**Write short notes on following -**

* **Scrum**

Scrum is an agile project management methodology or framework. Scrum is widely used by software development teams with the goal of delivering new software capability every 2-4 weeks. In fact it's the [most popular agile methodology](https://resources.collab.net/agile-101/agile-methodologies). Scrum uses Iterative process.

**Advantage of using Scrum framework -**

## Fast moving

## Money efficient.

## Works by dividing the large product into small sub-products.

## Customer satisfaction is given utmost importance.

## Scrum is adaptive in nature due to short sprints.

## Relies on constant feedback henceforth the quality of product increases in smaller amount of time

**Disadvantage of using Scrum framework -**

## Does not allow changes into their sprint.

## Scrum framework is not a fully described model Lots of details have to be filled in by the organization using the model.

## Difficult for the Scrum to plan, structure and organize a project that lacks a clear definition.

## Substantial resources are needed for daily scrum meetings.

## **Components of Agile Scrum Development -**

## Scrum Team

## Scrum Events

* Sprint

## Scrum Artifacts

## Scrum Rules

* **Lean Development**

Lean Software Development (LSD) is an [agile framework](https://www.productplan.com/glossary/agile-framework/).

**Key points -**

* Focuses on optimizing development time and resources
* Focuses on eliminating waste
* Deliver only what the product needs

In the Lean approach a team releases a bare-minimum version of its product to the market, learns from users what they like, don’t like and want to be added, and then iterates based on this feedback.

The philosophy of Lean Development came from the manufacturing industry, which originated the lean development process as a way to optimize production and assembly lines to minimize waste and maximize customer value.

### **Strengths -**

## Focused approach that allows more functionality to be delivered in less time

## Eliminates unnecessary activity hence reducing costs

## Empowers the development team to make decisions.

## **Weaknesses -**

## Heavily depends on the team involved, making it not as scalable as other frameworks

## Without strong documentation development mistakes can occur.

An organization must have an outstanding development team, and trust that team implicitly, for this approach to be successful.

* **Extreme programming (XP)**

Extreme programming (XP) is one of the most important software development frameworks of Agile models.

**Key Points -**

* It is used to improve software quality.
* Improve responsiveness to customer requirements.
* Encourages taking the best practices that have worked well in the past in program development projects to extreme levels.

XP is based on the frequent iteration through which the developers implement User Stories.

**Good practices needs to practiced extreme programming:** Some of the good practices that are suggested in XP are given below:

* **Code Review:** Code review detects and corrects errors efficiently.
* **Testing:** Testing code helps to remove errors and improves its reliability. XP suggests test-driven development (TDD) to continually write and execute test cases.
* **Incremental development:** Includes gaining customer feedback and based on this development team come up with new increments every few days after each iteration.
* **Simplicity:** Simplicity makes it easier to develop good quality code as well as to test and debug it.
* **Design:** Good quality design is important to develop a good quality software. So, everybody should design daily.
* **Integration testing:** It helps to identify bugs at the interfaces of different functionalities.

**Applications of Extreme Programming (XP):** Some of the projects that are suitable to develop using XP model are given below:

* **Small projects:** XP model is very useful in small projects consisting of small teams as face to face meetings are easier to achieve.
* **Projects involving new technology or Research projects:** This type of projects face changing requirements rapidly and technical problems. So the XP model is used to complete this type of project.
* **Adaptive Software Development (ASD)**

Adaptive Software Development is an agile methodology.

**Key Traits -**

* Helps in building complex software and systems.
* It focuses on human collaboration.
* Focuses on self-organisation.

ASD’s overall emphasis on the dynamics of self-organizing teams, interpersonal collaboration, and individual and team learning help in creating software project teams that have a much higher likelihood of success.

**ASD “life cycle” incorporates three phases namely:**

1. Speculation
2. Collaboration
3. Learning

These are explained as follows -

1. Speculation:

* Initiation of project and planning is conducted.
* Information like project requirements, user needs, customer mission statement etc, is gathered to define the set of release cycles that the project wants.

2. Collaboration:

For this phase to be successful workers needs to be motivated.

Teamwork and collaboration are focussed on along with emphasizes on individual creativity. People working together must trust each others to

* Criticize without animosity
* Assist without resentment
* Work as hard as possible
* Possession of skill set
* Communicate problems to find effective solutions.

3. Learning:

Learning helps the workers to increase their level of understanding over the project and helps them in estimating the effort needed for the project completion correctly.

Learning process is of 3 ways:

1. Focus groups
2. Technical reviews
3. Project post mortem

* **Feature Driven Development**

Feature-Driven Development is an agile iterative and incremental model. It focuses on progressing the features of a software with each release. It gives special emphasis on progress tracking and reporting so that timely working software can be delivered to the client.

#### **Feature Driven Development Life Cycle includes various stages of development -**

* Build overall model
* Build feature list
* Plan by feature
* Design by feature
* Build by feature

#### **Characteristics of FDD**

* **Short iterative:** Simple and short iterations help to efficiently finish the work on time and assists faster product development.
* **Customer focused:** Inspection of each feature is done by the client and before pushing to main build code.
* **Structured and feature focused:** More than 70% of efforts are given to the last 2 activities.
* **Frequent releases:** Features are released continuously.

#### **Advantages of FDD**

* Successful for larger size of teams and projects.
* Reduction in risk as the whole model and design is built in smaller segments in the form of features.
* Greater accuracy in cost estimation of the project.

#### **Disadvantages of FDD**

* Not good for smaller projects.
* High dependency on lead programmers, designers and mentors.
* Lack of documentation.