**Agile**

**Agile Overview**

**Agile Project Management**

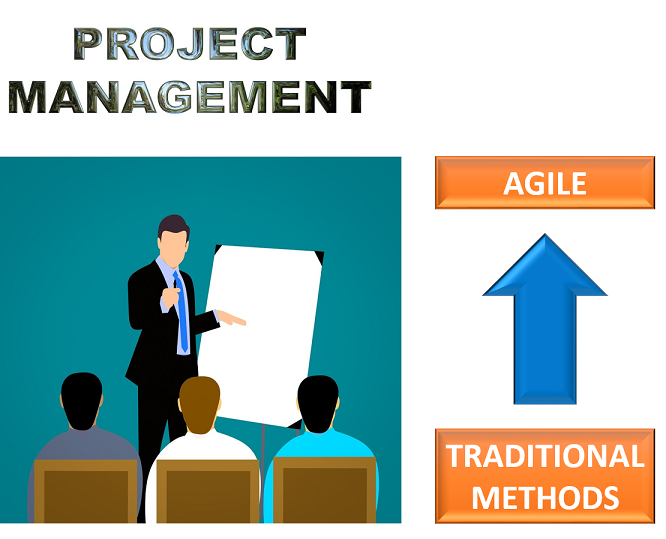
**Dividing Product Development Life Cycle into Small Components**

**Customer's Desired Requirements**

**Building the Right Product**

**Agile Project Management**

Software projects face the risk of failure during the life cycle due to misuse of resources, wrong prioritization and inaccurate understanding of customer demands. This has led companies to experiment with a variety of new methods to produce more productive projects. Agile is a new approach to project management methods resulting from these experiments. This method is based on an incremental and iterative approach that helps teams deliver a product to their customers faster and more desirable. In this respect, the framework of agile promises **significant advantages over traditional methods**. In recent years, most companies have adopted the agile approach, which they find more successful than traditional methods. In this context, we can say that almost every software developer has used the agile method in some form.



**Dividing Product Development Life Cycle into Small Components**

Agile provides a light framework for development teams. That helps them focus on fast delivery while maintaining functionality. In Agile, the product development life cycle is divided into small components (called iterations), so the product can be easily and rapidly developed and tested. In this way, changes can be made without having to wait for the final product.



**Customer's Desired Requirements**

The agile method provides that the product is optimized throughout the development process. Using **iterative planning** and getting **feedback** ensures that a delivered product is tailored to the customer's requirements. The status of the project is measured and evaluated throughout the process to easily adapt to changing needs. Measurement and evaluation provide accurate and early visibility into the development of each project.



**Building the Right Product**

It can be said that the agile method helps companies build the **right product**. Rather than attempting to sell the product before it is released, the agile method encourages teams to improve the product during its development. This helps the company to be in the marketplace as profitable as possible. That retains the vital business value and guarantees the desirable product. This is the reason why the agile method is an appealing option for both partners and developers.

**History of Agile**

**The Agile Manifesto**

In 2001, 17 independent Software Leaders met in America for brainstorming. They aimed to find out how to develop software better by using different knowledge and approaches. After two days of brainstorming, they released the Agile Manifesto.

The Agile Manifesto was a powerful statement, carefully crafted using only 68 words. Everyone agreed that **the Agile Manifesto was both short and authoritative**. While traditional methods advocated a stable plan and avoided changes, the manifesto focused on people, communication, the product, and flexibility.

The Agile Manifesto basically mentions the following four values:

**1. Individuals and interactions over processes and tools:**

Valuing people more highly than processes or tools is easy to understand. Because it is the human who responds to business needs and drives the development process.

**2. Working software over comprehensive documentation:**

Historically, great amounts of time have been spent on documenting the product for development and ultimate delivery. Technical specifications and requirements, interface design documents, test plans, documentation plans, and approvals required for each. The Agile Manifesto values documentation, but it values working software much more.

**3. Customer collaboration over contract negotiation:**

With traditional development models such as Waterfall, customers negotiate the requirements for the product, usually in great detail, prior to any work starts. This means the customer is involved in the process of development before development began and after it was completed, but not during the process. However, the Agile Manifesto describes a customer who is engaged and collaborates throughout the development process.

**4. Responding to change over following a plan:**

Traditional software development methods regard change as an expense, so it is to be avoided. With Agile, the shortness of an iteration cycle means priorities can be shifted from iteration to iteration and new features can be inserted into the next iteration. Agile’s view is that changes always improve a project and provide additional values.



