**Q. Write short notes on the following:**

* **Scrum**

Scrum is a lightweight Agile project management framework that can be used to

manage iterative and incremental projects of all types. Due to its simplicity, proven

productivity, and ability to incorporate various overarching practices promoted by other

Agile methods, it has gained immense popularity over the years.

**Characteristics:**

1. Scrumis the best choice when the *exact amount of work can’t be estimated* and the *release date is not fixed* as Scrum works well for long term, complex projects that require stakeholder feedback.
2. A Scrum team, comprising 5-9 people, *does not include any of the traditional software engineering roles*. Everyone on the project works together to complete the set of work they have collectively committed to complete within a sprint.
3. In Scrum, Product owner (someone from product management or marketing, a key stakeholder or a key user) usually works closely with the Scrum team to identify and [prioritize system functionality by creating a Product Backlog](https://www.blueprintsys.com/storyteller/backlog-management).
4. The *Product Backlog* consists of whatever needs to be done to successfully deliver a working software system, including features, bug fixes, [non-functional requirements](https://www.blueprintsys.com/blog/the-keys-to-accelerating-non-functional-requirements-definition), etc.
5. At the start of each sprint, a *sprint planning meeting* is held, during which the product owner presents the top items on the product backlog to the team. The work to be completed in the coming sprint is then moved from the *Product backlog to a Sprint backlog.*
6. Once a Sprint’s Product Backlog is committed, no additional functionality can be added to the Sprint except by the team.
7. Once a Sprint has been delivered, the Product Backlog is analyzed and reprioritized, if necessary, and the next set of deliverables is selected for the next Sprint.

* **Lean Development**

Lean Developmentis an iterative Agile methodology that focuses the team on delivering

value to the customer through effective value stream mapping. It is a highly flexible,

evolving methodology without rigid guidelines, rules, or methods.

**Characteristics:**

1. Lean development eliminates waste by asking users to [select only the truly valuable features for a system](https://www.blueprintsys.com/blog/automatic-feature-decomposition-surprising-way-ensure-agile-project-success), prioritize those features, and then work to deliver them in small batches.
2. Lean development uses the idea of a work product being “pulled” via customer request.
3. It relies on rapid and reliable feedback between programmers and customers, emphasizing the speed and efficiency of development workflows.
4. It gives decision-making authority to individuals and small teams since this has been proven to be a faster and more efficient method than a hierarchical flow of control.
5. Lean Development strongly recommends that automated unit tests be written at the same time the code is written.
6. It concentrates on the efficient use of team resources, trying to ensure that everyone is as productive as possible for the maximum amount of time.

* **Extreme programming (XP)**

Extreme programming is a disciplined approach for high-quality agile software

development, focused on speed and continuous delivery. It is intended to improve

software quality and responsiveness in the wake of changing customer requirements.

**Characteristics:**

Extreme programming has the following five values:

1. *Communication:* Extreme programming stresses on the importance of the appropriate

kind of communication - face to face discussion with the aid of a white board or other

drawing mechanism.

1. *Simplicity:* XP focuses on keeping the design of the system as simple as possible so that

it is easier to maintain, support and revise.

1. *Feedback:* Through constant feedback about their previous efforts, teams can identify

areas for improvement and revise their practices.

1. *Courage:* XP stresses on gathering courage to raise organizational issues that reduce

your team’s effectiveness.

1. *Respect:* The members of a team should 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.

* **Adaptive Software Development (ASD)**

Adaptive Software Development is a method to build complex software and system

with overall emphasis being on the dynamics of self-organizing teams, interpersonal

collaboration, and individual and team learning.

**Characteristics:**

ASD lifecycle incorporates the following phases:

1. *Speculation:* The first phase of ASD lifecycle involves initiation of project and planning. The project plan uses project initiation information like project requirements, user needs, customer mission statement etc, to define a set of release cycles that the project wants.
2. *Collaboration:* This phase involves collaboration and communication between different team members while ensuring that individual creative thinking is encouraged. In this phase, team members are expected to criticize without animosity, assist without resentment and communicate problems to find an effective solution.
3. *Learning:* Learning helps the workers to increase their level of understanding over the project. This ensures that any overestimation of their own understanding of the technology which may not lead to the desired result can be avoided.

* **Feature Driven Development**

Feature Driven Development is a model-driven, short-iteration process that was built

around software engineering best practices such as domain object modeling,

developing by feature, and code ownership. The blending of these practices that

resulted in a cohesive whole is the best characteristic of FDD.

**Characteristics:**

1. Feature Driven Development begins by establishing an overall model shape, which will

result in a feature list. The features are small, “useful in the eyes of the client” results.

1. It then continues with a series of two-week “plan by feature, design by feature, build by

feature” iterations.

1. FDD’s main purpose is to deliver tangible, working software in a timely manner,

repeatedly.

1. The advantage of using FDD is that it is scalable even to large teams due to the concept

of ‘just enough design initially’ (JEDI).

1. Because of its feature-centric process, FDD is a great solution to maintain control for

incremental and inherently complex Agile project management.