1.Explain the following agile methodologies Scrum, Kanban, Extreme Programming.

Scrum

Scrum is a process framework used to manage product development and other knowledge work. Scrum is empirical in that it provides a means for teams to establish a hypothesis of how they think something works, try it out, reflect on the experience, and make the appropriate adjustments. That is, when the framework is used properly.
Scrum is structured in a way that allows teams to incorporate practices from other frameworks where they make sense for the team’s context.

Scrum is best suited in the case where a cross functional team is working in a product development setting where there is a non trivial amount of work that lends itself to being split into more than one 2 – 4 week iteration.

Values

Teams following scrum are expected to learn and explore the following values:

Commitment

Team members personally commit to achieving team goals

Courage

Team members do the right thing and work on tough problems.

Focus

Concentrate on the work identified for the sprint and the goals of the team.

Openness

Team members and stakeholders are open about all the work and the challenges the team encounters.

Respect

Team members respect each other to be capable and independent.

Principles

The following principles underpin the empirical nature of scrum:

Transparency

The team must work in an environment where everyone is aware of what issues other team members are running into. Teams surface issues within the organization, often ones that have been there for a long time, that get in the way of the team’s success.

Inspection

Frequent inspection points built into the framework to allow the team an opportunity to reflect on how the process is working. These inspection points include the Daily Scrum meeting and the Sprint Review Meeting.

Adaptation

The team constantly investigates how things are going and revises those items that do not seem to make sense.

Practices

Events

Sprint

The Sprint is a timebox of one month or less during which the team produces a potentially shippable product Increment. Typical characteristics of Sprints:

- Maintain a consistent duration throughout a development effort
- A new Sprint immediately follows the conclusion of the previous Sprint
- Start date and end date of Sprint are fixed

Sprint Planning

A team starts out a Sprint with a discussion to determine which items from the product backlog they will work on during the Sprint. The end result of Sprint Planning is the Sprint Backlog.

Sprint Planning typically occurs in two parts. In the first part, the product owner and the rest of the team agree on which product backlog items will be included in the Sprint.

In the Second Part of Sprint Planning, the team determines how they will successfully deliver the identified product backlog items as part of the potentially shippable product increment. The team may identify specific tasks necessary to make that happen if that is one of their practices. The product backlog items identified for delivery and tasks if applicable, makes up the Sprint Backlog.

Once the team and product owner establish the scope of the Sprint as described by the product backlog items no more items can be added to the Sprint Backlog. This protects the team from scope changes within that Sprint.

Daily Scrum

The Daily Scrum is a short (usually limited to 15 minutes) discussion where the team coordinates their activities for the following day. The Daily Scrum is not intended to be a status reporting meeting or a problem solving discussion.

Sprint Review

At the end of the Sprint, the entire team (including product owner) reviews the results of the sprint with stakeholders of the product. The purpose of this discussion is to discuss, demonstrate, and potentially give the stakeholders a chance to use, the increment in order to get feedback. The Sprint Review is not intended to provide a status report. Feedback from the sprint review gets placed into the Product Backlog for future consideration.

Sprint Retrospective

At the end of the Sprint following the sprint review the team (including product owner) should reflect upon how things went during the previous sprint and identify adjustments they could make going forward. The result of this retrospective is at least one action item included on the following Sprint’s Sprint Backlog.

Artifacts

Product Backlog

The product backlog is an ordered list of all the possible changes that could be made to the product. Items on the product backlog are options, not commitments in that just because they exist on the Product Backlog does not guarantee they will be delivered.

The Product Owner maintains the product backlog on an ongoing basis including its content, availability, and ordering.

Sprint Backlog

The Sprint Backlog is the collection of product backlog items selected for delivery in the Sprint, and if the team identifies tasks, the tasks necessary to deliver those product backlog items and achieve the Sprint Goal.

Increment

The increment is the collection of the Product Backlog Items that meet the team’s Definition of Done by the end of the Sprint. The Product Owner may decide to release the increment or build upon it in future Sprints.

Definition of Done

The definition of done is a team’s shared agreement on the criteria that a Product Backlog Item must meet before it is considered done.

Roles

The Product Owner

The product owner is a role team responsible for managing the product backlog in order to achieve the desired outcome that the team seeks to accomplish.

The product owner role exists in Scrum to address challenges that product development teams had with multiple, conflicting direction, or no direction at all with respect to what to build.

The Scrum Master

The scrum master is the team role responsible for ensuring the team lives agile values and principles and follows the processes and practices that the team agreed they would use.

The name was initially intended to indicate someone who is an expert at Scrum and can therefore coach others.

The role does not generally have any actual authority. People filling this role have to lead from a position of influence, often taking a servant-leadership stance.

The Development Team

The development team consists of the people who deliver the product increment inside a Sprint.

The main responsibility of the development team is to deliver the increment that delivers value every Sprint. How the work is divided up to do that is left up to the team to determine based on the conditions at that time.

Lifecycle

Scrum is a framework that allows development teams flexibility to respond to changing situations. This framework has sufficient control points in place to ensure the team does not stray from the desired outcome, and that issues can be identified and resolved and process adjustments made while the effort is still underway.

The Scrum Lifecycle starts with a prioritized backlog, but does not provide any guidance as to how that backlog is developed or prioritized.

