## Assignment

Ques: Write short notes on following

· Scrum

· Lean Development

· Extreme programming (XP)

· Adaptive Software Development (ASD)

· Feature Driven Development

## Answer:

### Scrum:

Scrum is a lightweight agile project development framework that can be used to manage iterative and incremental projects of all types. The idea is to break large complex projects into smaller stages and review and adapt them along the way. Scrum is an adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer’s needs through an environment of transparency in communication, collective responsibility and continuous progress.

Scrum is an evolution of Agile Management. Scrum methodology is based on a set of defined practices and roles that must be used during the software development process. Scrum is executed in temporary blocks that are short and periodic, called Sprints, which usually range from 2 to 4 weeks, which is the term for feedback and reflection. Each Sprint is an entity in itself, that provides a variation of the final product that must be able to be delivered to the client with the least possible effort when requested.

The process has a starting point, list of objectives/ requirements that make up the project plan. It is the client of the project that prioritizes these objectives looking at the budget and cost. There are various advantages of scrum development:

1. Easily scalable: As scrum process is iterative and breaks into smaller pieces, it is easy to scale the modules in terms of design, scope, functionality and characteristics.
2. Compliance of expectations: The Product Owner establishes the priorities and on regular basis in spring demos, the Product Owner verifies that the expectations have been met.
3. Flexible to changes: Scrum methodology is designed to adapt to the changing requirements that complex projects entail.
4. Time to market reductions: Clients can use important functions before it is ready.
5. Higher software quality: Improvements after each iteration helps in developing good quality softwares.
6. Reduction of risks: The fact of carrying out the most valuable functionalities in the first place and of knowing the speed with which the team advances in the projects, allows to clear risks effectively.
7. Timely Predictions: We will be better able to estimate the time certain sprints will take and how long will it take to complete the project.

### Lean Development:

Lean software development is a set of principles that can be applied to software development to decrease programming effort, budgeting, and defect rates by one third. It is the concept that emphasizes optimizing efficiency and minimizing waste in the development of software. Lean development offers a solid conceptual framework, values and principles, as well as good practices, derived from experience, that support agile organizations.

The 7 main principles of Lean Software Development are:

1. Eliminate Waste: Waste is anything that does not contribute value to the final product such as inefficient processes or project churn, crossing boundaries/departments, and features that won’t be used. Eliminating waste is the guiding principle in lean software development.
2. Built Quality In: Building quality software is an important principle of lean development. It includes not building legacy code that lacks automated unit and acceptance tests is crucial to continuous integration and nested synchronization.
3. Create Knowledge: Encourage developers to build and record the knowledge necessary to develop a project. This includes all requirements, architecture, and technologies, which are known at project startup. Creating knowledge and recording it over the course of the project ensures that the final product is in line with customer expectations.
4. Defer Commitment: Deferring commitment is sometimes beneficial as more information is available at the latest possible moment before an irreversible decision needs to be made.
5. Deliver Fast: Fastly giving the product to customers can provide more feedback, allowing companies to take a more experimental approach to product/ feature development. Fast delivery is accomplished using short iterations that produce software in small increments by focusing on a limited number of the highest priority requirements.
6. Respect People: Respecting people means giving the development team members freedom to find the best way to accomplish a task, recognizing their efforts, and standing by them when those efforts are unsuccessful.
7. Optimize the Whole: Optimizing the whole development process generates better results than optimizing local processes, which is usually done at the expense of other local processes. Good products are often the result of a unique combination of technology and opportunity which is afforded by a lean software development process.

### Extreme Programming(XP):

Extreme Programming is a Software Development Methodology which is intended to improve software quality and responsiveness to changing customer requirements. It is a type of agile software development which have frequent releases in short development cycles, which is intended to improve productivity and introduce checkpoints at which new requirements can be adopted.

Values of Extreme Programming are:

1. Communication: Everyone on a team works jointly at every stage of the project. 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.
2. Simplicity: Developers strive to write simple code bringing more value to a product, as it saves time and efforts. Simplicity also means address only the requirements that you know about; don’t try to predict the future.
3. Feedback: Team members deliver software frequently, get feedback about it, and improve a product according to the new requirements. Through constant feedback about their previous efforts, teams can identify areas for improvement and revise their practices. Feedback also supports simple design.
4. 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.
5. Courage: Programmers objectively evaluate their own results without making excuses and are always ready to respond to changes.

### Adaptive Software Development(ASD):

Adaptive software development(ASD) is a technique for building complex software and systems. It’s focused on rapid creation and evolution of software systems. There is no pre planned steps in this process. ASD‘s overall emphasis on the dynamics of self-organizing teams, interpersonal collaboration, and individual and team learning yield software project teams. The method follows a dynamic lifecycle instead of the traditional, static lifecycle. Adaptive Software Development practices provide ability to accommodate change and are adaptable in turbulent environments with products evolving with little planning and learning.

The lifecycle of Adaptive software development consists of 3 phases:

1. Speculation: Speculation consists of Project initiation and Risk driven Adaptive cycle planning. During this phase, coders attempt to understand the exact nature of the software and the requirements of the users. This phase relies on bug and user reports to guide the project.
2. Collaboration: Effective collaboration with customers is very important. Communication, teamwork, individual creativity is part of effective collaboration. Collaborate would require the ability to work jointly to produce results, share knowledge or make decisions
3. Learning: This phase consists of Quality review and Final Q/A release. During the learning phase, the newest version of the software is released to users. This generates the bug and user reports used during the first phase of the project, and the cycle repeats itself.

The characteristics of Adaptive Software Development are:

1. It is mission driven in which activities in each development cycle must be justified against the overall project mission.
2. The method is iterative, which means to redo the development instead of doing it right the first time.
3. It is time boxed where we set fixed delivery times for projects.
4. It is risk driven where the development of high risk items should begin as early as possible.

### Feature Driven Development:

Feature-driven development (FDD) is a customer-centric software development methodology known for short iterations. It is a design-oriented agile process in which a project is divided into features. These features are small pieces of a complete project. With the help of FDD, you can create design, code, and code inspection schedules without going into elaborate paperwork.

Feature Driven Development consists of 5 basic activities:

1. Develop An Overall Model: The client and the development team define scope of the project and make an overall model. Detailed domain models are created and then these models are progressively merged into the overall model.
2. Build a Feature List: Information collected from the previous step is used to identify the list of features by functionally decomposing the domain in subject areas. Therefore the feature the team decides to work on must not take a long time to be implemented.
3. Plan by Feature: Now the development of features is planned. It is all about in which order the features will be implemented. Teams are selected and assigned feature sets.
4. Design by Feature: The chief programmer chooses the features and the domain classes that will be involved in designing the feature. Sequence diagrams are drawn. General designs of the features are also finalized.
5. Build by Feature: After the design inspection, the domain expert explains the specifics, class owners start building and implementing all the items necessary to support the design. Code is developed, unit tested and inspected and approved by Chief Programmer who then gives an ok and the completed feature is added to the main build.