

Task 1 - SDLC Models

1 Waterfall Model

The Waterfall Model is a linear and sequential approach to software development. It consists of distinct phases that must be completed in order, resembling a waterfall flowing downwards. The simple layout is easy to understand, with requirements properly defined for well-understood milestones. However, this model lacks the flexibility to accommodate changes once the development cycle has begun due to the rigid nature of requirements upfront. The model includes the following stages:

Requirements Analysis: Gathering and documenting the requirements for the software system.

Design: Creating a high-level design and architecture based on the requirements.

Implementation: Writing code and developing the software based on the design.

Testing: Verifying and validating the software against the requirements and design.

Deployment: Installing and deploying the software in the production environment.

Maintenance: Providing ongoing support, updates, and software maintenance.

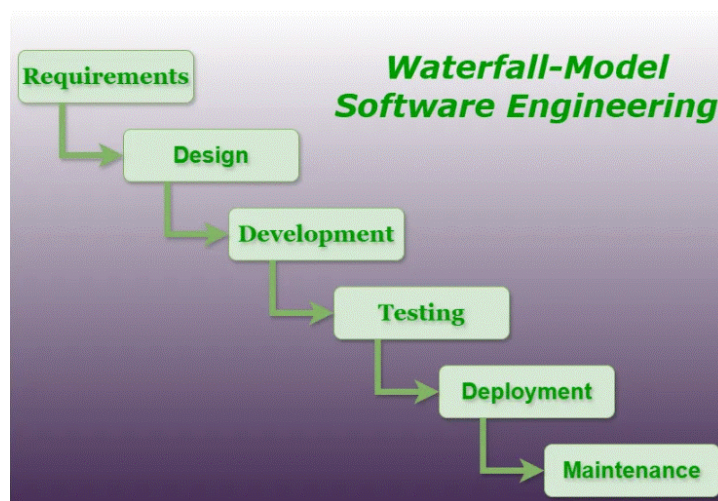


Figure 1: Waterfall Model (Source: GeeksforGeeks)

2 V-Shaped Model

The V-Shaped model is similar to the Waterfall Model in its sequential development cycle but emphasises testing at each stage. The model follows a V-shaped pattern, where the left side represents the development phases, and the right side represents the corresponding testing phases. This ensures early detection of defects and reduces the cost of fixing issues further in the development lifecycle.

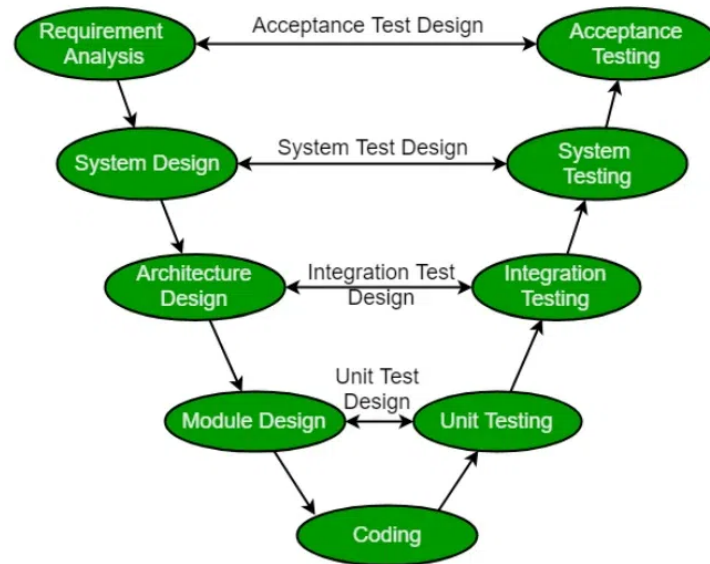


Figure 2: V-Shaped Model (Source: GeeksforGeeks)

3 Rapid Application Development Model

The Rapid Application Development (RAD) model is a short development cycle prioritising rapid prototyping and iterative development. RAD involves building small prototypes quickly to gather useful feedback and make necessary adjustments. This model is for projects with evolving requirements or tight deadlines, as it emphasises rapid delivery of software increments. The RAD model stages are:

Requirements Planning: Identifying and prioritising the key requirements for the software.

User Design: Involving end-users in the design process to rapidly prototype and iterate on user interfaces and functionality.

Construction: Rapidly building the software using iterative development cycles, focusing on delivering working prototypes quickly.

Cutover: Transitioning the software from development to production, including data migration and user training.

Maintenance: Providing ongoing support and maintenance for the software, incorporating feedback and making continuous improvements.

4 Spiral Model

The Spiral Model combines the iterative prototyping approach with the waterfall model's systematic development. It consists of multiple cycles, each of which includes four phases: planning, risk analysis, engineering, and evaluation. The model begins with a small set of requirements and iteratively expands upon them in subsequent cycles. The model emphasises risk management and accommodates changes throughout the development process. Spiral model includes the following stages:

Planning: Identifying objectives, constraints, and alternatives for the software project.

Risk Analysis: Evaluating potential risks and uncertainties associated with the project, including technical, schedule, and cost risks.