**TheAgileManifesto**

**Agile Principles**

In the months following the publication of the Agile Manifesto, the original signatories continued to communicate. They augmented the four values of the manifesto with the following 12 principles.

1. Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software.
2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with preference to the **shorter timescale**.
4. Business people and developers must **work together** daily throughout the project.
5. Build projects around **motivated individuals**. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is **face to face conversation**.
7. Working software is the primary **measure of progress**.
8. Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a **constant pace** indefinitely.
9. **Continuous attention** to technical excellence and **good design** enhances agility.
10. Simplicity -the art of **maximizing the amount of work** not done- is essential.
11. The best architectures, requirements, and designs emerge from **self-organizing teams**.
12. At **regular intervals**, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

So far you have learned about the agile method. In the next lesson, you will watch a comprehensive interactive video about agile.

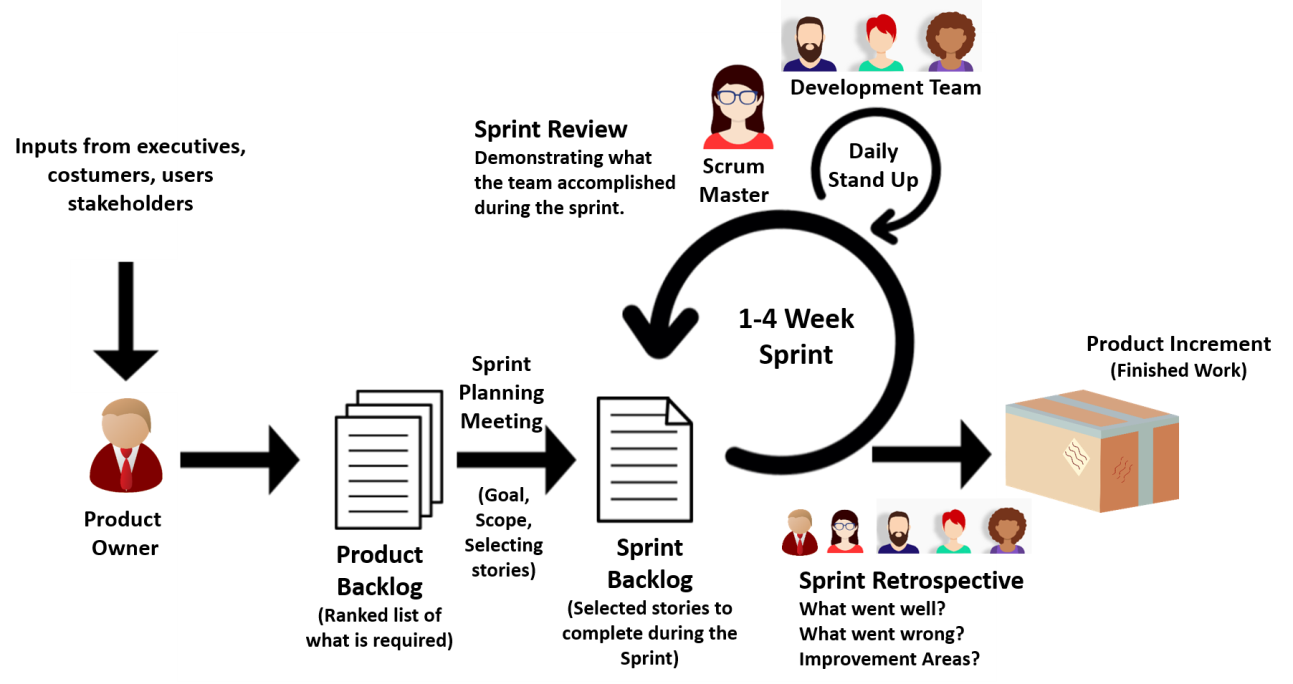
**Q**: Do you know about Agile Manifesto & its Principles? Explain in brief.  
**A**: There are four values in the manifesto. Individuals and interactions, working software, customer collaboration and responding to the changes are the values. Stemming from these values there are 12 principles in agile. These principles can be summarized as to satisfy the customer, to welcome changing requirements, good cooperation between business people and developers (working together), face to face conversation, motivated individuals and simplicity.

