**Software Development Methodologies**

**1️Waterfall Model**

**A linear, phase-by-phase approach best suited for projects with fixed requirements.**

* **Phases: Requirements → Design → Implementation → Testing → Deployment → Maintenance**
* **Easy to manage due to its structured flow**
* **Not ideal for projects with evolving needs**

**2️Agile Methodology**

**An iterative and flexible approach focused on collaboration and customer feedback.**

* **Work is delivered in short cycles called sprints**
* **Encourages adaptability and continuous improvement**
* **Customer involvement is central throughout development**

**Agile Principles (11 Key Guidelines)**

**Agile is built on values that promote quality, adaptability, and teamwork:**

* **Continuous delivery for customer satisfaction**
* **Embrace change at any stage**
* **Deliver working software frequently**
* **Close collaboration between business and developers**
* **Empower motivated individuals**
* **Prioritize face-to-face communication**
* **Measure progress through working software**
* **Maintain sustainable pace**
* **Focus on technical excellence**
* **Keep things simple**
* **Encourage self-organizing teams and reflection**

**Scrum Roles & Responsibilities**

**Scrum Master**

**Facilitator and coach for the team.**

* **Leads Scrum ceremonies (Sprint Planning, Daily Standups)**
* **Removes obstacles to progress**
* **Promotes Agile values and self-organization**

**Product Owner**

**Voice of the customer and stakeholder.**

* **Manages the product backlog**
* **Defines feature priorities and acceptance criteria**
* **Plans releases and ensures value delivery**

**Cross-Functional Teams**

**Teams with diverse skills to deliver complete product increments.**

* **Includes developers, testers, designers, analysts**
* **Reduces external dependencies**
* **Enhances collaboration and ownership**

**Agile: Pros & Cons**

**Advantages**

* **Rapid feature delivery**
* **High adaptability**
* **Continuous testing improves quality**
* **Strong customer engagement**

**Disadvantages**

* **Requires skilled, committed teams**
* **Less emphasis on documentation**
* **Harder to estimate timelines and budgets**
* **May not scale well for large teams**

**CI/CD (Continuous Integration & Deployment)**

**Automates the development pipeline for faster, safer releases.**

* **Frequent code integration and testing**
* **Automatic deployment to production**
* **Reduces manual errors and accelerates feedback**

**Scrum Framework**

* **Roles: Product Owner, Scrum Master, Development Team**
* **Ceremonies: Sprint Planning, Daily Scrum, Sprint Review, Retrospective**
* **Artifacts: Product Backlog, Sprint Backlog, Increment**

**Application in Project Management**

* **Sprint Planning defines goals and tasks**
* **Progress tracked via burn-down charts and standups**
* **Sprint Review gathers feedback and showcases work**

**Scrum: Pros & Cons**

* **Advantages: Transparency, adaptability, faster delivery**
* **Disadvantages: Requires skilled teams, scaling challenges**

**DevOps & Tools**

**Docker**

**Containerization platform for consistent and portable deployments.**

* **Packages apps with dependencies**
* **Ensures uniform behavior across environments**
* **Reduces infrastructure complexity**

**DevOps Tools**

* **Jenkins: CI/CD automation**
* **Git & GitHub: Version control and collaboration**
* **Selenium: Automated testing**

**DevOps Lifecycle**

**A continuous loop integrating development and operations:**

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

**Kanban Methodology**

**A visual workflow system that enhances efficiency and transparency.**

* **Uses boards and cards to track tasks**
* **Identifies bottlenecks**
* **Supports continuous delivery**