**EPP ASSIGNMENT**

**YASH BANSAL UCO2531**

**Write short notes on the following.**

**1) 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.

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.

**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.

**2) Lean Development**

Lean Software Development (LSD) is an agile framework based on optimizing development time and resources, eliminating waste, and ultimately delivering only what the product needs. The Lean approach is also often referred to as the Minimum Viable Product strategy, in which 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.

### LSD’s strengths include:

* Streamlined approach allows more functionality to be delivered in less time
* Eliminates unnecessary activity, and as a result can reduce costs
* Empowers the development team to make decisions, which can also boost morale

### LSD’s weaknesses include:

* Heavily depends on the team involved, making it not as scalable as other frameworks
* Depends on strong documentation, and failure to do so can result in development mistakes

· **3)** **Extreme programming (XP)**

Extreme programming (XP) is one of the most important software development framework of Agile models. It is used to improve software quality and responsive to customer requirements. The extreme programming model recommends taking the best practices that have worked well in the past in program development projects to extreme levels.

Some of the good practices that have been recognized in the extreme programming model and suggested to maximize their use are given below:

* **Code Review:** Code review detects and corrects errors efficiently. It suggests pair programming as coding and reviewing of written code carried out by a pair of programmers who switch their works between them every hour.
* **Testing:** Testing code helps to remove errors and improves its reliability. XP suggests test-driven development (TDD) to continually write and execute test cases. In the TDD approach test cases are written even before any code is written.
* **Incremental development:** Incremental development is very good because customer feedback is gained 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. Extreme programming suggests that the developers should achieve continuous integration by building and performing integration testing several times a day.

**4) Adaptive Software Development (ASD)**

Adaptive Software Development is a method to build complex software and systems. ASD focuses on human collaboration and self-organisation. ASD life cycle incorporates three phases which are explained below.

**1. Speculation:**

During this phase, project is initiated and planning is conducted. The project plan uses project initiation information like project requirements, user needs, customer mission statement etc, to define set of release cycles that the project wants.

**2. Collaboration:**

It is the difficult part of ASD as it needs the workers to be motivated. It collaborates communication and teamwork but emphasizes individualism as individual creativity plays a major role in creative thinking. People working together must trust each others to

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

**3. Learning:**

The workers may have a overestimate of their own understanding of the technology which may not lead to the desired result. Learning helps the workers to increase their level of understanding over the project.

**5) Feature Driven Development**

FDD is a model-driven, and short-iteration process that was developed around software engineering best practices including domain object modeling, developing by feature, and code ownership.

Feature-driven development begins with the establishment of an overall model that is expected to result in the feature list. These features are usually small yet useful and effective on the users’ eyes. Due to this tactic of product development, large teams are allowed to move products forward with a regular success that matters for the clients.

The general objective of FDD is to deliver concrete and flexible software in a short time. Its greatest advantage is that the process is scalable even for large teams. As a result, FDD is considered to be an effective solution to support the control of comparatively complex Agile projects.

### **Advantages of feature-driven development**

* Gives the team a very good understanding of the project’s scope and context.
* Requires fewer meetings. One of the frequent complaints about agile is that there are too many meetings. Scrum uses the daily meetings to communicate. FDD uses documentation to communicate.
* Uses a user-centric approach. With scrum, the product manager is usually considered the end user. With FDD, the client is the end user.
* Works well with large-scale, long-term, or ongoing projects.