**PRACTICAL-1**

**AIM: Describe various software development models with appropriate diagram.**

Software development models, also known as Software Development Life Cycle (SDLC) models, provide a structured approach to developing software. Each model follows a series of phases to ensure the successful completion of a project.

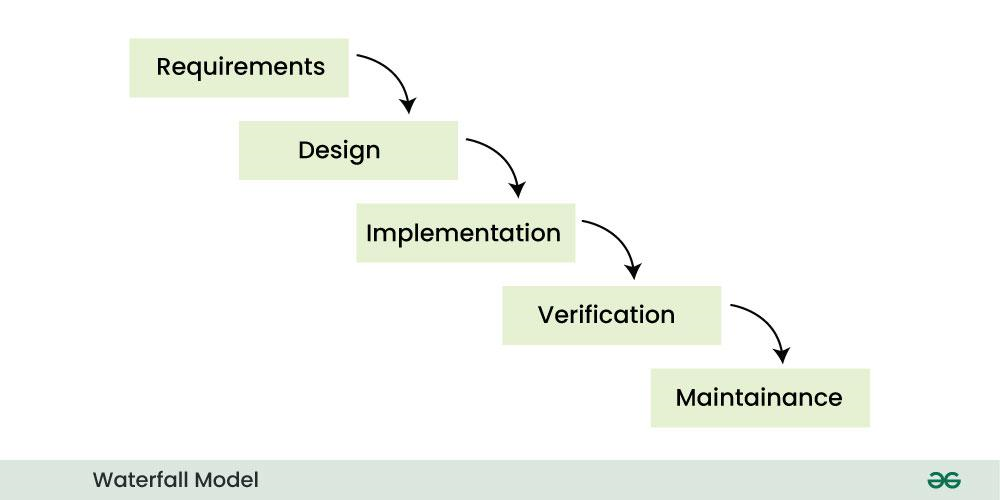
**1. Waterfall Model:**

**The Waterfall Model** is a linear and sequential approach where each phase must be completed before the next one begins. It is one of the most straightforward models.

**Phases:**

1. Requirements
2. Design
3. Implementation
4. Verification (Testing)
5. Maintenance

**WORKING:** Requirements → Design → Implementation → Testing → Maintenance



**Advantages:**

* Simple and easy to understand.
* Works well for small, well-defined projects.

**Disadvantages:**

* Inflexible; changes are difficult to implement once the process has started.
* Not suitable for large or complex projects.

**2. Agile Model:**

The Agile Model is an iterative and incremental approach that focuses on delivering small, functional pieces of software in short iterations (called sprints). It emphasizes customer collaboration and adaptability.

**Phases:**

* Planning
* Design
* Development
* Testing
* Review
* Deployment

**Diagram:** Sprint 1: Plan → Design → Develop → Test → Review → Deploy

Sprint 2: Plan → Design → Develop → Test → Review → Deploy



**Advantages:**

* Highly flexible and adaptable to changes.
* Continuous delivery of working software.

**Disadvantages:**

* Requires active customer involvement.
* May lack documentation compared to other models.

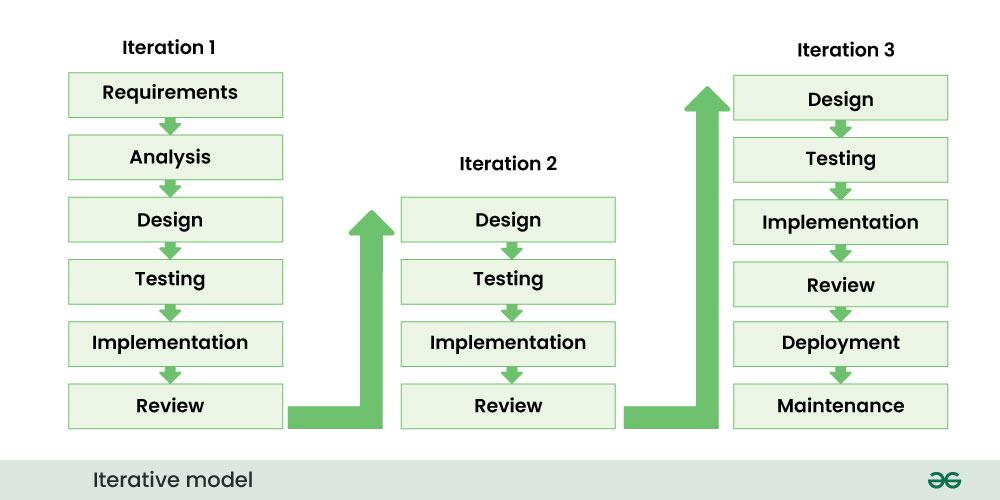
**3. Iterative Model:**

The Iterative Model develops software through repeated cycles (iterations) and incremental improvements. Each iteration produces a working version of the software, which is refined in subsequent iterations.