The Scrum Lifecycle consists of a series of Sprints, where the end result is a potentially shippable product increment. Inside of these sprints, all of the activities necessary for the development of the product occur on a small subset of the overall product. Below is a description of the key steps in the Scrum Lifecycle:

1. Establish the Product Backlog.
2. The product owner and development team conduct Sprint Planning. Determine the scope of the Sprint in the first part of Sprint Planning and the plan for delivering that scope in the second half of Sprint Planning.
3. As the Sprint progresses, development team perform the work necessary to deliver the selected product backlog items.
4. On a daily basis, the development team coordinate their work in a Daily Scrum.
5. At the end of the Sprint the development team delivers the Product Backlog Items selected during Sprint Planning. The development team holds a Sprint Review to show the customer the increment and get feedback. The development team and product owner also reflect on how the Sprint has proceeded so far and adapting their processes accordingly during a retrospective.
6. The Team repeats steps 2–5 until the desired outcome of the product have been met.

Kanban

Kanban is a method for managing the creation of products with an emphasis on continual delivery while not overburdening the development team. Like Scrum, Kanban is a process designed to help teams work together more effectively.

Kanban is based on 3 basic principles:

- Visualize what you do today (workflow): seeing all the items in context of each other can be very informative
- Limit the amount of work in progress (WIP): this helps balance the flow-based approach so teams don’t start and commit to too much work at once
- Enhance flow: when something is finished, the next highest thing from the backlog is pulled into play

Kanban promotes continuous collaboration and encourages active, ongoing learning and improving by defining the best possible team workflow.

Benefits:

- Shorter cycle times can deliver features faster.
- Responsiveness to Change:
- When priorities change very frequently, Kanban is ideal.
- Balancing demand against throughput guarantees that most the customer-centric features are always being worked.
- Requires fewer organization / room set-up changes to get started
- Reducing waste and removing activities that don’t add value to the team/department/organization
- Rapid feedback loops improve the chances of more motivated, empowered and higher-performing team members

Extreme Programming

Extreme Programming is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

Values

The five values of Extreme Programming are communication, simplicity, feedback, courage, and respect and are described in more detail below.

Communication

Software development is inherently a team sport that relies on communication to transfer knowledge from one team member to everyone else on the team. XP stresses the importance of the appropriate kind of communication – face to face discussion with the aid of a white board or other drawing mechanism.

Simplicity

Simplicity means “what is the simplest thing that will work?” The purpose of this is to avoid waste and do only absolutely necessary things such as keep the design of the system as simple as possible so that it is easier to maintain, support, and revise. Simplicity also means address only the requirements that you know about; don’t try to predict the future.

Feedback

Through constant feedback about their previous efforts, teams can identify areas for improvement and revise their practices. Feedback also supports simple design. Your team builds something, gathers feedback on your design and implementation, and then adjust your product going forward.

Courage

Kent Beck defined courage as “effective action in the face of fear” (Extreme Programming Explained P. 20). This definition shows a preference for action based on other principles so that the results aren’t harmful to the team. You need courage to raise organizational issues that reduce your team’s effectiveness. You need courage to stop doing something that doesn’t work and try something else. You need courage to accept and act on feedback, even when it’s difficult to accept.

Respect

The members of your team need to respect each other in order to communicate with each other, provide and accept feedback that honors your relationship, and to work together to identify simple designs and solutions.

2.Who are the members of an agile team and what are their roles?

- **Development team:** The people who create the product. Programmers, testers, designers, writers, and anyone else who has a hands-on role in product development are development team members.
- **Product owner:** This person is responsible for bridging the gaps between the customer, business stakeholders, and the development team. The product owner is an expert on the product and the customer’s needs and priorities and helps clarify project requirements.
  Product owners make the decisions about what the product does and does not include. Add to that the responsibility of deciding what to release to the market and when to do it, and you see that you need a smart and savvy person to fill this role.
- **Scrum master:** This scrum team member is responsible for supporting the development team, clearing organizational roadblocks, and keeping processes

true to agile principles.

A scrum master is different from a project manager. Teams using traditional project approaches work for a project manager. A scrum master, on the other hand, is a servant-leader who supports the team so that it is fully functional and productive. The scrum master role is an enabling role, rather than an accountability role.

- **Stakeholders:** A stakeholder is anyone with an interest in the project. Stakeholders are not ultimately responsible for the product, but they provide input and are affected by the project's outcome. The group of stakeholders is diverse and can include people from different departments, or even different companies. Stakeholders can include:
    - The customer
    - Technical people, such as infrastructure architects or system administrators.
    - The legal department, account managers, sales people, marketing experts, and customer service representatives.
    - Product experts besides the product owner.
- Stakeholders may help provide key insights about the product and its use. Stakeholders may work closely with the product owner during the sprint and give feedback about the product during the sprint review at the end of each sprint.
- **Agile mentor:** Someone who has experience implementing agile projects and can share that experience with a project team. A mentor is a great idea for any area in which you want to develop new expertise. The agile mentor can provide valuable feedback and advice to new project teams and to project teams that want to perform at a higher level.

3.List 5 Project Management Tools

1.Asana

2.Trello

3.Jira

4.Slack

5.Gantt Charts

4.List the Phases of Software Development Life Cycle

1.Planning

2.Analysis

3.Design

4.Development/Implemetation

5.Testing

6.Deployment

7.Maintenance