Emerging Programming Paradigms (CED12) Assignment

Emerging Programming Paradigms (CED12)

Sudhanshu Sah
2017UCO1637
Computer Engineering

Contents

Scrum Agile Framework

Definition

Scrum is a lightweight **Agile Framework**, it is an approach to **Project Management**. Scrum helps the progress of a project by dividing the goals in backlog into smaller time limited blocks of work called **sprints**. These sprints are usually time limited to 2 weeks.

Scrum is **both Iterative and Incremental**. Scrum's use of repeating cycles signifies that it is iterative and since the work is delivered in small batches, it is incremental.

Applications

Scrum is usually used by **cross-functional teams with 5-10 members**. Teams using Scrum are **self-organizing,** i.e. they are not managed in a top-down (hierarchical) manner.

Advantages

- Fast Feedback
- Continuous Improvements
- Rapid adaptation to changes
- Faster time to market and accelerated delivery

Disadvantages

- **Strictness**, for e.g. teams are not allowed to host more than one Scrum review in a single sprint.
- **Pressure on team members** due to frequent and short deadlines.

Values

- **Commitment**, team members commit personally to achieve team objectives.
- **Courage**, team members work on complex problems encountered and solve them.
- **Focus**, team members concentrate on the problems identified in sprint planning.
- **Openness**, team members are open about the problems and solutions they encounter.
- **Respect**, team members respect each other and believe in the skills of colleagues.

Roles

- **Product Owner**, team responsible for managing the product backlog
- **Scrum Master**, team responsible for removing distracting elements and making sure that team follows agile principles.
- **Development Team**, team delivering the incremental work in sprints.

Lean Software Development (LSD)

Definition

Lean Software Development is an Agile Framework, it focuses on **optimizing the usage of resources and the development time**. It has **7 principles** which are derived from Lean Manufacturing developed by Toyota (Toyota Production System, TPS).

It aims at providing **Minimum Viable Product (MVP)** and capturing the user feedback regarding the MVP, this feedback is taken into consideration into the next development cycle.

Applications

LSD is usually used when the **project is bloated and the development cost is high**. LSD is used to **cut away the extra features** so that only the MVP is delivered at a much reduced development cost while using the least amount of resources.

LSD is used in projects where **efficiency is paramount.**

LSD is also used in cases where it is uncertain what the users want, LSD can be used to **get crucial user feedback.**

Advantages

- **Streamline the flow,** delivering more features in less time by cutting away the waste.
- **Eliminate extra and redundant activities** to reduce the development cost.
- **Boosts the morale of team** by giving them power to choose which features will be present in the product

Disadvantages

- **Not much scalability** due to heavy dependency on the team involvement.
- **Largely dependent on good documentation**, failure to do so can lead to problems related to collaboration later in the development cycle.
- **LSD allows the SRS to evolve** to maintain flexibility but doing so can lead to development of a project in a direction that was initially not expected.

Principles

- **Eliminate Waste**; remove extra features, unpolished work, etc.
- **Amplify learning**; allow the design to evolve, encourage systematic learning, etc
- **Defer Commitment**; schedule irreversible decisions for the last possible moment.
- **Deliver Fast**; Don't let customers change their mind, early introduction to market leads to better cost advantage.

- **Empower the team**; responsibility based control and planning.
- **Build Quality In**: keep the codebase simple, do good issue tracking, do continuous integration, etc.
- **Optimize the Whole**: modern softwares are more than sums of their parts, integration also plays a major role, one defect can lead to reduced productivity in more than one department.

eXtreme Programming (XP)

Definition

eXtreme Programming is an Agile Framework. It aims to produce **higher quality products** to satisfy stakeholders as well as **make lifes of developers easier.** It includes practices like **coding in pairs**, not implementing features until they are needed, etc. It leverages dynamically changing software requirements.

Applications

XP is usually deployed in teams which are **handling products with highly flexible software requirements.** If the team is not sure about the requirements and expects them to change then XP is the preferred framework.

It is also used in teams where the **number of members is less** and the **members work closely with the managers.**

Advantages

- Save development **cost**.
- **Avoid risks related to dynamic software requirements** in traditional SDLCs.
- Emphasis on **quality of life** of the team members.
- Team members have the **accountability** for the team's work.

Disadvantages

- Little emphasis on code quality.
- More emphasis on code over design.
- Not suitable for teams working remotely, all m**embers need to be co-located.**

Practices

- **Sit Together;** do face-to-face communication.
- **Pair Programming**; collaborating with other members on the same machine while developing code leads to faster development with lesser issues as the code is getting reviewed in real time.
- **Stories**; describe the features users want in short stories with terms relevant to customers and users.
- **Continuous Integration**; code changes are integrated into the larger codebase and tested immediately to make sure the build is working properly.
- **Whole Team**; teams should consist of cross-functional members with necessary roles.
- **Energized Work**; do not overwork, stay healthy,stay focused and show respect to colleagues.

Adaptive Software Development (ASD)

Definition

Adaptive Software Development is an Agile Framework. It was developed after extending **Rapid Application Development (RAD)**. It aims to make the teams **invulnerable to penalties due to rapidly changing software requirements** and the market. In ASD, the **product is continuously evolved by lightweight planning and continuous learning.** There are three phases involved in the ASD process: **speculate, collaborate and learn.**

Applications

- Teams who want to prioritize **rapid development and quick delivery of the product.**
- Teams who want their product to be **continuously evolving.**

Advantages

- More **emphasis on end user experience.**
- **Quick delivery of product.**
- **Emphasis on transparency** between users and developers.

Disadvantages

- **Demands high user feedback** and involvement.
- Incurred **cost due to extensive testing.**

Phases

- **Speculate**

  Teams are free to do **experimentation and exploration**. This phase extends the planning part by incorporating the results from the experimentation and exploration.

- **Collaborate**

  Collaborate phase focuses on combining the knowledge gained by the individual members to solve a larger and more complex problem. Complex solutions are not built in a day, they evolve over time, this evolution is made possible by individual efforts of the members, their individual **efforts are collected, analyzed and applied to the problem.**

- **Learn**

  Teams need to evolve their knowledge continuously by doing reviews of mistakes.

Feature Driven Development (FDD)

Definition

Feature Driven Development is an Agile Framework. It **focuses on making progress on features.** These features are not necessarily product features but are more like stories in Scrum.

Applications

- Teams working on **large projects.**
- Teams which are organized in **top-down management.**

Advantages

- **Rapid development** due to simple 5 step process (Explained in Phases part).
- **Allows larger teams** to collaborate easily.
- Requires **fewer meetings.**

Disadvantages

- **Not efficient** for smaller projects.
- Involves **less documentation**; could be a problem for future development.
- Puts more **dependency on project managers** and lead developers.

Phases

- **Develop overall model**; Define outline of domain model.
- **Build feature list**; Identify features required by the client, features are not necessarily product features, they are more like scrum stories.
- **Plan by feature**; Give ranking to the features in feature list to determine the order in which the features will be developed.
- **Design by feature**; lead programmer decides which features will be implemented in a time limited (2 weeks) development iteration. A design review is done later on to finalize the design.
- **Build by feature**; All supporting code is implemented to test the feature and then unit testing is done.