Date
Page

Software life cycles models

Software life cycle models are structured approaches to software development, providing a framework for planning, managing and controlling the process of developing an information systems.

The most common software life cycle models are:

(1) Water fall model

It is linear and sequential approach where each phase must be completed before the next one begins

Phases:
Requirement Analysis
System Design:
Implementation.
Integration and Testing
Deployment.
Maintenance:

Advantages:

- Simple and easy to understand.
- Well suited for smaller project and clearly defined requirements

Disadvantages:

- Difficult to go back to any stages once it's completed.
- Inflexible to change during the development process

Scanned with CamScanner

V-Model (verification and validation model)

The v-model is an extension of the waterfall mod
Instead of moving down in a linear way, the process steps are bent upwards after the coding phase to form a v shape

Phase:

Requirement analysis ⟵ Acceptance Test Design ⟶ Acceptance Testing

System Design ⟵ System Test Design ⟶ System testing

Architecture Design ⟵ Integration Test Design ⟶ Integration Testing

Model Design ⟵ Unit Test Design ⟶ Unit Testing

Coding

Advantage

Emphasizes verification and validation activities.
Defects are detected in on early stages.

Disadvantages.

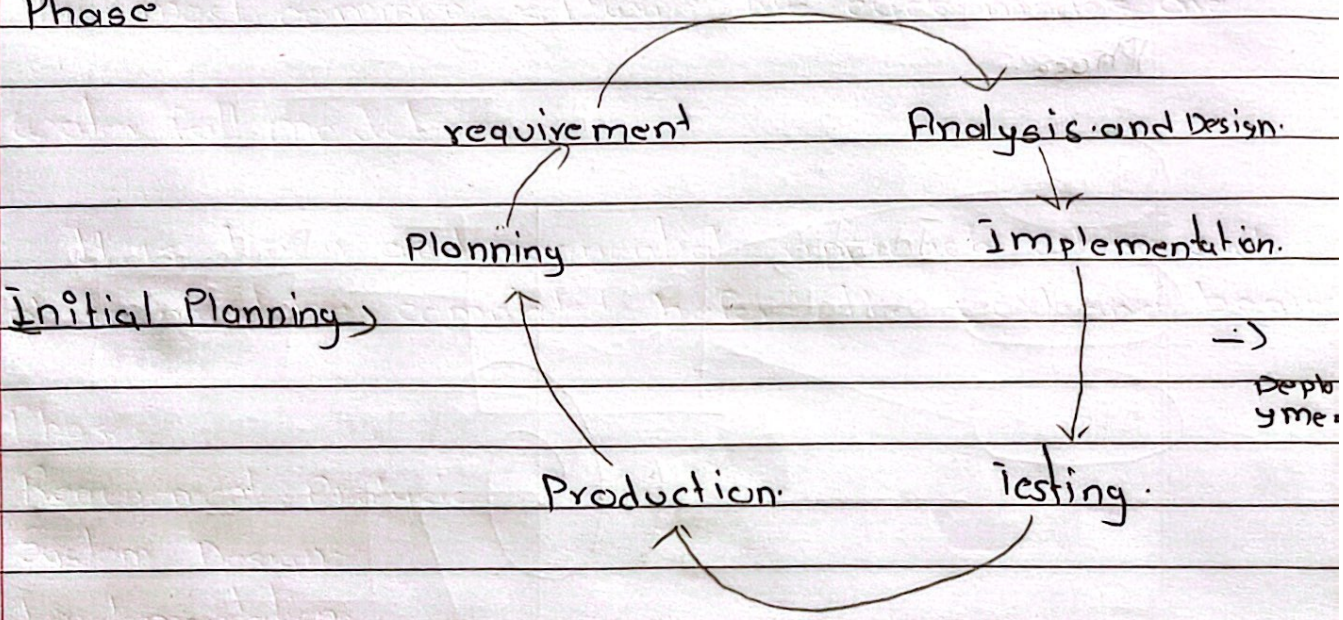
Bit is inflexible
little flexibility for changing requirements.

Scanned with CamScanner

Iterative and incremental model

This model focuses on the iterative refinement of system through multiple cycles and incremental development, where each increment adds functional capabilities.

Phase



Advantages

More flexible and adaptable to changing requirements
Progress in visible after each iteration.

