**Vikas**

**2017UCO1623**

**EPP**

**Write short notes on following:**

* **Scrum**
* **Lean Development**
* **Extreme programming (XP)**
* **Adaptive Software Development (ASD)**
* **Feature Driven Development**

***Scrum, Lean development, Extreme programming, ASD and Feature driven development all of them are agile software development frameworks.They all follow agile principles i.e. they are iterative and incremental in nature.***

**Scrum**

Scrum defines three different roles to the entire project team which are :

*Developer team* : group of programmers who work in sprints to produce the product.

*Scrum master*: person who ensures that the team works in accordance to the predefined principles of scrum framework.

*Product owner*: Customer is the product owner who acts same as project manager in context of other methodologies and ensures quality of user stories and organizes the features list in backlog refinement sessions.

*Scrum Ceremonies* : These are the essential scrum meetings and comprises Sprint Planning, Daily Stand-up (daily scrum), Sprint Review and Retrospective.

*1. Sprint Planning* is held at the beginning of a new sprint where the Product owner explains the requirements and all of the use cases. The Team asks questions to clear their doubts,effort estimation is done using the planning poker method to assign tasks to individuals.

*2. Daily Scrum or Stand-up* : This meeting takes place each day with the goal to keep everyone on the same page.

*3. Sprint/Iteration Review :*  It usually takes 1-2 hours where the developer team gives demos of work done and receives feedback from the participants of the meeting.

*4. Sprint Retrospective :* Goal of this meeting is to find out any issue that is affecting the progress of the last iteration.

In addition to all these meetings, Scrum also specifies a few artifacts such as the Product Backlog, the Sprint Backlog,Burn-Down Chart and the Product Increment.

The Product Backlog : list of features that are needed as part of the end product

The Sprint Backlog is the items selected from the product backlog for the Sprint in consideration.

Product Increment : It contains items completed during a this Sprint with the increments of previous Sprints

Sprint Burn-Down Chart is used to keep track of the time taken by the Scrum Team.

for a task in a sprint.

**Lean development**

This software development methodology evolved to decrease programming effort and budget with less waste.In this methodology user stories are directly gathered by the programmers so that they directly address the features required by the customer. That is why to complete the project, a team with a mixed skill set is required otherwise you are just aiming for a huge blow in the development process.

It is faster because now programmers do not have to write a bunch of unnecessary lines of codes just for the sake of completing a package.The removal of waste or unnecessary requirements directly increases the efficiency of the development . Short iterations are adopted so that customers can tell new requirements after getting a gist of what is in progress.short iterations means Delivering the product earlier which is a good thing.

The principles were founded by Mary and Tom Poppendieck.This methodology was initially used in companies like toyota.

The key benefits of Lean development include:

1. Learning quickly

2. Measuring real customer behavior, rather than focus groups

3. Becoming closer to your customer

4. Focusing on actionable metrics

5. Becoming comfortable with pivoting

**Extreme Programming (XP)**

It improves software quality and is agile to dynamic customer requirements.The focus of XP is code rather than the design and follows continuous integration and deployment.It comes with a lot of pressure working with tight deadlines. It also includes Pair programming where the team gathers in pairs where they share ideas and work on the same task.

It is different than other traditional models because:

It puts individuals and interactions over processes and tools.

Working software over documentation ideology is considered.

Customer collaboration over contract negotiation gets precedence and

responding to change over following a plan is implemented.

Extreme programming follows the following principles:

* The Planning Game
* Small Releases
* Metaphor
* Simple Design
* Testing
* Refactoring
* Pair Programming
* Collective Ownership
* Continuous Integration
* 40-hour week
* On-site Customer
* Coding Standard

Roles

*Customer :* What features to be included and what is the budget, all driven aspects of a project are prioritized by the customer.

*Developer :*Everyone on the team except the customer is labeled a developer.they are responsible for making the product.

*Coach:* For a new team with no experience in extreme programming , a coach might be helpful who is familiar to this methodology.

*Tracker:* Tracker as the name suggests tracks the relevant metrics and identifies areas for improvement.

In extreme programming, there are daily meetings called stand-ups that last up to 15 minutes.

**Adaptive Software Development (ASD)**

Adaptive Software Development) is an evolved form of Rapid Application Development (RAD). Jim Highsmith published Adaptive Software Development in 2000.

Its main goal is to make a product as quickly as possible with already available templates and architectures that is why it is the perfect choice for on-time and even early delivery products. Because most of the work in this is computer aided , it is easier to change and hence we can call it adaptive software development.It requires a lot of user involvement which might always not be favourable. Because of computer aided softwares it is costly.

Phases of ASD Life Cycle

*1.Speculate* : this is the part where all the planning goes and various complex problems the team might face is brainstormed. That is why short iterations are encouraged ,so that the team can learn from small rather than large mistakes.

*2.Collaborate*: it is required that a large volume of information be gathered and processed to help in solving the problem and it can be accomplished only in a collaborative environment.

*3.Learn*: Team should keep learning through technical reviews and project retrospectives so that they can apply the new information to solve the problems.

Speculate, collaborate and learn is a cycle where one property depends on another two.

**Feature Driven Development (FDD)**

This Software development is centred on features. Features are the same as user stories in Scrum.

Software development life cycle in FDD includes the following steps :

* Develop an overall model
* Build a features list
* Plan by feature
* Design by feature
* Build by feature

This simple five-step process helps in achieving the end product rapidly hence for projects which need to be immediately completed you should go for it. Standards defined for FDD are somewhat easier to follow along as well which acts as a strength for this model.

It is similar to scrum in a lot of manners such as how they gather the user requirements and has meetings where the team is briefed about the tasks that both somewhat matches the scrum methodology.For large projects with top down decision making structure can implement FDD. however it doesn’t work for smaller projects. You can’t take much out of documentation because it is small and not enough for newbies that is why senior developers with experience are required to work around this methodology.