**Software Development Notes**

**1. Waterfall Model**

The Waterfall Model is a traditional software development methodology that follows a linear and sequential design process. Each phase must be completed before the next one begins, making it suitable for projects with well-defined requirements.

• Phases include Requirements, Design, Implementation, Testing, Deployment, and Maintenance.

• Simple and easy to manage due to its rigidity.

• Not suitable for projects with changing requirements.

**2. Agile Methodology**

Agile is an iterative and incremental approach to software development. It emphasizes flexibility, collaboration, and customer satisfaction. Projects are divided into small cycles called sprints that deliver working software frequently.

• Promotes adaptability and continuous improvement.

• Customer feedback is integrated at each stage.

• Encourages teamwork and transparency.

**3. Principles of Agile**

Agile is based on 11 key principles that guide development and teamwork to achieve high-quality and adaptive results.

• Customer satisfaction through continuous delivery.

• Welcoming changes at any stage of development.

• Deliver working software frequently.

• Business and developers must collaborate closely.

• Projects are built around motivated individuals.

• Face-to-face communication is most effective.

• Working software is the main measure of progress.

• Sustainable development pace must be maintained.

• Focus on technical excellence and design.

• Simplicity is essential for efficiency.

• Teams should self-organize and reflect for improvement.

**4. Scrum Master and His Tasks**

The Scrum Master acts as a coach and facilitator for the Scrum team. He ensures the team understands and follows Agile practices, removes obstacles, and supports collaboration.

• Facilitates Scrum ceremonies such as Sprint Planning and Daily Standups.

• Helps the team stay focused on goals.

• Removes impediments that slow progress.

• Promotes self-organization and Agile principles.

**5. Product Owner**

The Product Owner represents the stakeholders and customers. They are responsible for defining the product vision and ensuring the team delivers value.

• Manages and prioritizes the product backlog.

• Acts as a bridge between stakeholders and team.

• Defines acceptance criteria for features.

• Decides on product release plans.

**6. Cross Functional Teams**

Cross-functional teams include members with different skills necessary to deliver a complete product increment. They reduce dependency on external teams and promote collaboration.

• Includes developers, testers, designers, and business analysts.

• Promotes faster delivery and better problem-solving.

• Encourages team ownership of the product.

**7. Advantages of Agile Methodology**

Agile methodology offers several advantages that make it popular in modern software development.

• Faster delivery of features.

• Flexibility to adapt to changes.

• Improved quality through continuous testing.

• Greater customer satisfaction.

**8. Disadvantages of Agile Methodology**

Despite its benefits, Agile also has some challenges.

• Requires experienced and dedicated teams.

• Less focus on documentation.

• Hard to predict project timelines and costs.

• May not work well for very large teams.

**9. CI/CD (Continuous Integration and Continuous Deployment)**

CI/CD automates the software delivery process. Continuous Integration ensures code changes are merged and tested frequently, while Continuous Deployment delivers these changes automatically to production environments.

• Reduces manual errors in testing and deployment.

• Provides faster feedback to developers.

• Ensures reliable and repeatable releases.

**10. Scrum and Scrum Framework**

Scrum is an Agile framework used to manage complex projects. It focuses on delivering small increments of work within short iterations called sprints.

• Roles: Product Owner, Scrum Master, Development Team.

• Ceremonies: Sprint Planning, Daily Scrum, Sprint Review, Retrospective.

• Artifacts: Product Backlog, Sprint Backlog, Increment.

**11. Application of Scrum in Project Management**

Scrum can be applied to project management in the following ways:

• Sprint Planning Meeting: Defines sprint goals and tasks.

• Tracking Progress: Monitored using burn-down charts and daily standups.

• Sprint Review: Showcases completed work and gathers feedback.

**12. Advantages and Disadvantages of Scrum**

Scrum offers several strengths but also some limitations.

• Advantages: Transparency, adaptability, faster delivery.

• Disadvantages: Needs skilled teams, difficult to scale for large projects.

**13. Docker**

Docker is a containerization platform that packages applications with their dependencies in lightweight containers. It provides consistency across environments and simplifies deployment.

• Ensures applications run the same everywhere.

• Improves scalability and portability.

• Reduces infrastructure overhead.

**14. DevOps Tools**

DevOps uses various tools to enable automation, version control, and testing.

• Jenkins: Automation server for CI/CD pipelines.

• Git and GitHub: Version control and collaboration platform.

• Selenium: Tool for automated functional testing.

**15. Scrum Ceremonies and Artifacts**

Scrum ceremonies are events for managing work, while artifacts are documents or tools that track progress.

• Ceremonies: Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective.

• Artifacts: Product Backlog, Sprint Backlog, Increment.

**16. Kanban and Its Advantages**

Kanban is a workflow management method that visualizes tasks on a board. It helps teams manage work in progress and improve efficiency.

• Visual representation of tasks using boards and cards.

• Helps identify bottlenecks in the process.

• Promotes continuous delivery and efficiency.

**17. DevOps Lifecycle**

The DevOps lifecycle integrates development and operations into one continuous process to ensure rapid and reliable delivery.

• Plan → Code → Build → Test → Release → Deploy → Operate → Monitor

**18. Spring Boot Project with Git and Jenkins**

Students should practice creating a Spring Boot project, pushing the code to GitHub, and using Jenkins for CI/CD automation.

• Develop a Spring Boot project.

• Push code to GitHub repository.

• Integrate Jenkins for automated builds and deployments.

**19. Submission Requirements**

For completion of this module, students must submit the following:

• Project files.

• PowerPoint Presentation (PPT).

• Word file documentation for each day.

• Repository link.