Disadvantages

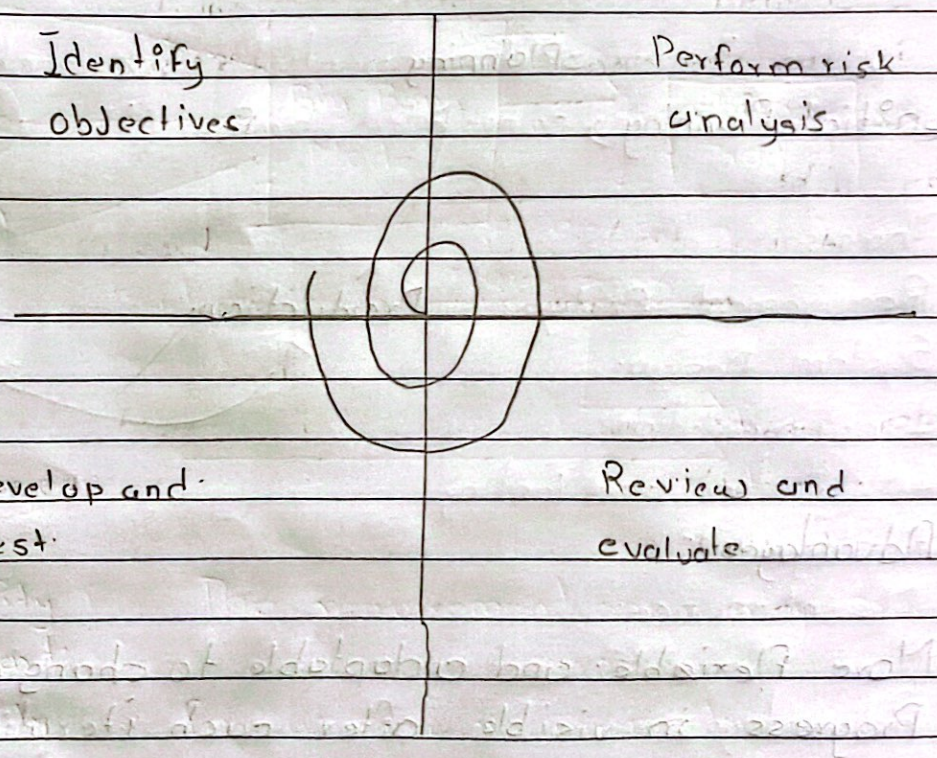
Requires carefull planning and design
Management complexity increases with more iterations

Scanned with CamScanner

4 Spiral model

The spiral model combines the idea of iterative development (prototyping) with the systematic aspect of the waterfall model. It focuses on risk assessment and reduction

Phase:



