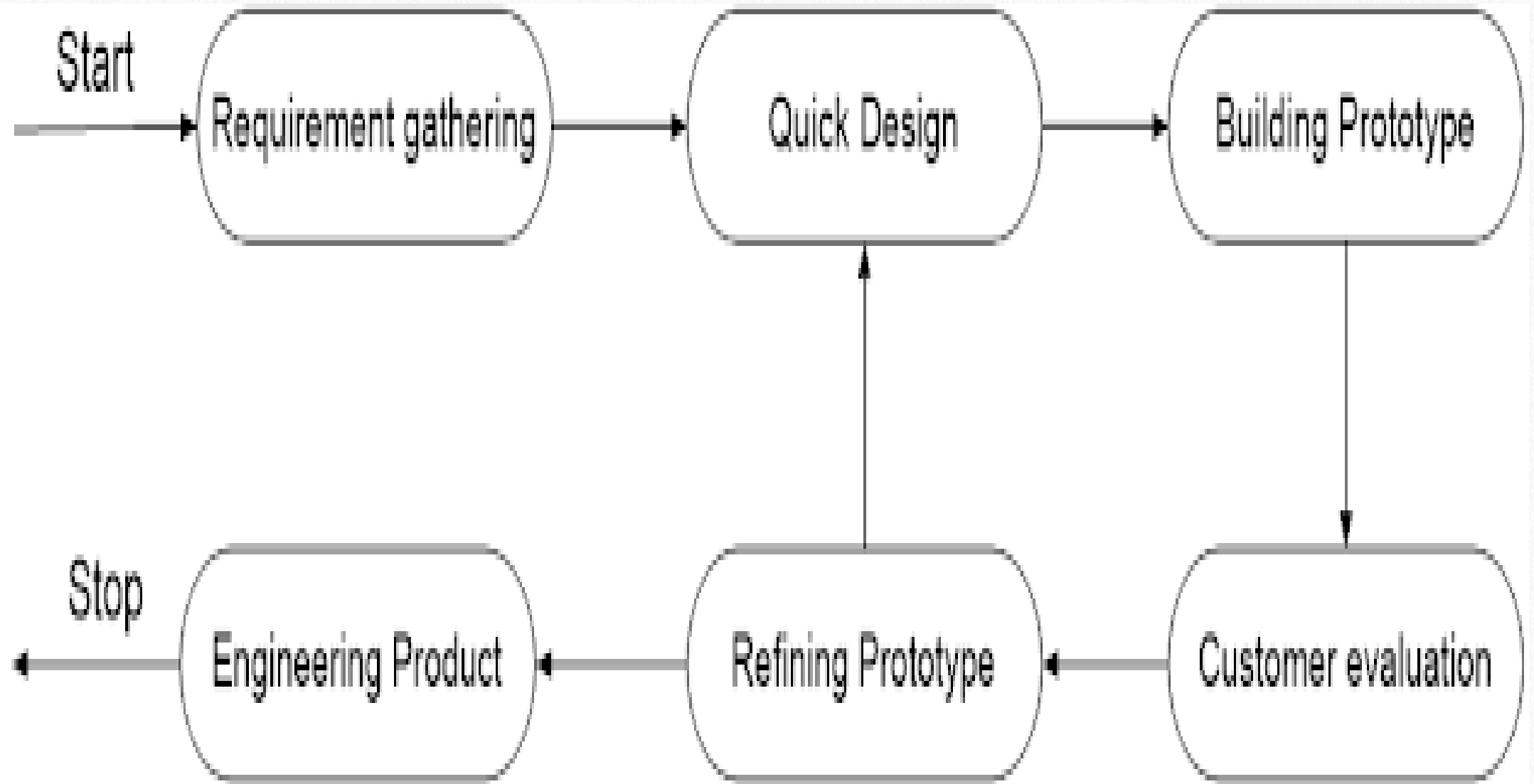
- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model, phases are processed and completed one at a time. Phases do not overlap.
- The waterfall model works well for smaller projects where requirements are very well understood.

