1. SCRUM

What is scrum? *"A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value."*

So it’s a *framework*, and like any other framework it can be, and regularly is, used the wrong way. Using scrum effectively requires not merely adopting the structure set out by scrum, but having a deep understanding and appreciation for agile principles across the entire team.

Scrum consists of three roles: **Product Owner**, **Scrum Master**, and **Development Team**; four ceremonies: **Planning Meeting**, **Daily Scrum**, **Sprint Review**, and **Sprint Retrospective**; and three artifacts: **Product Backlog**, **Sprint Backlog**, and **Product Increment**. It is organized into regular time frames, which we call **sprints**. Sprints should last between one and four weeks.

The **Product Owner**, or PO, is responsible for guiding the project’s direction. As new tasks and features are determined, the PO adds them to the **Product Backlog**. A **sprint** starts with a **Planning Meeting** where the **Development Team** selects the tasks from the Product Backlog to work on and plans how they will be implemented. That is followed by development, during which the Dev Team uses the **Sprint Backlog** to track progress and meets for the **Daily Scrum** in order to synchronize activities and adjust the plan, if needed. The result of development should be a **Product Increment**, something that can be applied to the product and released immediately. At the end of the sprint, the Product Increment is presented to the Product Owner at the **Sprint Review**, where the product backlog is augmented if further changes are needed. Afterwards, the whole team attends the **Sprint Retrospective** (also known as the Pub Meeting) where they talk about the work process and how it can be improved.

KANBAN

Kanban is a simple method that aims for just-in-time delivery while not overloading the team members. It is similar to scrum in that the goal is to deliver maximum value at the end, but it is much more flexible than scrum.

There are no strict procedures that you should follow, and no strict way you should implement and use kanban; it is, rather, a set of principles and practices, and you can choose from these practices to suit your needs. But there is one most-often used implementation of kanban in software development that includes the usage of a **kanban board**, consisting of columns representing stages of work, and tasks.

Columns represent the state of a task in the development process. The simplest example consists of three columns: “To Do,” “In Progress,” and “Done.” So, tasks are added to “To Do,” moved to “In Progress” when development starts, and considered “Done” when moved to the last column. But of course, it could be more complex:

EXTREME PROGRAMMING

Extreme Programming (XP) is a highly disciplined management method, which focuses on continually improving quality and speed of software delivery. The development team works closely with customers, continuously planning, testing and providing feedback to developers, to quickly deliver valuable software. Like Scrum, delivery is in iterations of 1-3 weeks.

**The Extreme Programming method follows these lifecycle stages:**

* **Planning** – setting goals for the entire project and specific iterative cycles. Planning is done together with the customer, who formulates a vision of the product and defines user stories. Developers estimate the stories and turn them into more granular tasks.
* **Designing** – developers are responsible for designing the main features in the next iteration. XP emphasizes simplicity, so design should be as simple as possible.
* **Coding** – throughout an iteration, developers constantly refactor the code to bring it down to the simplest, most elegant form possible. Pair programming is commonly used in XP projects to boost innovation and code quality.
* **Testing** – testing is done in tandem with writing the code, not afterwards as a separate development stage, typically using Test Driven Development (TDD).

2. Members of an Agile team and their role

An Agile team working in Scrum has three roles:

* The Product Owner – Often an executive or key stakeholder, the Product Owner has a vision for the end product and a sense of how it will fit into the company’s long-term goals. This person will need to direct communication efforts, alerting the team to major developments and stepping in to course-correct and implement high-level changes as necessary.
* The Scrum Master – The Scrum Master is most akin to a project manager. They are guardians of process, givers of feedback, and mentors to junior team members. They oversee day-to-day functions, maintain the Scrum board, check in with team members, and make sure tasks are being completed on target.
* The Team Member – Team members are the makers: front- and back-end engineers, copywriters, designers, videographers, you name it. Team members have varied roles and skills but all are responsible for getting stuff done on time and in excellent quality.

3. 5 project management tools.

Ace project

ProWorkFlo

Zoho project

TeamWork CRM

BuilderTrend

4. the Phases of Software Development Life Cycle.

Planning

Analysis

Design

Development & Implementation

Teasting

Maintenance