**Write short notes on following**

1. **Scrum**

Scrum is an agile Framework for developing, managing and building products. It helps teams work together. It helps the teams to learn from experiences and organize well while working on a problem. It helps the team to collaborate effectively on a project to deliver products of great value. Scrum is a collection of rules, tools, roles that work together to help the team to manage their work effectively. Scrum helps to deal with the changing priorities and gives a better visibility of the project.

The Scrum team consists of a Product Owner, Development Team and Scrum Master. The team is a self organizing team. They choose the best possible way to accomplish the task instead of being directed by some other team. It increases productivity, flexibility and creativity.

Events in a Scrum are sprint, sprint planning, daily scrum, sprint review, sprint retrospective.

A **sprint** is a time frame of one month or less during which a usable and a releasable product increment is created.

**Sprint planning** is deciding the work that needs to be done in the sprint. It takes place before the start of each sprint.Scrum master organizes the sprint planning. Work is selected from the product backlog and is included in the sprint backlog.

**Daily scrum** is a daily meeting for max 15 mins to help the development team synchronize and decide the work for the next 24 hours.

After the end of the sprint, a **sprint review** is done to inspect the work done during the sprint. **Sprint retrospective** is where the scrum team inspects itself and looks for the improvements that can be done in the next sprint.

Scrum values include Courage, Focus, Commitment, Respect and Openness.

1. **Lean Development**

Lean approach is a Minimum Viable Product (MVP) strategy. Application of lean approach to software development is called lean development. It is an agile framework for optimizing development time and minimizing waste and delivering only what product needs. Here a team releases a bare minimum version of the product and then iterates based on the feedback.

Lean development helps to deliver more functionality in less time. It helps to avoid unnecessary activity and also reduces cost. It empowers the development team to make decisions. Development should be excellent for the success of lean development.

Main idea of lean development is to work only on what is most important and eliminate any wastage and also ensure good quality product. This is achieved through lean development principles which are:

* 1. **Eliminate waste**: Eliminate anything that does not add value to the customer.
  2. **Build quality in**: Quality is everyone’s job and not just of the quality analyst. Aim is to build quality products without generating excessive waste.
  3. **Build knowledge**: Properly document and retain valuable learning.
  4. **Defer Commitment**: Team should keep their options open and continuously collect information, rather than making decisions without necessary data.
  5. **Deliver Fast**: It does not mean working for long hours. Idea is to build a simple solution and increment upon it.
  6. **Respect for people**: Communication within the team should be proactive and effective. Healthy conflicts are encouraged. Each one should be empowered to their best.
  7. **Optimize the whole**: Instead of focusing on suboptimizing, focus should be on optimizing the whole.

1. **Extreme programming (XP)**

Extreme programming is an agile software development framework which aims to produce high quality software. It is designed to improve the quality of the software and properly adapt to the changing needs of the customer. It aims to provide iterative and frequent small releases allowing both the team members and customers to review and examine the product throughout the lifecycle.

Extreme programming focuses on customer satisfaction and emphasis team work. Customers, Managers, Developers are all equal partners in a collaborative team. It implements an effective environment making teams highly productive. Teams self organizes around the problem to solve it as effectively as possible.

Extreme programming is built on the top of five fundamental values which are:

* 1. **Simplicity**: Do what is needed and asked for, nothing more.
  2. **Communication**: Everyone is the part of the team and knowledge should be shared among the team members with face to face discussions.
  3. **Feedback**: Through constant feedback on their previous works, teams can identify areas of improvement and revise their practises.
  4. **Respect**: Members of the team respect each other and work together for the accomplishment of the project.
  5. **Courage**: Raise organizational issues that reduce your team's effectiveness. Accept and act on feedback and keep moving forward.

Rules of extreme programming are:

* 1. **Planning**: User stories are written. Project is divided into iterations.
  2. **Managing**: Give the team a dedicated open work space. Set a sustainable pace.
  3. **Designing**: Make a simple design, No functionality is added early, refactor whenever possible.
  4. **Coding**: Code must be written in agreed standards, all the production code is pair programmed, only one pair integrates at a time. Integrate often.
  5. **Testing**: All codes must have unit tests. A code must pass all unit tests before it can be released.

1. **Adaptive Software Development (ASD)**

Adaptive software development is a technique to build large and complex softwares. It focuses on human collaboration and self-organization. It enables the team to quickly adapt to the changing requirements and market needs by evolving their products with continuous planning and learning. It is a shorter version of Rapid action development agile framework.

It focuses on end users which helps to make better and intuitive products. It allows on time delivery of the products.

Teams develop according to a three phase process:

* 1. **Speculation**: During this phase the project is initiated and planning is conducted. Project requirements, user needs, customer needs are used to initialize the project and define a set of release cycles that the project wants. Speculation encourages experimentation and exploration.
  2. **Collaboration**: Large amounts of information is needed to be collected, analyzed and applied to a problem. This can be handled only with team collaboration. Team members work jointly to produce results, share knowledge and make decisions. People working together must trust each other to criticize without animosity, assis without resentment, work hard and communicate problems to find effective solutions.
  3. **Learning**: Team should enhance its knowledge constantly. Members may have an overestimate of their own understanding of technology that may lead to undesired results. Learning helps the members to increase their level of understanding over the project. Learning can be done using technical reviews, focus groups and project retrospective.

Results after each iteration are analyzed to learn the direction of the next. ASD emphasis on dynamics of self-organizing team, interpersonal collaboration, individual and team learning yield software project teams that have a much higher likelihood of success.

1. **Feature Driven Development**

Feature driven development is an agile framework that organizes software development on making progress on features. Features are not necessarily product features, they are more like user stories in scrum. It is an iterative and incremental software development process. Its main purpose is to deliver a tangible, working software repeatedly in a timely manner.

Feature driven development follows a 5 step process which includes:

1. **Develop an overall model**: More shape than content. An object model.
2. **Build a features list**: A list of features grouped into sets and subject areas.
3. **Plan by feature**: Produce a development plan and assign ownership of features as classes to the programmers.
4. **Design by feature**: Add more content to the design model. Design package is produced for each feature.
5. **Build by feature**: Class owners develop codes for their classes. After unit testing and successful code inspection, the completed feature is promoted to the main build.

This 5 step process allows for more rapid development. It allows large teams to move projects forward with continuous success. It leverages a predefined development standards. It does not work efficiently for small projects.