Disadvantages of waterfall model

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought-out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Not suitable for large or bigger project

The Prototyping Model

- Prototyping is a working model which is based on the interaction between analyst and user. **Small prototype(model or sample) is designed at first.** It is developed on the basis of preliminary information gives the idea of how the full software will look likes and what it would perform? The user evaluates the prototype and refines the requirements to develop the full system. It is an attractive model to generate ideas for very large systems where there is no manual system. The prototype is helpful to obtain the details of the requirements from the user.



Advantages of Prototype model

- Users are actively involved in the development.
- Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
- Errors can be detected much earlier.
- Quicker user feedback is available leading to better solutions.

Disadvantages of Prototype model

- This leads to implementing and then repairing way of building systems.
- Practically, this methodology may increase the complexity of this system as the scope of the system may expand beyond original plans.
- An incomplete application may cause the application not to be used as the full system was designed Incomplete or inadequate problem analysis.

Spiral Model

- In System Development Life Cycle (SDLC), whenever you talk about risk and risk finding, solutions, etc. then the first thing that you should think of is the spiral model because the spiral model is a type of model in SDLC which basically focuses on risk analysis. In the spiral model, the basic meaning of the spiral is every loop that is iterating. Every rotation of 360deg is called a spiral. And every spiral consist of 4 phases. They are:
 - **1. Analysis 2.Evaluation 3.Development 4.Planning**

Four phases of spiral model

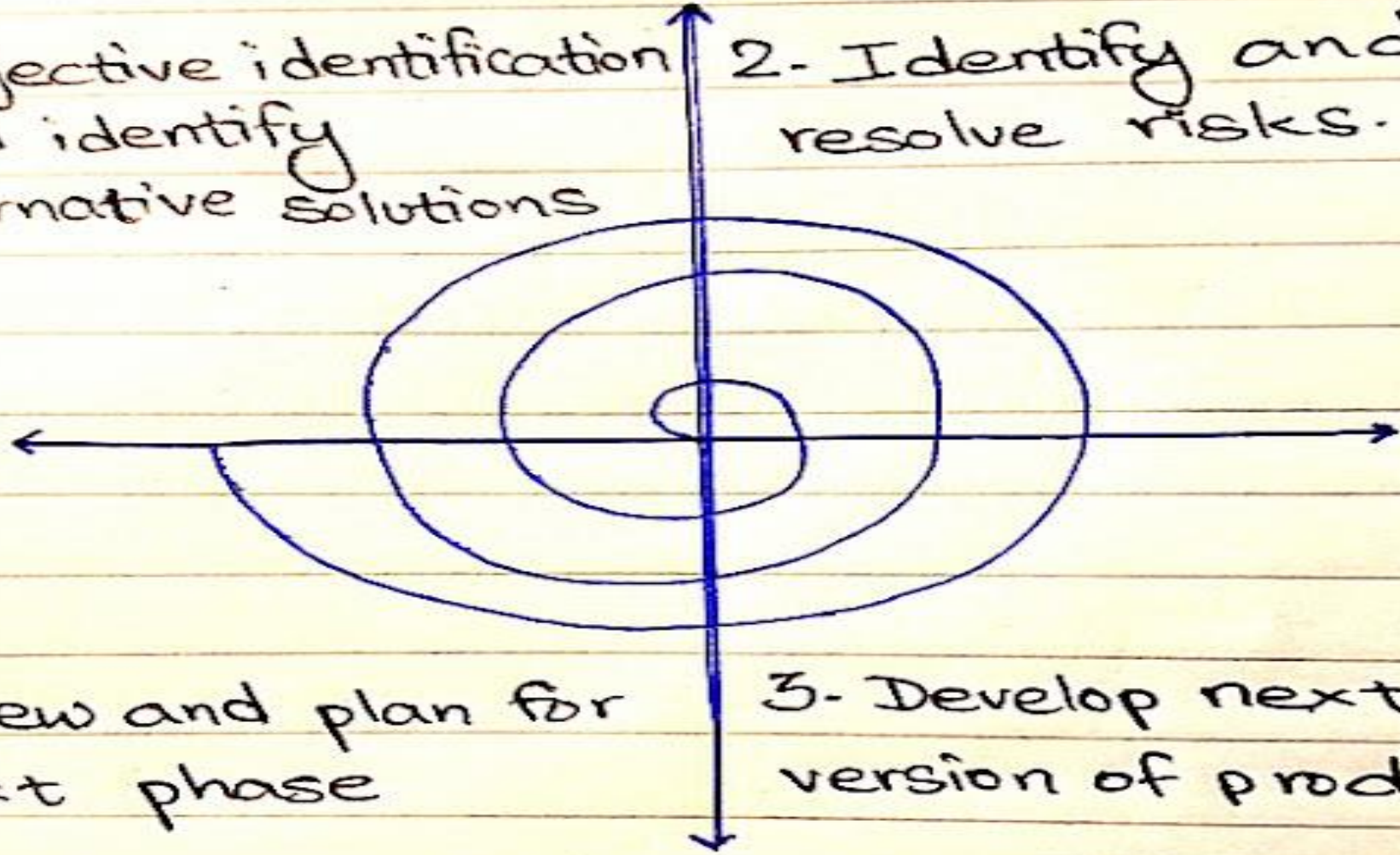
- Objective Identification and finding alternative solutions.
- Identification and resolving risks.
- Developing the next version of the project.
- Review and plan for the next phase.