Advantages
Focuses on risk management
Suitable for large and complex projects

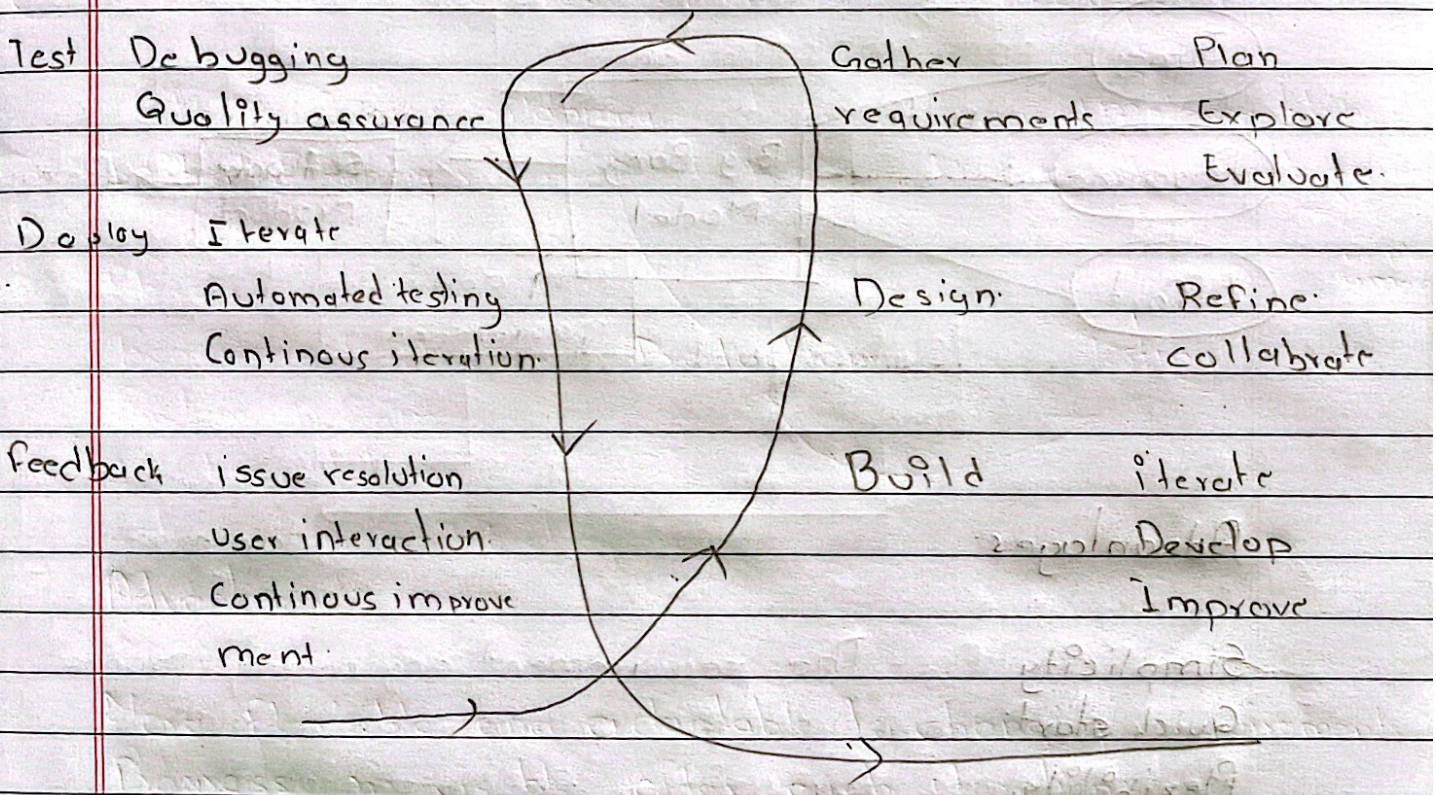
Disadvantage
Can be costly and time consuming
Requires expertise in risk assessment

Scanned with CamScanner

Agile Model **Agile Model**

Agile model is an iterative and incremental model that focuses on customer satisfaction through continuous delivery of functional software. It emphasizes flexibility, customer involvement and quick releases.

Phases.



Advantages.

- Highly flexible to changes.
- Continious customer feedback and involvement.

Disadvantages.

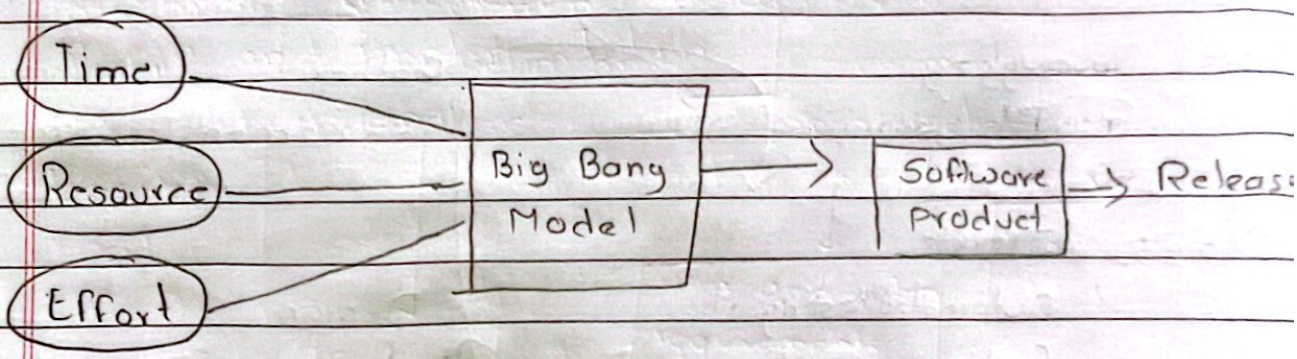
- Requires a high level of collaboration and communication
- Can be difficult to predict final outcome.