**Phases:**

* Initial Planning
* Design & Development
* Testing
* Evaluation
* Repeat

**Diagram:** Initial Planning → [Design & Develop → Test → Evaluate] → Repeat



**Advantages:**

* Allows for early delivery of partial solutions.
* Easier to incorporate changes.

**Disadvantages:**

* Requires careful planning and management.
* Can be resource-intensive.

**4. Spiral Model:**

The Spiral Model combines elements of both the Waterfall and Iterative models. It emphasizes risk analysis and is divided into four quadrants: Planning, Risk Analysis, Engineering, and Evaluation.

**Phases:**

1. Planning
2. Risk Analysis
3. Engineering (Design, Development, Testing)
4. Evaluation

**Diagram:** Start → Planning → Risk Analysis → Engineering → Evaluation → Repeat



**Advantages:**

* Focuses on risk management.
* Suitable for large, complex projects.

**Disadvantages:**

* Can be expensive and time-consuming.
* Requires expertise in risk analysis.

**5. V-Model (Verification and Validation Model):**

The V-Model is an extension of the Waterfall Model that emphasizes testing at each stage of development. Each development phase has a corresponding testing phase.

**Phases:**

* Requirements → Acceptance Testing
* Design → System Testing
* Implementation → Integration Testing
* Unit Testing

**Diagram:** Requirements → Design → Implementation

↓ ↓ ↓

Acceptance Testing → System Testing → Integration Testing → Unit Testing



**Advantages:**

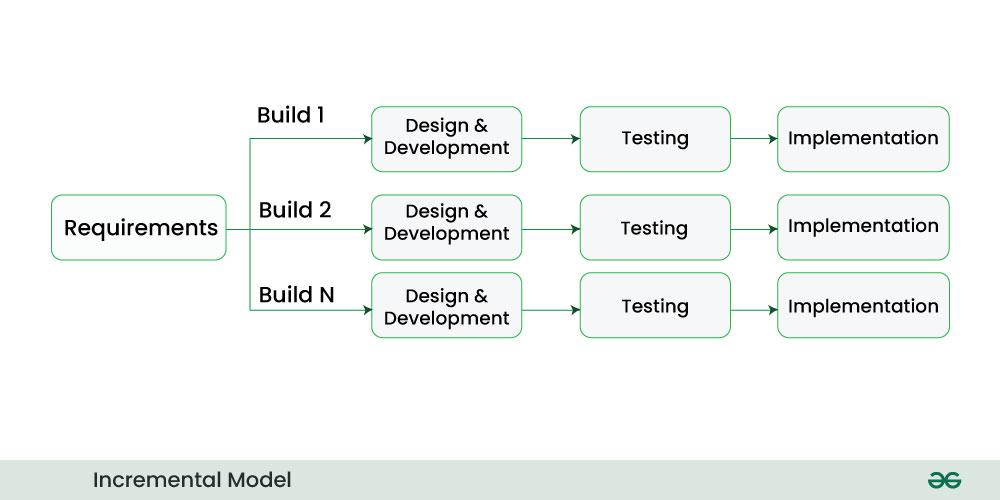
* Ensures high-quality software through rigorous testing.
* Clear and structured approach.

**Disadvantages:**

* Inflexible to changes.
* Not suitable for projects with evolving requirements.

**6. Incremental Model:**

In Incremental Model, the [software development process](https://www.geeksforgeeks.org/software-development-process/) is divided into several increments and the same phases are followed in each increment. In simple language, under this model a complex project is developed in many modules or builds.



**Phases of Incremental Model:**

* **Communication**: In the first phase, we talk face to face with the customer and collect his mandatory requirements. Like what functionalities does the customer want in his software, etc.
* **Planning**: In this phase the requirements are divided into multiple modules and planning is done on their basis.
* **Modeling**: In this phase the design of each module is prepared. After the design is ready, we take a particular module among many modules and save it in DDS (Design Document Specification). Diagrams like ERDs and DFDs are included in this document.
* **Construction**: Here we start construction based on the design of that particular module. That is, the design of the module is implemented in coding. **Deployment**: After the testing of the code is completed, if the module is working properly then it is given to the customer for use. After this, the next module is developed through the same phases and is combined with the previous module. This makes new functionality available to the customer.

**Advantages of Incremental Model:**

* This model is flexible and less expensive to change requirements and scope.
* Project progress can be measured.
* It is easier to test and debug during a short iteration.
* Errors are easy to identify.

