Agile Software Development – Software Engineering

- Agile Software Development is an **iterative, incremental, and adaptive approach** to building software that focuses on **early delivery of value, continuous customer involvement, and rapid response to change**. Unlike traditional plan-driven models, Agile accepts that requirements evolve over time and treats change as an opportunity rather than a problem.
- Agile is used because it enables teams to **deliver working software early and continuously**, allowing customers to see real progress, provide feedback, and influence development decisions. By focusing on **value-driven features**, Agile minimizes unnecessary work and reduces the risk of building unwanted functionality.

- Beyond a process, Agile represents a **mindset and cultural shift** that prioritizes **people, collaboration, transparency, and shared responsibility**. Teams are empowered to self-organize, communicate openly, and make decisions that improve product quality and team morale.
- The **Agile Software Development Process** is executed through short, time-boxed iterations (sprints). Each iteration includes:

1. **Requirements Gathering**
2. **Planning**
3. **Development**
4. **Testing**
5. **Deployment**
6. **Maintenance**

- Agile development is guided by the **four core values of the Agile Manifesto**:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

- These values are reinforced by **12 principles** that emphasize early delivery, welcoming change, frequent releases, close collaboration, sustainable development pace, technical excellence, simplicity, and continuous improvement.

- Common **Agile practices** such as **Scrum, Kanban, Continuous Integration, Test-Driven Development, and Pair Programming** support frequent feedback, high code quality, and fast adaptation.
- Agile offers major benefits including **faster time-to-market, improved customer satisfaction, higher software quality, reduced risk, and increased team motivation**.

However, it also presents challenges such as **limited predictability, scope creep, dependency on customer availability, reduced documentation, and difficulties in large-scale projects**.

- In conclusion, Agile is **not a single method**, but a **framework of values and practices** that creates an environment where effective software solutions can continuously emerge in fast-changing conditions.

Advantages and Disadvantages of Agile Software Development.

- **Advantages:**
- Early and continuous delivery
- High customer satisfaction
- Flexibility to change
- Improved quality through continuous testing
- Strong team collaboration
- **Disadvantages:**
- Uncertain schedules and costs
- Risk of scope creep
- Dependence on customer involvement
- Less documentation
- Difficulties in large projects

Important Review

- **Agile mainly** emphasizes delivering working software and the ability to adapt to changing requirements.
- **Agile development delivers software** frequently in small, incremental releases.
- **Agile reduces project risk by** delivering software early, detecting issues quickly, and incorporating continuous customer feedback.
- **Agile can be difficult to apply in large organizations** because coordination and face-to-face communication become challenging, and effective decision-making often requires experienced teams.