Spiral model (continue)

- You can even see these phases in the diagram of the spiral model of SDLC. Here, the very first phase is objective identification and finding alternative solutions. In this phase first of all we identify the objective of the product and requirements of the customer and find out the possible solutions to that requirement. Then the second phase starts, it says identification and resolving risks which basically means that we have to identify what kinds of loopholes i.e. risks are prevailing in our system, and then we have to find out the possible solutions to these loopholes and risks i.e. resolving risks. Then comes the third phase of the spiral model in which we develop the next version of the project after resolving all the risk factors in our project. Here we solve all the risks and then apply the solutions to our project and develop the next version of it. But here, the best practice can be that we can develop a prototype and then follow these steps so that we can save time as well as money. Then at the last phase, we review our project and make changes as per the user requirement and then start planning for the next phase which means that we have to keep the loop running

1. Objective identification
and identify
alternative solutions

2. Identify and
resolve risks.



4. Review and plan for
next phase

3. Develop next
version of product

Fig. Spiral Model in SDLC

Advantages

- Risk handling
- We can create large projects.
- It is a flexible model
- Customer Satisfaction

DisAdvantages

- It is a complex model.
- It is a comparatively expensive model.
- Too much risk analysis
- Lack of time management

Agile Development Approach

- Let us take a real-life example, we will consider smartphones in this regard. We all know that smartphones are such a thing in the technology industry that is developing and evolving so quickly. Have you ever thought, how these smartphones are released so frequently and also perform well in the market? If not then the answer is that they follow the agile model. The agile model is such a powerful model that many tech giants are also following in the current time. So, we have the introduction now let's discuss how the agile model work.
- Also, this model is mainly used for large projects. So, basically, we, first of all, break our project into small chunks. There is a special term for this i.e. iteration. And those iterations run in parallel so that development can be faster. This is regarding development but this model focuses mainly on releasing the product as fast as possible in the market.

advantages

- Frequently Delivery of the product.
- Phase to Phase communication with the user regarding the product.
- Instant changes to the product as per requirement.
- Minimal time to release the product.

Disadvantages

- Less documentation and negotiations.
- Maintenance difficulty.

Documentation

- 1. System manual
- 2. Program Manual
- 3. User Manual

Thank you

...