AGILE&SCRUM

Introduction

- In the world of software development, traditional models such as the **Waterfall approach** were often too rigid and slow.
- To adapt to changing customer needs, the industry shifted towards **Agile methodology**, which focuses on flexibility, collaboration, and iterative delivery.
- One of the most widely used Agile frameworks is **Scrum**, which provides a structured way to manage projects in smaller, manageable iterations called **sprints**.

What is Agile?

- **Agile** is a **software development methodology** based on **iterative development**, where requirements and solutions evolve through collaboration between cross-functional teams.
- It emphasizes **adaptability**, **customer collaboration**, and **continuous improvement**.

Key Principles of Agile

1. Customer satisfaction through **early and continuous delivery** of valuable software.
2. Welcoming **changing requirements**, even late in development.
3. Delivering working software **frequently** (weeks rather than months).
4. Close collaboration between **business stakeholders and developers**.
5. Building projects around **motivated individuals** and trusting them to get the job done.
6. Focusing on **face-to-face communication** for effectiveness.
7. Measuring progress through **working software**, not lengthy documentation.
8. Promoting **sustainable development** where pace can be maintained indefinitely.
9. Continuous attention to **technical excellence** and good design.

10. Simplicity — maximizing the amount of work not done.
11. Teams should be **self-organizing**.
12. Regularly reflecting on how to become more effective, then adjusting accordingly.

What is Scrum?

- **Scrum** is one of the most popular frameworks used to implement Agile principles.
- It is a **lightweight, simple-to-understand, but difficult-to-master framework**.
- Scrum divides the work into small, time-boxed iterations called **sprints** (usually 2–4 weeks long).

Key Characteristics of Scrum:

- **Iterative development**: Work is delivered in cycles (sprints).
- **Incremental delivery**: Each sprint delivers a usable product increment.
- **Defined roles**: Scrum defines specific roles for accountability (Product Owner, Scrum Master, Development Team).
- **Events**: Scrum follows structured events like Sprint Planning, Daily Scrum (stand-up), Sprint Review, and Sprint Retrospective.
- **Artifacts**: Scrum uses artifacts such as the Product Backlog, Sprint Backlog, and Increment.

Key Roles in Scrum

a) Product Owner

- Represents the **voice of the customer**.
- Responsible for **defining the product vision** and ensuring that the team delivers value to the business.
- Maintains and prioritizes the **Product Backlog** (list of tasks and features).
- Decides what features should be developed first, based on business value and customer needs.
- Works closely with stakeholders and the Scrum Team.

b) Scrum Master

- Acts as a **servant leader** and coach for the Scrum Team.
- Ensures the team understands and follows **Scrum practices**.
- Helps remove **impediments** that block the team's progress.
- Facilitates meetings like Sprint Planning, Daily Scrum, Sprint Review, and Retrospective.
- Protects the development team from unnecessary interruptions.
- Promotes a culture of **continuous improvement**.

Benefits of Agile and Scrum

- **Flexibility**: Adapts to changing requirements quickly.
- **Faster Delivery**: Working software is delivered frequently.
- **Higher Quality**: Continuous testing and feedback improve product quality.
- **Transparency**: Progress is visible to stakeholders at all times.
- **Better Collaboration**: Strong communication between teams and stakeholders.
- **Customer Satisfaction**: Regular delivery of value ensures happy customers.
- **Continuous Improvement**: Teams reflect and improve after every sprint.