2017UCO1629

Ankit Arya

EPP

Write short notes on following

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

**SCRUM**

SCRUM is an agile framework which is used or developing and delivering complex projects. It is a iterative and incremental framework for managing complex work. SCRUM follow a key principle and it says that there will be some degree of unpredictability as the customers will change their minds about what they want or need , therefore a predictive and planned approach is not suitable. So when the problem cannot be understood or defined upfront , Scrum may be the best choice.

A SCRUM team has 5 to 9 members. The team members break their work into goals such these goals can be completed within timeboxed iterations. The term *sprints* is used for these timeboxed iterations. Usually, *sprints* are limited to one calendar month. During each *sprint* a potentially releasable product is created.

-Product backlog: It is a prioritized features list containing every desired feature or change to the product.

-Sprint planning meeting: A sprint planning meeting is held at the start of each sprint, in which the product owner presents the highest priority items on the product backlog to the team. The team then selects the work they can complete during the coming sprint. That work is then moved from the product backlog to a sprint backlog.

-Daily Sprint: A brief meeting is conducted during each day of the sprint to set the context for each day’s work.

-Sprint review meeting: A sprint review meeting is held at the end of each sprint. During this meeting the team shows what they accomplished during the sprint.

-Sprint retrospective: This meeting is also held at the end of each sprint. During this meeting the team including the scrum master and product owner reflects on how well the Scrum is working for them and what changes they may wish to make for it to work even better.

**Lean Development**

While agile principles teach us to work in short cycles and deliver more value faster and short cycles means more no. Of cycles.On the other hand Lean development principles focuses on limiting the no. of those cycles to prevent work from piling up and which in turn helps us to pull more work into the process.

In lean development, the team releases a bare-minimum version of the product to the market. Learn from users and their feedback and iterate based on the feedback.

The lean development follows 7 lean principles:

**-Eliminate waste**

First we need to understand what is a waste. According to the lean philosophy, anything that does not add any value to the customer is considered a waste.

So one of the key elements of practicing lean development is eliminate the waste.

**-Build quality in**

Every team wants to build quality in to their work. Lean development ensures that this a part of disciplined practice so that the development team follows it striclty. Lean development provides lean development tools to build quality into their work because many teams create waste while ensuring quality into their work.

**-Create knowledge**

Lean development tools like pair programming, code reviews, documentation, knowledge sharing sessions etc. to enure that the team members can learn continously during the entire process.

**-Decide as late as possible**

Some degree of uncertaininty is always present in the field of software development. So it’s always better to delay decision as much as possible until they can be made based on facts and not on uncertain assumptions and predictions.

**-Deliver as fast as possible**

Delivering an end product as fast as possible opens up the opportunity for fast feedback so that based on the feedback the changes can be integrated in the next iteration as fast as possible.

**-Empower the team**

**-Optimize the whole**

The suboptimization of code may sometimes result in a product which does not meet the quality requirements. As the pressure on the developers increases they may release code that is not optimized. So Lean develoment ensure that the product as whole is optimized so the above issue can be avoided.

**Extreme Programming**

In this approach instead of including all the features in the product, we try to deliver the product with the most important feature. So that the customer has something which is working. And in the next iterations, we keep on adding more. Many of the times when we try to include all the feature in the product at once, we might miss the deadline or product could be a failure. May be while adding new features in the product we might have to get rid off all the previous code to add new features. The process of changing your code to add the new functionalities is called *refactoring.*

The five values of XP are:

**-Communication:** The use of appropraite mode of communication is important for the transfer of knowledge from one team member to others.

**-Simplicity:** Keeping the design of the system as simple as possible makes it easier to maintain, support and revise. The purpose of this is to avoid waste and only do the things that are necessary.

**-Feedback:** Constant feedback is very important as it helps in identifying the areas of improvement so that they can work upon it. If the team receives good feedback then it also encourages them.

**-Courage:** You need courage to raise organizational issues that reduce yourteam’s effectiveness. You need courage to stop doing something that doesn’t work and try something else. You need courage to accept and act on feedback, even when it’s difficult to accept.

**-Respect:** Every team member having respect for other members helps in better communication which results in more quality and effective work, and the knowledge can be shared even better.

**Adaptive software development**

Adaptive software development practices are adaptable in an enviroment in which changes are unexpected and unpredictable. It provides the ability to accomodate change in the system.

There are three phases in adaptive development life cycle:

-Speculate

The term *plan* indicates determinism and has a high degree of certainity about the desired result. Following a plan restricts the manager’s ability to steer towards innovative ideas.

In adaptive software development, this term *plan* is replace by the term *sepculate.* While speculating, the team does not abandon planning, but it acknowledges the reality of uncertainty in complex problems. Speculate encourages exploration and experimentation. Iterations with short cycles are encouraged.

-Collaborate

Complex applications requires diverse knowledge as it requires that a large volume of information be collected, analyzed, and applied to the problem. And team collaboration is an important aspect for having diverse knowledge.

Collaborate would require the ability to work jointly to produce results, share knowledge or make decisions.

-Learn

The Learn part of the Lifecycle is vital for the success of the project. Team has to enhance their knowledge constantly, using practices such as −

* Technical Reviews
* Project Retrospectives
* Customer Focus Groups

Reviews should be done after each iteration. Both, the developers and customers examine their assumptions and use the results of each development cycle to learn the direction of the next. The team learns −

* About product changes
* More fundamental changes in underlying assumptions about how the products are being developed

**Feature driven development**

Feature Driven Development (FDD) is an agile framework that, as its name suggests, organizes software development around making progress on features.

A feature is a small, client-valued function expressed in the form . For example, "Calculate the total of a sale", "Validate the password of a user", and "Authorize the sales transaction of a customer".

FDD was designed to follow a five-step development process, built largely around discrete “feature” projects. That project lifecycle looks like this:

1. Develop an overall model   
   outcome - An object model.
2. Build a features list   
   outcome - A list of features grouped into sets and subject areas
3. Plan by feature   
   outcome - A development plan
4. Design by feature  
   outcome – A design package
5. Build by feature   
   outcome – completed client-valued function