

Waterfall Methodology:

- Waterfall methodology is a linear and **sequential approach** to software development. It contains a certain number of steps that need to be followed and the result of one step will be input to the next step.
- **Steps** in Waterfall methodology:
 1. Requirement Analysis: Gathering and documenting what the project needs to accomplish.
 2. System Design: Defining what requirements need to achieve through architecture and design.
 3. Development: Develop the system based on System Design.
 4. Integration and testing: Integrate different features and develop testing for those.
 5. Deployment: Deploy the build system to customers.
 6. Maintenance Ongoing support and enhancement after the product is deployed.
- The **drawbacks** of the Waterfall methodology include:
 1. Inflexibility: Difficult to make changes once a phase is completed.
 2. Late Testing: Issues are discovered late in the process.
 3. Delayed Feedback: Clients see the product only near the end, risking dissatisfaction.
 4. Assumption of Perfect Requirements: Hard to accommodate evolving requirements.
 5. Long Project Timelines: Sequential phases can extend project durations.
 6. Risk of Obsolescence: Delivered product may be outdated.
 7. Limited Customer Involvement: Minimal client interaction during development.
 8. Overhead: Extensive documentation and formal reviews can increase costs and delays.

Agile Methodology:

- Agile is a way of developing software that focuses on being flexible, working closely with customers, and delivering small, working pieces of the software frequently. Instead of planning everything out from the beginning and sticking to a strict plan, Agile teams adjust their plans as they go based on feedback and changes in requirements. The goal is to quickly respond to changes and continuously improve the product.

- **Core Values**

- Individuals and Interactions over Processes and Tools:
 - Emphasizes the importance of people and their interactions rather than relying solely on processes and tools.
- Working Software over Comprehensive Documentation:
 - Focuses on delivering functional software over producing extensive documentation.
- Customer Collaboration over Contract Negotiation:
 - Prioritizes ongoing collaboration with customers rather than strict contract terms.
- Responding to Change over Following a Plan:
 - Values the ability to adapt to changing requirements over strictly following a predefined plan.

- **Principles**

- a. Customer satisfaction through early and continuous delivery of valuable software.
- b. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- c. Deliver working software frequently, with a preference for a shorter timescale.
- d. Business people and developers must work together daily throughout the project.
- e. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- f. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- g. Working software is the primary measure of progress.
- h. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- i. Continuous attention to technical excellence and good design enhances agility.
- j. Simplicity—the art of maximizing the amount of work not done—is essential.
- k. The best architectures, requirements, and designs emerge from self-organizing teams.
- l. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Scrum

- Scrum is a popular Agile framework that helps teams work together to develop and deliver products incrementally. Here are the key steps followed in Scrum:
 - **Product Backlog Creation:**
 - **Product Owner** creates and maintains a prioritized list of features, enhancements, bug fixes, and requirements known as the product backlog.
 - **Sprint Planning:**
 - At the start of each sprint (a time-boxed period, usually 2-4 weeks), the team holds a sprint planning meeting.
 - The team selects items from the product backlog to work on during the sprint, creating a sprint backlog.
 - They define the sprint goal, which is a clear and concise statement of what the sprint will achieve.
 - **Sprint Execution:**
 - The team works on the tasks in the sprint backlog, aiming to complete them by the end of the sprint.
 - Daily stand-up meetings (daily scrums) are held to discuss progress, plans for the day, and any obstacles faced.
 - **Daily Stand-Up Meetings:**
 - These are short, time-boxed meetings (usually 15 minutes) where team members answer three questions:
 - What did I do yesterday?
 - What will I do today?
 - Are there any impediments in my way?
 - **Sprint Review:**
 - At the end of the sprint, the team holds a sprint review meeting to demonstrate the work completed to the stakeholders.
 - Feedback is gathered and the product backlog is updated based on this feedback.
 - **Sprint Retrospective:**
 - Following the sprint review, the team holds a sprint retrospective meeting to reflect on the sprint.
 - They discuss what went well, what didn't go well, and how they can improve in the next sprint.

- **Repeat:**
 - The next sprint begins immediately after the previous one concludes, starting with sprint planning.
- By following these steps, Scrum helps teams deliver products incrementally, adapt to changes quickly, and continuously improve their processes.
- Three important roles in a Scrum team which are product owner, scrum master and Development team. Here is their specified roles:
 - **Product Owner:**
 - **Backlog Management:** Creates, prioritizes, and maintains the product backlog to ensure it reflects the project's needs and goals.
 - **Requirement Clarification:** Provides clear requirements and acceptance criteria for backlog items.
 - **Stakeholder Liaison:** Acts as the primary point of contact between stakeholders and the Scrum team, gathering feedback and ensuring the product meets their needs.
 - **Vision and Direction:** Clearly communicates the product vision and goals to the team, aligning their work with the overall project objectives.
 - **Scrum Master:**
 - **Facilitation:** Facilitates Scrum ceremonies (e.g., sprint planning, daily stand-ups, sprint reviews, and retrospectives) to ensure they are productive and focused.
 - **Impediment Removal:** Identifies and removes obstacles that hinder the Development Team's progress.
 - **Process Improvement:** Guides the team in adopting and improving Agile practices and principles.
 - **Team Support:** Acts as a coach and mentor, helping the team to self-organize and work collaboratively.
 - **Shielding the Team:** Protects the team from outside interruptions and distractions to maintain focus on the sprint goals.
 - **Development Team:**
 - **Execution:** Works on the tasks in the sprint backlog to deliver potentially shippable increments of the product by the end of each sprint.
 - **Self-Organization:** Organizes and manages their own work without being directed by others.

- **Collaboration:** Collaborates closely with each other and with the Product Owner to ensure understanding of requirements and to deliver quality work.
- **Quality Assurance:** Ensures the quality of the product through practices like pair programming, code reviews, and continuous integration.
- **Continuous Improvement:** Participates in retrospectives and actively seeks ways to improve their processes and performance.

Scrum Artifacts

- **Product Backlog:**
 - **Description:** A prioritized list of all desired work on the project. It includes features, bug fixes, technical tasks, and knowledge acquisition.
 - **Purpose:** To capture and prioritize requirements for the product, ensuring that the most valuable work is completed first.
 - **Owner:** Maintained by the Product Owner.
- **Sprint Backlog:**
 - **Description:** A subset of the Product Backlog that the Development Team commits to completing during a specific sprint. It includes tasks and the sprint goal.
 - **Purpose:** To provide a detailed plan for the sprint, outlining what will be delivered and how it will be achieved.
 - **Owner:** Managed by the Development Team.
- **Product Increment:**
 - **Description:** The sum of all Product Backlog items completed during a sprint and all previous sprints. It is a potentially releasable piece of the product that meets the Definition of Done.
 - **Purpose:** To ensure that the product is incrementally improved with each sprint, delivering value and maintaining a potentially shippable state.
 - **Owner:** Created by the Development Team.