- **Interview Q&A**

**Scrum**

**Scrum Framework and Sprint Concept**

Scrum is the most common methodology to implement the Agile. It is an iterative development model used in complex software development processes. In scrum, larger projects are divided into smaller parts that can be managed with **sprints** . Sprints are the periods from one to four weeks. It can be even a few days when needed. Steady sprint length reduces variability; a scrum team can safely predict what they can do on each sprint based on what they have done in previous sprints. The implementation of sprints allows scrum teams to make arrangements for instant improvement, rather than at the end of the project. At the end of each sprint, something remarkable is revealed. For gaining feedback from users or investors, the product produced during each iteration should be demonstrated. The scrum framework defines specific roles, artifacts, and activities for projects. The following figure shows all of these components of the scrum framework that we will discuss later one by one. |



|  |
| --- |
| *The Scrum Framework* |

**Tips:**

* Sprint is also called iteration.

**Q**: What is the duration of a scrum sprint?  
**A**: It depends on the number of people in the development team and the size of the project. In general, a scrum sprint is completed in 1-4 weeks.

- **Interview Q&A**

**Scrum Roles**

There are three main roles in scrum projects. These are the **Product Owner**, **Scrum Master**, and **Development Team**.

**Product Owner** (PO) is the business representative in the team and speaks for the needs of the project for maximizing the value delivered in each sprint. The product owner represents stakeholders and is the voice of the customer. Therefore, the product owner works together with stakeholders and prioritizes the product requirements.

**Scrum Master** coaches the team, protects the team from organizational distraction, clears any obstacles encountered and helps team members focus on what they do. Scrum Master ensures that scrum is understood well by the team members and it is working properly. Scrum Master constantly improves the team's environment. While the product owner has a directing role, Scrum Master has an enabling role in a scrum team.

A **Development team** usually consists of 3-9 people and performs daily tasks. The team is project-oriented and dedicated to the success of the project. Each team member is very talented that is, the team members are skilled in certain subjects. Each member can do more than one job on the project. Discipline and integrity are the key terms for a successful team.

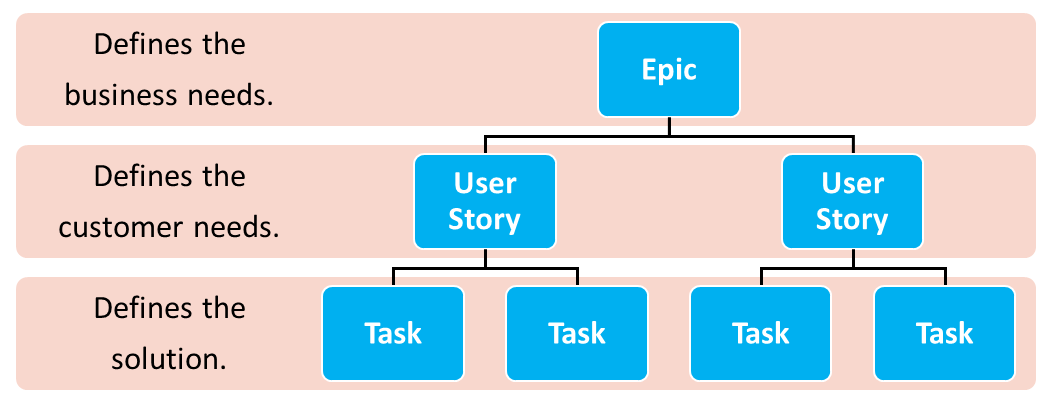


ScrumRoles

**Q**: What is the role of the Scrum Master?  
**A**: Scrum Master coaches the team, protects the team from organizational distraction, clears any obstacles encountered and helps team members focus on what they do. Scrum Master ensures that scrum is understood well by the team members and it is working properly. Scrum Master constantly improves the team's environment. While the product owner has a directing role, Scrum Master has an enabling role in a scrum team.

- **Interview Q&A**

**Epic, User Story and Task**

The three basic terms used in scrum projects are epic, user story and task. In order to fully comprehend a scrum project, it is very important to understand what these concepts are. The hierarchy between these concepts is explained as follows.

**Tiobe\_Index**

**Hierarchy Between Epic, User Story and Task**

**Epic**

An epic refers to a set of jobs that cannot be easily achieved in a single sprint. It usually takes months to perform an epic. Normally, it refers to a series of requirements that have not yet been split into user stories. We can also consider the epic as one of the major goals for your agile team to work on, yet not simplified and divided into various tasks.

Epics are generally large-scaled and do not contain details. They must be divided into multiple user stories before being worked on. Therefore, they are located at the top of the hierarchy. An epic could be a feature of the product, customer demand or business requirement.