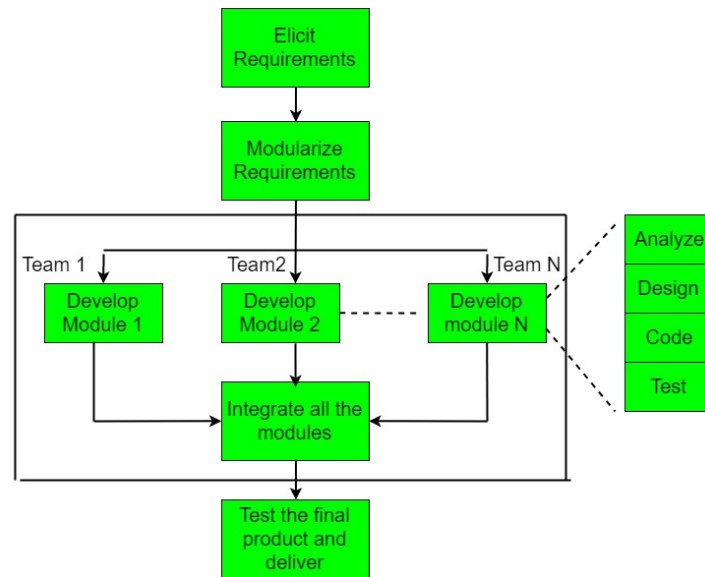


Figure 3: RAD Model (Source: GeeksforGeeks)

Engineering: Developing the software using iterative cycles, each of which includes requirements analysis, design, implementation, and testing.

Evaluation: Reviewing and assessing the project, including the identification of risks and potential changes to the project plan.

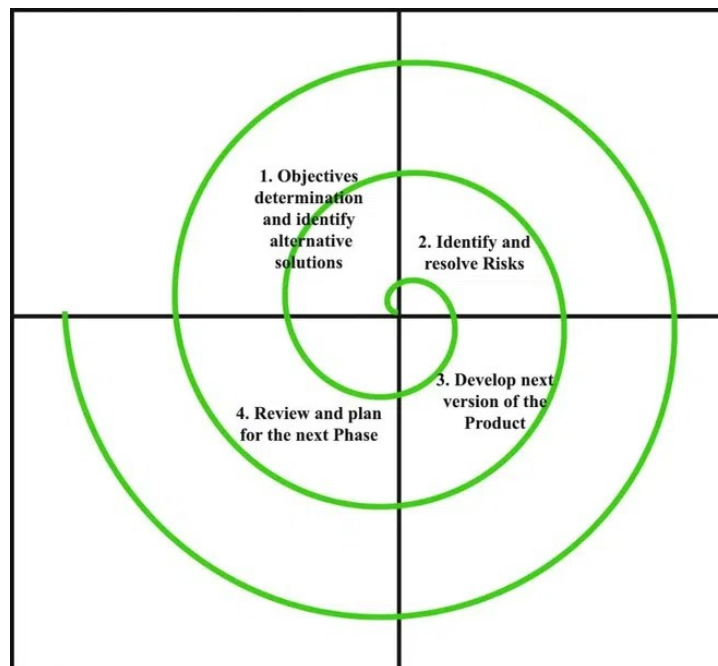


Figure 4: Spiral Model (Source: GeeksforGeeks)

5 Agile Model

The Agile Model is an iterative and incremental approach that emphasises adaptive planning, iterative delivery, and continuous improvement. Agile methods, such as Scrum and Kanban, prioritise collaboration, flexibility, and customer satisfaction. The development process is divided into short iterations called sprints, during which cross-functional teams deliver working software increments. Agile promotes rapid response to change and enables teams to deliver high-quality software efficiently.

Requirement Gathering: Identify requirements from the customer and plan time and work allocation to build the project. This allows for the evaluation of technical and economical feasibility.

Design the Requirements: Show new features and application to existing software through user-flow diagrams and UML diagrams. Wireframes and the design of the user interfaces are conducted at this stage.

Construction / Iteration: Start project work with aims to deploy a working product.

Testing / Quality Assurance: Testing such as unit testing, integration testing, and system testing is performed.

Deployment: Deploy the working project to end users.

Feedback: Gather feedback on product and correct bugs.

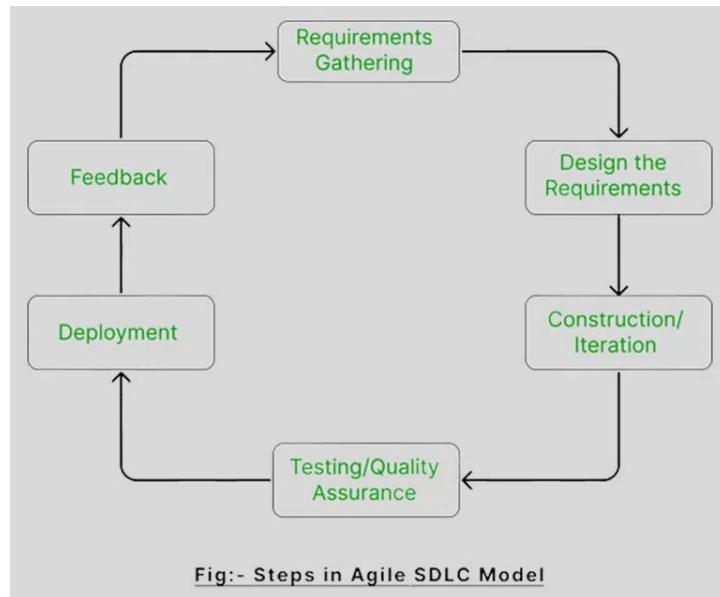


Figure 5: Steps in Agile Model (Source: GeeksforGeeks)