

Emerging Programming Paradigms

Assignment For Assessment

Date: 26/07/2020

By: 2017UC01692 (Deepesh Chugh)

1. Write short notes on:

1.1. Scrum

Scrum is an agile process framework created by Jeff Sutherland and Ken Schwaber meant to be used for effective team collaboration on complex product development; It is useful for any team project with aggressive deadlines and complex requirements, and is used in various disciplines especially in software development.

A Scrum team is typically of 5 to 9 members , however scrum can be used by teams of any size ranging from 1 person teams to teams of over 100 members, In a scrum team for software development, the usual roles of programmer, designer, tester etc are discarded; In the place of these roles, the participants are divided into:

- Product Owner
The product owner is a single person that represents the project's key stakeholders, users ,customers or others that the product will affect and is responsible for maximizing the value of the product resulting from the work of the development team.
- Development Team
The development team are the professionals solely responsible for delivering a potentially releasable increment of the product at the end of a sprint, they are self organizing and cross functional.
- Scrum Master
The scrum master is responsible for maximizing productivity of the development team by integrating scrum principles into the process, removing obstacles to the progress of the team, preventing outside influence etc

As a part of the scrum process , some key artifacts are defined to maximize transparency of key information and provide opportunities for inspection and adaptation, these artifacts are:

- Product Backlog
The product backlog is a prioritized list of desired features for the product, the Product Owner is the sole individual held accountable for the product backlog even if they themselves have not created it.

- Sprint Backlog

The sprint backlog is the set of product backlog items selected for the sprint in the sprint planning meeting and a plan for delivering the expected product increment as part of the Sprint goal.

- Increment

An increment is a body of inspectable, "Done" work that represents the sum of all product backlog items completed during a sprint and the value of the increments of all previous sprints.

During the Scrum Process, multiple predefined time-boxed events are used to maintain regularity while minimizing the need for unnecessary meetings, These Scrum Events are:

- Sprint

In Scrum, the project is released as "Done", usable and potentially releasable product increments created in a time-box of one month or less; Such a time box is called a Sprint.

- Sprint Planning

A sprint planning meeting is held at the start of each sprint; In this meeting the scrum team pulls work from the product backlog into the sprint backlog.

- Daily Scrum

A brief meeting is held everyday in the sprint to set the context for each days' work and for keeping the team on track, this is called the Daily Scrum

- Sprint Review

A sprint review meeting is held at the end of a sprint for the development team to informally demonstrate the completed functionality.

- Sprint Retrospective

A sprint retrospective is held at the end of each sprint where the entire scrum team analyzes the efficiency of the Scrum process so far and how they could increase this efficiency.

1.2. Lean Development

Lean development is an iterative agile methodology translated from the manufacturing industry to optimize production and focuses specifically on the removal of waste (i.e. non-value adding activities) from a process, it was first introduced in software development by Dr. Robert Charette in 1993 and was popularized by Mary and Tom Poppendieck in their book of the same name.

There are 7 core lean principles for building more efficient software products , these principles are:

-
1. Eliminate Waste
Anything that does not add value to the product for the customer is considered waste, this can include extra or wrong processes and features.
 2. Build Quality
Ensure that quality is built at the coding level and maintained by often refactoring, this is done to ensure that defects are prevented from the start rather than searched for near the end.
 3. Amplify Learning And Create Knowledge
Expect the application to evolve over time and learn through an iterative process of releasing functioning versions and collecting feedback from users.
 4. Defer Critical Decisions/Commitments
Irreversible decisions should be delayed to the last responsible moment, so that they can be made with the most relevant up-to-date information in mind.
 5. Deliver Fast
Short development cycles are used to develop functioning versions to gather user feedback and respond faster to customers' needs.
 6. Respect People
Respect is given to team members in the context of their opinions being valued without them being micromanaged, this not only leads to reduced frustration but also encourages healthy and productive discussions related to key business decisions.
 7. Optimize The Whole
The project should be viewed as a combination of teams or components working together while considering where to optimize rather than looking at the components or teams individually.

1.3. Extreme Programming (XP)

Extreme programming is an agile software development framework used to improve software quality and responsiveness to customer requirements while also improving the quality of life for the development team; It recommends taking best practices that have been proven to work in the past to an extreme level.

Extreme programming consists of five core values, namely:

1. Communication
XP stresses on the appropriate kind of communication being a face to face discussion with the aid of a drawing mechanism such as a white board and describes software development to inherently be a team sport requiring knowledge transfer among team members.

2. Simplicity

Simplicity implies the removal of waste and to focus on only known requirements during design and development to make it easier to maintain, support and revise as required in the future.

3. Feedback

Paired with simplicity, feedback allows the team to dynamically improve their practices while also adjusting the product according to the customers' needs going forward.

4. Courage

Courage entails team members having a bias for action in the face of fear, whether it be while raising organizational issues, stopping the use of age-old practices if they reduce the teams' effectiveness, or to scrap a hard built feature if there is a better alternative available.

5. Respect

Respect should be maintained between team members to be able to effectively communicate , provide and accept feedback and work together going forward.

1.4. Adaptive Software Development (ASD)

Adaptive Software Development is a software development practice evolved from Rapid Application Development (RAD) practices by Jim Highsmith in 2000, It focuses on providing the ability to accommodate change and adapt to turbulent environments with little planning and learning.

The ASD “life cycle” consists of three phases namely:

1. Speculate

The speculation phase is a non deterministic replacement for the planning phase of a general software development life cycle. During this phase, the project is initiated and planning is conducted while keeping uncertainty of the complex problem at hand in mind. This phase uses initiation information like requirements, user needs etc to define a set of short release cycles.

2. Collaboration

Due to the high rate of information flow that needs to be collected, analyzed and applied to the problem, there is a requirement for the individual members of the team to collaborate i.e. work jointly to produce results, share knowledge and make decisions while utilizing their individual skill sets.

3. Learning

The learning phase is essential for the success of the project and requires the team to increase their knowledge of the project technology using techniques such as:

- a. Focus Groups

-
- b. Technical Reviews
 - c. Project Retrospectives

To effectively use the learn phase, the iterations need to be short so that the mistakes the team is learning from are small rather than large

1.5. Feature Driven Development (FDD)

Feature Driven Development (FDD) is an agile process that organizes software development around the progress made on features , where a feature is a small client-valued function/task similar to user stories in Scrum.

The FDD life cycle consists of five stages:

1. Develop An Overall Model
The domain and development teams alongside a Chief Architect focus more on the overall scope of the product rather than going into feature-heavy detail, i.e the team focuses on capturing the target audience, the context the software should be used in and the necessary structure.
2. Build A Features List
A Team of Chief Programmers decompose the overall model developed in the previous phase into a number of areas and then further into a number of activities (feature sets) where each step within an activity is defined as a feature
3. Plan By Feature
In this stage, important decisions related to feature implementation are taken , Feature sets are assigned to Chief Programmers and Key Classes are assigned to Developers
4. Design By Feature
In this stage, the chief programmer forms a feature team by identifying class owners likely to be involved in the development of selected features , refines the object model based on sequence diagrams and assigns the developers to write class and method prologues. At the end of the phase a design inspection is held.
5. Build By Feature
In this stage the class owners work with the design package produced in the previous phase to implement the items necessary for their class to support the design for the features in the work package, this code is unit tested and inspected according to the Chief Programmer and finally permitted to build.

The FDD framework leads to clearer project management, minimizes complexity and leads to the building of better products rapidly, It is developed while keeping larger teams in mind and will not work as efficiently for smaller projects.