***Examples of Epics:***

* As a bank, we want a facial recognition system in our branches.
* As the marketing department, we want a mobile application and a website to reach more customers.

**User Story**

At the hierarchy, the user story is between the epic and the task. It describes a software feature from the customer’s perspective and includes the **type of user**, **what they want**, and **why they want it**. Therefore, it answers the ‘who’, ‘what’ and ‘why’ in a simple language. The product owner has the responsibility of user stories.

Leaving out the technical aspect, it should describe the behavior from a user’s perspective.

**💡Tips:**

* Template of a User Story:  
  As a < type of user >, I want < some goal > so that < some reason>

***Examples of User Stories:***

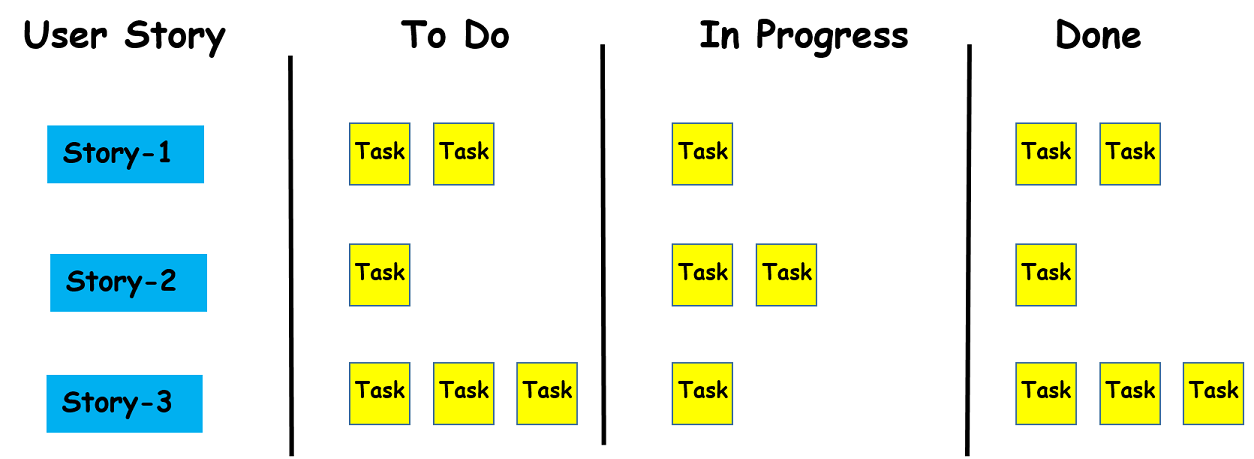
* As a registered user, I want to add items to the cart so that I can purchase multiple items at once.
* As a student, I want to apply for the exam online so that I can save time.

