NAME:IKORO ERIC ONYEKACHI

MATRIC NUMBER:23/12137

COURSE CODE:SEN 201

1. Waterfall Model

- Explanation: The Waterfall Model is a linear and sequential approach where each phase must be completed before moving to the next. Phases include Requirements, Design, Implementation, Testing, Deployment, and Maintenance.

- Characteristics: Simple to understand and manage due to its sequential flow.

- Benefits: Best suited for small projects with well-defined requirements.

- Limitations: Rigid, with little flexibility for changes once a phase is completed, making it unsuitable for projects with evolving requirements.

2. V-Model (Verification and Validation Model)

- Explanation: An extension of the Waterfall Model where each development stage has a corresponding testing phase. Phases include Requirements Analysis, System Design, Implementation, and Testing, mapped in a “V” shape.

- Characteristics: Emphasizes verification and validation at every development stage.

- Benefits: High reliability due to early defect detection.

- Limitations: Similar to Waterfall, it is inflexible and less effective for complex projects with changing requirements.

3. Iterative Model

- Explanation: This model involves building software in small, iterative cycles, where each cycle produces a working version of the software that is gradually refined. Phases repeat for each iteration, including Planning, Design, Implementation, Testing, and Evaluation.

- Characteristics: Allows revisiting and improving software in successive iterations.

- Benefits: Reduces risk by providing early feedback and identifying issues early.

- Limitations: Requires good planning and management due to the potential complexity of multiple iterations.

4. Spiral Model

- Explanation: Combines elements of both iterative and Waterfall models, focusing on risk analysis. The phases include Planning, Risk Analysis, Engineering, and Evaluation, with each cycle adding functionality.

- Characteristics: Emphasizes risk management and allows multiple iterations.

- Benefits: Ideal for large, high-risk projects due to its iterative approach and risk focus.

- Limitations: Complex to manage and costly due to its focus on risk analysis and frequent planning.

5. Agile Model

- Explanation: Agile emphasizes flexibility, collaboration, and customer feedback, breaking development into small, iterative sprints. Phases include Planning, Design, Development, Testing, and Review within each sprint.

- Characteristics: Prioritizes adaptability to changing requirements.

- Benefits: Highly flexible and responsive to change, delivering functional parts of the software quickly.

- Limitations: Requires active customer involvement and may lack documentation due to the fast-paced nature.

6. Scrum

- Explanation: Scrum is an Agile framework focused on short development cycles or "sprints," typically lasting 2-4 weeks. Phases include Sprint Planning, Development, Daily Standups, Sprint Review, and Sprint Retrospective.

- Characteristics: Uses time-boxed sprints and emphasizes team roles like Scrum Master and Product Owner.

- Benefits: Improves productivity through short cycles and team focus.

- Limitations: Can lead to scope creep if sprint goals aren’t managed effectively.

7. Kanban

- Explanation: Kanban is a visual Agile methodology focused on continuous delivery without fixed sprints, using a Kanban board to track work stages. Phases are organized by task status (e.g., To Do, In Progress, Done).

- Characteristics: Visualizes workflow and limits work-in-progress (WIP) to improve focus.

- Benefits: Increases efficiency and reduces bottlenecks by focusing on task flow.

- Limitations: Not well-suited for projects that require strict planning and deadlines.

8. RAD (Rapid Application Development)

- Explanation: RAD emphasizes rapid prototyping and quick feedback over long planning cycles. Phases include Requirements Planning, User Design, Construction, and Cutover.

- Characteristics: Focuses on speed and client feedback to adapt the design.

- Benefits: Fast development cycle and high user involvement.

- Limitations: Needs highly skilled developers and is less suitable for large, complex systems.

9. Prototype Model

- Explanation: In this model, an initial prototype is built based on preliminary requirements, which is refined through user feedback. Phases include Requirements Gathering, Prototype Building, Testing, and Refinement.

- Characteristics: Reduces misunderstandings by focusing on early user interaction.

- Benefits: Helps clarify requirements and improve user satisfaction.

- Limitations: Can be time-consuming and costly if too many iterations are required.

10. DevOps Model

- Explanation: DevOps integrates development and operations teams to enhance collaboration, automate deployment, and maintain continuous delivery. Key phases include Continuous Development, Continuous Testing, Continuous Integration, and Continuous Monitoring.

- Characteristics: Encourages collaboration, automation, and continuous feedback.

- Benefits: Speeds up development and release cycles, increasing quality and reliability.

- Limitations: Complex to implement, requiring cultural changes and specific tools for automation.