**What is Agile?**

**Agile is an iterative and incremental approach to software development, compared to traditional methods like Waterfall.**

* **Waterfall Model: This is a linear process where all requirements are gathered upfront, and development follows a strict step-by-step process.**
  + ***Downside*: Late changes are costly and time-consuming.**
* **Agile: Agile delivers work in small, manageable increments (called sprints). These sprints allow continuous feedback and adjustment.**

**Key features:**

* **Continuous delivery of small parts of the project.**
* **Requirements are evaluated continuously, allowing changes.**
* **More flexible QA processes.**
* **Frequent communication between teams and management.**
* **High product quality due to regular testing and feedback.**

**Why Agile?**

* **Short development cycles (sprints) allow faster product delivery.**
* **Products are released in iterations, making them more adaptive to market changes.**
* **Time is spent on what the market or users demand, resulting in effective results.**

**Principles of Agile Manifesto**

* **Customer satisfaction through rapid, continuous delivery of software.**
* **Frequent delivery of software (weeks rather than months).**
* **Close collaboration between business teams and developers.**
* **Self-organizing teams that work with little external control.**
* **Continuous attention to technical excellence and good design.**

**Benefits of Agile**

1. **Higher product quality: Continuous testing and feedback improve quality.**
2. **Reduced risk: Frequent deliveries mean any issues are spotted early.**
3. **Better visibility: Everyone involved knows the project's current status.**
4. **Increased project control: The project team adapts to changes quickly.**
5. **Better predictability: Regular sprints and feedback lead to a clear path forward.**

**Drawbacks of Agile**

1. **Lack of documentation: Agile prioritizes working code over detailed documentation.**
2. **Scope creep: Since Agile is flexible, there’s a risk of adding more features than initially planned.**
3. **High time demand: Daily meetings and feedback require constant team involvement.**
4. **Decision-making in meetings: Teams must be able to make critical decisions quickly.**
5. **Cost: Rapid changes in scope can increase development costs.**

**Agile Methodology: Scrum**

**Scrum** is one of the most popular frameworks under Agile. It helps teams work together to develop software **piece by piece**.

* Each piece is delivered in a **sprint** (usually 2-4 weeks long).
* The last sprint results in the entire software.

**Roles in Scrum:**

1. **Scrum Master (SM)**:
   * Leads the team, ensuring they follow Agile principles.
   * Removes obstacles and improves team efficiency.
2. **Product Owner (PO)**:
   * Represents stakeholders and customers.
   * Defines and prioritizes features in the **Product Backlog**.
   * Works with the team to ensure value delivery.
3. **Scrum Team**:
   * Consists of developers and testers.
   * Team size: Typically 7 developers and 3 testers (ratio varies).
   * Responsible for coding, testing, and delivering increments.

**Scrum Workflow**

Here’s a visual breakdown of how Scrum works:

**1. Kickoff Meeting**

* **Objective**: Identify the product owner and align key stakeholders (SHs).
* No time box.

**2. Product Backlog Grooming (PBL)**

* **Objective**: Gather all **user stories** (features and requirements).
* Participants: PO, stakeholders, Scrum Master (SM).
* No time box.

**3. Sprint Planning Meeting**

* **Objective**: Choose specific **user stories** from the product backlog to include in the next sprint.
* Participants: PO, SM, Scrum Team, SH.
* Time box: 3-4 hours.

**4. Sprint Development**

* Daily Scrum meetings are held to discuss progress, challenges, and plans for the next 24 hours.
* Time box: 15-30 minutes.

**5. Sprint Review Meeting**

* **Objective**: Present completed work to stakeholders and gather feedback.
* Time box: 2-3 hours.

**6. Sprint Retrospective**

* **Objective**: Reflect on the sprint, discussing what went well and what can be improved.
* Time box: 2-3 hours.

**7. Sprint Release**

* After the review, the sprint is released, and the team continues maintaining and supporting the product.

**Diagram: Scrum Process Overview**

Product Owner (PO) Scrum Master (SM)

/ | |

Product Backlog Sprint Planning

(User Stories) Meeting

| |

Sprint Backlog -> Daily Scrum -> Sprint Development

(Sprint Goals) (Short Meetings) (Code + Test)

|

Sprint Review Meeting -> Sprint Release -> Sprint Retrospective

**Scrum Events and Artifacts**

1. **Sprint**: The basic unit of work, usually 2-4 weeks long.
2. **Sprint Planning**: Defines the scope and goals of the sprint.
3. **Daily Scrum**: Short, daily meetings to assess progress.
4. **Sprint Review**: Present completed work to stakeholders and get feedback.
5. **Sprint Retrospective**: Reflect on what went well and what could be improved.

**Artifacts:**

* **Product Backlog**: A list of all desired features or changes.
* **Sprint Backlog**: Subset of the product backlog selected for the sprint.
* **Increment**: The sum of all completed features delivered in a sprint.

**Roles and Responsibilities**

* **Client**: Helps in Product Owner selection and gives feedback during reviews.
* **Stakeholders (SH)**: Involved in meetings, providing user stories and feedback.
* **Product Owner (PO)**: Defines user stories, prioritizes them, and manages the backlog.
* **Scrum Master (SM)**: Facilitates the process, removes obstacles, and ensures the team adheres to Scrum practices.
* **Scrum Team**: Develops and tests features within the sprint.

**Scrum Ceremonies (Meetings)**

1. **Kickoff Meeting**: No time box.
2. **Product Backlog Grooming**: Continuous refinement of the backlog.
3. **Sprint Planning**: Time box of 3-4 hours.
4. **Daily Scrum**: 15-30 minutes daily.
5. **Sprint Review**: Time box of 2-3 hours.
6. **Sprint Retrospective**: Time box of 2-3 hours.
7. **Product Backlog Refinement**: Time box of 2-3 hours.

**Burn Down Chart**

A **burndown chart** is used to track the team's progress over the sprint. It shows the remaining work (in hours or story points) against time, helping to predict whether the team will complete the sprint on time.