**Q**: What is a User Story?

- **Interview Question**

Task

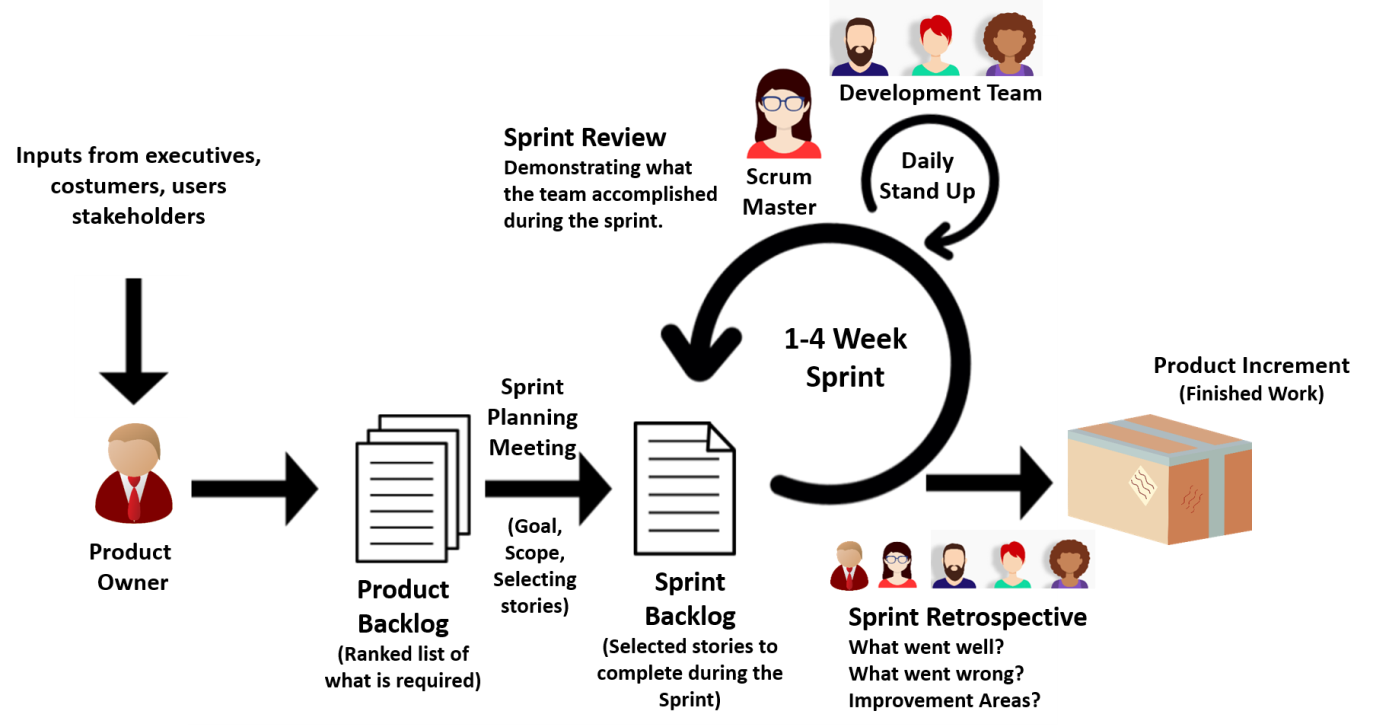
Tasks are detailed pieces of work that are necessary to realize a user story. In other words, they define the solutions for customer needs. The time for tasks can range from a couple of hours to several hours and assigned to a team member. Tasks are displayed on a scrum board for easy tracking, as shown in the figure below.



The scrum board helps to visualize the tasks and usually contains cards or post-it notes on a whiteboard. The board is usually divided into three categories: *to do*, *in progress*, and *done*. The team members update the board by moving task cards through the columns on the board.

**Scrum Artifacts**

In scrum, artifacts serve to capture the common understanding of the team. The Scrum framework defines three major artifacts. These concepts are **product backlog**, **sprint backlog**, and **product increment**.



**Product Backlog**

The product backlog refers to the **list of everything that needs to be done to complete the project**. Beside all user stories, it also includes technical tasks. The product backlog is the responsibility of the product owner. The product owner fulfills this responsibility by creating the product backlog, **prioritizing** the requirements in the product backlog list and constantly updating this list. The product owner updates the product backlog because once a story is completed, it should be removed from the list. Sometimes, however, new stories are added, as the project grows.

**Sprint Backlog**

The sprint backlog can be defined as a subset of the product backlog. The sprint backlog is generated from the product backlog during the sprint planning meeting at the beginning of each sprint. The user stories selected from the product backlog, which will be completed during the sprint constitute the sprint backlog.

The sprint backlog is **not a flexible list** like a product backlog. That means the sprint backlog is unchanged during the sprint period. Once agreed upon in the sprint planning meeting, the stories, and steps to complete them remain stable during the sprint length. If there are stories left still unfinished by the end of the sprint, they will be added back to the product backlog and addressed during the next sprint.

**Product Increment**

The Product Increment is the sum of all the product backlog items finished during the sprint. In other words, by the end of each sprint, the development team creates a new software that gets built into the main product and this new software is called product increment. The product increment aims to invest in **small amounts** in the new features of the main product. This helps to shorten the time before receiving feedback.

As the name implies, product increment continues to increase within the subsequent sprints. That means each product increment includes all the previous sprint increment values as it is cumulative. It should be a **working product** at the end of each sprint, which means that the product should be ready for shipping. But that doesn’t mean it can be released in the market. It should go through a detailed testing phase and be of high quality before releasing it in the market.

**Q**: Explain the term ‘increment' in Scrum.  
**A**: The Product Increment is the sum of all the product backlog items finished during the sprint. In other words, by the end of each sprint, the development team creates a new software that gets built into the main product and this new software is called product increment. The product increment aims to invest in small amounts in the new features of the main product. This helps to shorten the time before receiving feedback. As the name implies, product increment continues to increase within the subsequent sprints. That means each product increment includes all the previous sprint increment values as it is cumulative.