# LAB – 2

**Name – Amit Kumar Sahu**

**Reg. No- 18mis7250**

## 1.Describe about Scrum, Extreme Programming, Feature Driven development, Lean Software Development.

### Scrum

In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented.

Scrum relies on a self-organizing, cross-functional team. The scrum team is self-organizing in that there is no overall team leader who decides which person will do which task or how a problem will be solved. Those are issues that are decided by the team as a whole.And in Scrum, a team is cross functional, meaning everyone is needed to take a feature from idea to implementation.

Within agile development, Scrum teams are supported by two specific roles. The first is a ScrumMaster, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level.

The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

### Extreme Programming

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

First, start off by describing the desired results of the project by having customers define a set of stories. As these stories are being created, the team estimates the size of each story. This size estimate, along with relative benefit as estimated by the customer can provide an indication of relative value which the customer can use to determine priority of the stories. If the team identifies some stories that they are unable to estimate because they don’t

understand all of the technical considerations involved, they can introduce a spike to do some focused research on that particular story or a common aspect of multiple stories. Spikes are short, time-boxed time frames set aside for the purposes of doing research on a particular aspect of the project. Spikes can occur before regular iterations start or alongside ongoing iterations.

Next, the entire team gets together to create a release plan that everyone feels is reasonable. This release plan is a first pass at what stories will be delivered in a particular quarter, or release. The stories delivered should be based on what value they provide and considerations about how various stories support each other.

Then the team launches into a series of weekly cycles. At the beginning of each weekly cycle, the team (including the customer) gets together to decide which stories will be realized during that week. The team then breaks those stories into tasks to be completed within that week. At the end of the week, the team and customer review progress to date and the customer can decide whether the project should continue, or if sufficient value has been delivered.

## Feature Driven Development

The five steps of the FDD project life cycle and core principles of FDD

- **Domain object modeling:** Teams build class diagrams to describe objects in a domain and the relationships between them. This process saves time by helping you uncover what function to add for each feature.
- **Developing by feature:** If a function cannot be implemented within two weeks, it should be broken down into smaller, manageable features.
- **Individual class (code) ownership:** Each class or group of code is assigned to a single owner.
- **Feature teams:** Although one person is responsible for the performance and quality of each class, a feature may involve more than one class, so everyone in the feature team contributes to design and implementation decisions.
- **Inspections:** FDD teams perform inspections to detect defects and ensure the best quality.
- **Configuration management:** This practice involves identifying source code for all features and documenting changes.
- **Regular build schedule:** This best practice will make sure that the team always has an up-to-date system that they can demonstrate for the client.
- **Progress reports:** Project managers should provide frequent progress reports of completed work.

## Lean Software 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 (MVP) 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.

## 2.Discuss the advantages, limitations and challenges in adopting the above software development methodologies

### Advantages of Scrum

- Scrum can help teams complete project deliverables quickly and efficiently
- Scrum ensures effective use of time and money
- Large projects are divided into easily manageable sprints
- Developments are coded and tested during the sprint review
- Works well for fast-moving development projects
- The team gets clear visibility through scrum meetings
- Scrum, being agile, adopts feedback from customers and stakeholders
- Short sprints enable changes based on feedback a lot more easily
- The individual effort of each team member is visible during daily scrum meetings

### Limitations & Challenges of Scrum

- Scrum often leads to scope creep, due to the lack of a definite end-date
- The chances of project failure are high if individuals aren't very committed or cooperative
- Adopting the Scrum framework in large teams is challenging
- The framework can be successful only with experienced team members
- Daily meetings sometimes frustrate team members
- If any team member leaves in the middle of a project, it can have a huge negative impact on the project
- Quality is hard to implement until the team goes through an aggressive testing process

### Advantages of Extreme Programming

- Close contact with the customer
- No unnecessary programming work
- Stable software through continuous testing
- Error avoidance through pair programming

- No overtime, teams work at their own pace
- Changes can be made at short notice
- Code is clear and comprehensible at all times

## Disadvantages of Extreme Programming

- Additional work
- Customer must participate in the process
- Relatively large time investment
- Relatively high costs
- Requires version management
- Requires self-discipline to practice

## Advantages of FDD

- 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. This methodology is very scalable and can grow as your company and the project grows. The five, well-defined steps make it easier for new team members or new hires to come up to speed on the project very quickly.
- Breaks feature sets into smaller chunks and regular iterative releases, which makes it easier to track and fix coding errors, reduces risk, and allows you to make a quick turnaround to meet your client's needs.

## Limitations & Challenges of FDD

- FDD is not ideal on smaller projects and does not work for projects where there is only one developer because it is hard for one or very few people to take on the various roles without help.
- FDD is very scalable from small teams to large cross-functional teams because it is designed to always focus on what the customer needs and wants.

- Places a high dependency on a chief programmer who needs to be able to act as a coordinator, lead designer, and mentor to new team members.

- Provides no written documentation to the client, although there is a lot of documented communication among team members during the project development cycles. Therefore, the client is not able to get a proof for their own software.
- Emphasizes individual code ownership instead of a shared team ownership.
- May not work well with older systems because there is already a system in place and no overall model to define it. You may need to start over and work from the ground up.

## Advantages of Lean Software Development

- The elimination of waste leads to the overall efficiency of the development process. This is turn speeds up the process of software development which reduces project time and cost. This is absolutely vital in today's environment. Anything which allows Organisations to deliver more projects in the same timeframe is going to be popular!
- Delivering the product early is a definite advantage. It means your development team can deliver more functionality in a shorter period of time, hence enabling more projects to be delivered. This will only please both your finance department, but also the end customers.
- Empowerment of the development team helps in developing the decision making ability of the team members which in turn, creates a more motivated team. This benefit really cannot be overstressed enough. Developers hate nothing more than being micro-managed and having decisions forced upon them. This way they can determine how best to develop the functionality which will usually result in a much better end product.

## Disadvantages of Lean Software Development

- 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
- Success in the project depends on how disciplined the team members are and how exceptional are their technical skills. If you don't have a team of individuals with good skills which complement each other, then you have an immediate problem.