Emerging Programming Paradigms

Assignment-Short Notes

By: Rakesh Kumar

2017UCO1625

COE-2

S.No.	Content
1.	Scrum
2.	Lean Development
3.	Extreme Programming (XP)
4.	Adaptive Software Development (ASD)
5.	Feature Driven Development

1. Scrum:

Scrum is an iterative and incremental approach for projects or application development. It is structured in cycles of work called **Sprints**, which generally lasts less than a month. Requirements are prioritized by scrum team and commits to complete the selected items in the sprint. During the Sprint, the chosen items do not change. Whole scrum team gathers every day for analysis of progress. On sprint end, the team reviews the Sprint with stakeholders, and give their feedback. Feedback items are completed in the next sprint.

Roles in Scrum:

Roles in Scrum Team are:
1. The Product Owner tells the team on which items to work on. Product owner is responsible for the *Return of Investment* for the product.
2. The Team It is cross-functional. It is a self-organizing team which has the people who have the potential to develop the product. They have a high degree of autonomy and accountability.
3. The ScrumMaster The ScrumMaster helps the product group learn and apply Scrum to achieve business value

Scrum Ceremonies:

1. Sprint Planning: It is done at the start of a sprint, in which requirements and timeline is decided. It also has a doubt session.
2. Daily Scrum: It is a short meeting to keep everyone updated about the work-items.
3. Product Backlog: 5-10% of each scrum is dedicated to the refinement of the product backlog. In this, remaining work item are done which can be feedback items also.
4. Sprint Review: This meeting is done to retrospect sprint and check scope for improvement.

2. Lean Development:

Lean Software Development is an iterative agile methodology which was developed by Mary and Tom Poppendieckaken. It is derived from Lean manufacturing; whose goal is to have a simplified and streamlined approach. Lean software development methodology is used to reduce the cost of the project by reducing '*Waste Items*'. It is widely used in IT Industry to reduce programming time. Lean development is based on following principles:

1. Eliminating Waste
2. Building Quality In
3. Amplifying Knowledge
4. Delaying Commitment
5. Delivering Fast
6. Respecting people
7. Optimizing the whole thing

Benefits of Lean Development Methodology:

1. It is a faster way to develop a product, with less budget and other resources. Efficiency is improved by a considerable margin. It allows to teams to have a more stream lined approach.
2. Scope of learning: Learning is increased with code reviewing. Lean development uses pair programming with which skillset of developers is enhanced.
3. Fast Delivery: Lean Development focuses on small but fast releases which allows developers to study the market/custom behavior. This allows a product to be developed incrementally along with continuous feedback.
4. Lean also tells to develop unit test cases alongside the code development. Which means testing can be done alongside the development. Thus, reducing the time spent on testing the products.

Pitfalls of using Lean Development Methodology:

1. Implementation is not as it easy as it seems. Leader has to decide the goals which team can understand and do within timeframe. It requires complete knowledge of the organization.

3. Extreme Programming (XP):

Extreme programming is developed by Kent Beck. Extreme programming is a structured approach toward developing software quickly as well as maintaining quality. It has high customer involvement, multiple feedback loops and continuous planning. In XP, a working incremental software update is given every 1-3 weeks. It also includes pair programming which allows the skills to flow across the developers.

Principles of Extreme Programming are:

❖ Small Releases
❖ Simple Design
❖ Testing
❖ Refactoring
❖ Metaphor
❖ Pair Programming
❖ Collective Code Ownership
❖ Continuous Integration
❖ Weeks with 40 hours
❖ On-site Customer
❖ Maintained standard for code

Benefits of extreme programming:

1. Focus on continuous customer involvement increase transparency, reduces time for development and allows to fix issues immediately.
2. Reduces time to incur changes with active customer involvement and running pre-determined unit tests alongside development.
3. Reduced misunderstanding between customer and development team.
4. Changes can be accommodated anytime during the development.

Roles in Extreme programming:

1. Customer: Owner of the product, has set of requirements and budget.
2. Developers: Team which develops the product. Everyone but customer is developer.
3. Coach: Leads the team about the increment in the product.
4. Tracker: Tracks the standards, timeline and scope for improvement.

4. Adaptive Software Development (ASD):

Adaptive Software Development is a cyclic software development model. It is an upgraded version on Rapid Application Development (RAD). ASD means to develop a new software by adapting to the available projects which are used as a template, which leads to a faster development cycle.

Phases of Adaptive Software Development:
1. <u>Speculation:</u> This is the planning phase of the development. Complete project path from requirements, user needs, timeline to release cycles are decided in this phase. Short iterations are encouraged in planning to avoid scaled mistakes.
2. <u>Collaboration:</u> Evolved applications require analyzing large volumes of information which can be applied to solve a problem. This requires wide knowledge requirements that can be achieved only with team collaboration. Collaborate enables teams to wok jointly.
3. <u>Learn:</u> Reviews are done after each iteration, which helps teams to enhance knowledge. It also helps in deciding next work items among the teams.

Benefits of Adaptive Software Development:

1. Focuses on end users, which helps in better products.
2. Timely delivery, scope for early delivery also.
3. Transparency between customer and developers.

Pitfalls of Adaptive Software Development:

1. Unit test integration at every stage which increases the cost for development.
2. Continuous feedback from customer, faster development and scope for rapid iterative may lead to slope creep.
3. Requires user involvement which is not always available.

5. Feature Driven Development:

Feature Driven Developments is a stakeholder focused development strategy. A feature is a client valued functions, it is similar to user stories in **Scrum**. It is five step development process and follows standards which are easier to keep. Unlike Scrum, it uses backlog items or user features to develop one by one. Deciding features helps in faster development times.

Processes in Feature Driven Development:

1. Develop overall model.
2. Create a list of features to include.
3. Plan features one by one.
4. Design features one by one.
5. Build features one by one.

Benefits of Feature Driven Development:

1. Feature Hierarchy helps in prioritizing which features need to be developed first.
2. Feature list decomposes the user requirements which helps in avoiding slope creep and deciding time frame for development correctly.
3. Feature Driven Development uses Unified Modeling Language (UML), which helps in expressing feature requirements. And reduces miscommunication among large teams.