Scanned with CamScanner

Big Bang model

The Big Bang model involves with minimum planning and jumps directly into development This focus on using all available resource to build software product

Phases



Advantages:

Simplicity
Quick start.
Flexibility

Disadvantages:

High Risk
Quality issues.
Poor Scalability

Scanned with CamScanner

Requirement analysis and specification

Requirement analysis is significant and essential activity after elicitation we analyze, refine, and scrutinize (inspect closely) the gathered requirement to make consisent and unambiguous requirements

The various steps of requirement analysis are

Draw the Context Diagram

Develop prototypes optional

Model the requirements

Finalise the requirements

Techniques for Requirement Analysis

(I) Interviews
(II) Questionnaires
(III) Workshop
(IV) Observations
(V) Document Analysis
(VI) Prototyping

Scanned with CamScanner

Requirement specification is the process of documenting the requirements in a detailed and precise manner. This documentation servers as a reference for both developers and stakeholder throughout the project.

Requirement Engineering process

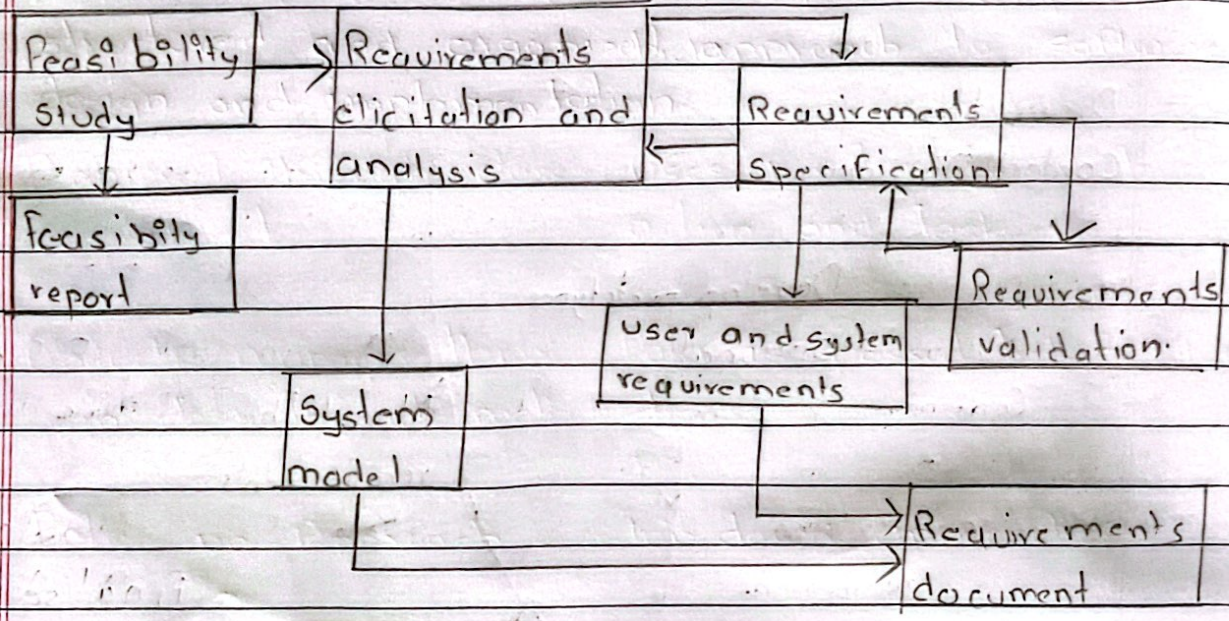


Fig:- The requirement engineering process

Best Practices for writing Requirements

- Clarity and Precision: Requirements should be clear, unambiguous and consise
- Consistency: Ensure that there are no conflict requirements
- Completeness Cover all aspects of the system and its functionalities

Scanned with CamScanner

Traceability: Each requirement should be traceable back to the source

Testability: Requirements should be defined in a way that allows them to be tested.

Object oriented software Development

Object-oriented software Development (OOSD) is a paradigm that uses the concept of "object" to design and build software systems

It is based on several key principles and methodolies that helps organize software design around data or objects, rather than function or logic.

Stages of ~~software~~ object-oriented software Developmer

1 Requirement Analysis
2 Object oriented Analysis
  ↳ identify and define the objects, and their attributes, methods and relationship based on the requirement.
  ↳ Use techniques like use case diagrams and class diagra
3 Object oriented Design
  ↳ Refine the analysis model and design the system architecture
  ↳ Use design pattern to solve common design problems
4 Implementation
5 Testing

Scanned with CamScanner

Benefits of objects-oriented software Developme

1. Modularity
2. Reusability
3. Scalability
4. Maintainability
5. Productivity

Object-oriented software development offers a structured and organized approach to software design and implementation.

Scanned with CamScanner