**Disadvantages of Incremental Model:**

* Management is a continuous activity that must be handled.
* Before the project can be dismantled and built incrementally,
* The complete requirements of the software should be clear.
* This requires good planning and designing.

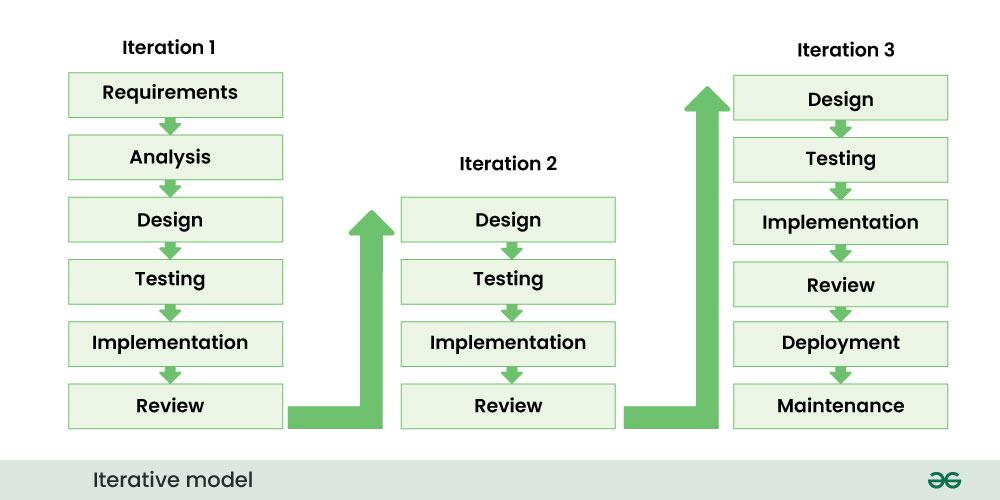
**7. Prototype Model:**

The Prototype Model involves creating an initial version of the software (prototype) to gather feedback from users. The prototype is refined iteratively until the final product is achieved.

**Phases:**

* Requirements Gathering
* Quick Design
* Prototype Development
* User Evaluation
* Refinement

**Diagram:** Requirements → Quick Design → Prototype → User Evaluation → Refinement



**Advantages:**

* Early user feedback improves the final product.
* Reduces the risk of misunderstandings.

**Disadvantages:**

* Can lead to scope creep.
* May require additional time and resources.

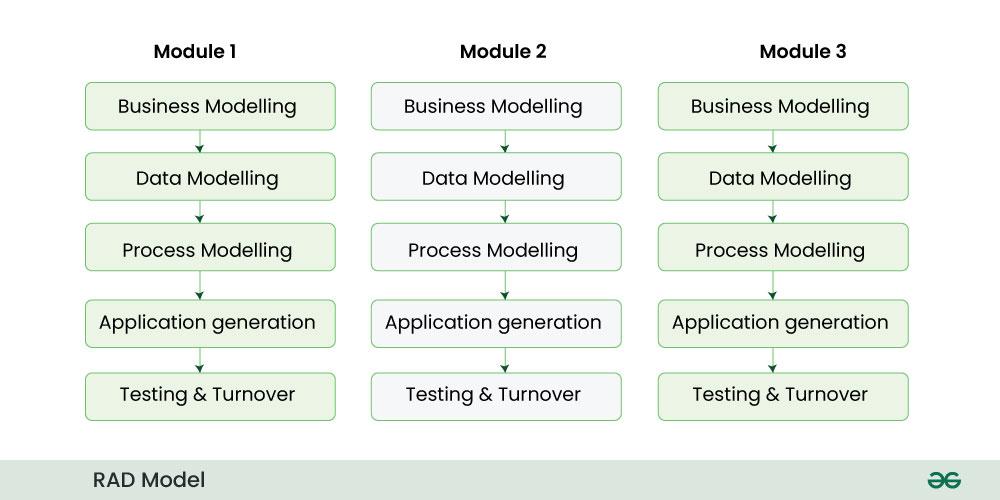
**8. RAD Model (Rapid Application Development):**

The RAD Model focuses on rapid prototyping and iterative development. It emphasizes user feedback and quick delivery of functional components.

**Phases:**

* Requirements Planning
* User Design
* Construction
* Cutover (Deployment)

**Diagram:** Requirements Planning → User Design → Construction → Cutover



**Advantages:**

* Faster delivery of software.
* High user involvement ensures better alignment with requirements.

**Disadvantages:**

* Requires highly skilled developers.
* Not suitable for small projects.