Emerging Programming Paradigms

SUBMITTED TO: **Kanika Mam**

SUBMITTED BY: **Rahul Bansal**
2017UCO1661

Write short notes on following

1.Lean Development

Lean Software Development (LSD) is an agile framework based on optimizing development time and resources, eliminating waste, and ultimately delivering only what the product needs. The Lean approach is also often referred to as the Minimum Viable Product (MVP) strategy, in which a team releases a bare-minimum version of its product to the market, learns from users what they like, don't like and want to be added, and then iterates based on this feedback.Lean development methodology principles can be applied in any IT environment for improved programming practices. The practice is based on seven principles:

- Waste Elimination
- Amplifying Learning.
- Late Decision Making
- Fast Delivery
- Team Empowerment
- Built-in Integrity
- View Applications as a Whole

Strengths:- Delivered in less time, eliminate unnecessary activity thus reduce costs ,boosts morale of team.
Weakness:- Not scalable as too dependent on team work, depend on strong documentation.

2. Scrum

Scrum is a process framework used to manage product development and other knowledge work. Scrum is empirical in that it provides a means for teams to establish a hypothesis of how they think something works, try it out, reflect on the experience, and make the appropriate adjustments. That is, when the framework is used properly.Scrum is

structured in a way that allows teams to incorporate practices from other frameworks where they make sense for the team's context.

Applicablity: Scrum is best suited in the case where a cross functional team is working in a product development setting where there is a non trivial amount of work that lends itself to being split into more than one 2 – 4 week iteration.

Values:-

Commitment, Courage, Focus, Openness, Respect

Principles:-

Transparency, Inspection, Adaptation

Roles:-

The Product Owner
The Scrum Master
The Development Team

Events:-

Sprint: The Sprint is a timebox of one month or less during which the team produces a potentially shippable product Increment.

Sprint Planning:A team starts out a Sprint with a discussion to determine which items from the product backlog they will work on during the Sprint. The end result of Sprint Planning is the Sprint Backlog.

Daily Scrum: The Daily Scrum is a short (usually limited to 15 minutes) discussion where the team coordinates their activities for the following day. The Daily Scrum is not intended to be a status reporting meeting or a problem solving discussion.

Sprint Review: At the end of the Sprint, the entire team (including product owner) reviews the results of the sprint with stakeholders of the product.

Sprint Retrospective: At the end of the Sprint following the sprint review the team (including product owner) should reflect upon how things went during the previous sprint and identify adjustments they could make going forward.

3. Extreme programming (XP)

XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software.E**x**treme **P**rogramming (XP) was conceived and developed to address the specific needs of software development by small teams in the face of vague and changing requirements.Extreme Programming is one of the Agile software development methodologies. It provides values and principles to guide the team behavior. The team is expected to self-organize. Extreme Programming provides specific core practices where −

- Each practice is simple and self-complete.
- Combination of practices produces more complex and emergent behavior.

Applicablity:

- Dynamically changing software requirements
- Risks caused by fixed time projects using new technology
- Small, co-located extended development team
- The technology you are using allows for automated unit and functional tests

Extreme Programming involves −

- Writing unit tests before programming and keeping all of the tests running at all times. The unit tests are automated and eliminates defects early, thus reducing the costs.
- Starting with a simple design just enough to code the features at hand and redesigning when required.
- Programming in pairs (called pair programming), with two programmers at one screen, taking turns to use the keyboard. While one of them is at the keyboard, the other constantly reviews and provides inputs.
- Integrating and testing the whole system several times a day.
- Putting a minimal working system into the production quickly and upgrading it whenever required.
- Keeping the customer involved all the time and obtaining constant feedback.

Advantages

Slipped schedules, Cancelled projects, Costs incurred in changes

VALUES OF XP

Communication, Simplicity, Feedback, Courage, Respect.

4. Adaptive Software Development (ASD)

Adaptive Software Development has evolved from RAD practices. The team aspects also were added to these practices. Companies from New Zealand to Canada, for a wide range of project and product types, have used adaptive Software Development.

Adaptive Software Development is cyclical like the Evolutionary model, with the phase names reflecting the unpredictability in the complex systems. The phases in the Adaptive development life cycle are −

1.Speculate: The term plan is too deterministic and indicates a reasonably high degree of certainty about the desired result. The implicit and explicit goal of conformance to plan, restricts the manager's ability to steer the project in innovative directions.

2.Collaborate: Complex applications are not built, they evolve. Complex applications require that a large volume of information be collected, analyzed, and applied to the problem. Turbulent environments have high rates of information flow. Hence, complex applications require that a large volume of information be collected, analyzed, and applied to the problem.

3.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.

Strengths: Focused on the end users, allows for on-time and even early delivery, encourages more transparency between developers and clients

Weaknesses: Demands extensive user involvement, integrates testing into every stage, emphasis on rapid iterating and continuous feedback can lead to scope creep.

Characteristics of ASD:-

- Mission Driven
- Component Based
- Iterative
- Time Boxed
- Change Tolerant
- Risk Driven

5. Feature Driven Development

Feature Driven Development (FDD) is an agile framework that, as its name suggests, organizes software development around making progress on features. Features in the FDD context, though, are not necessarily product features in the commonly understood sense. They are, rather, more akin to user stories in Scrum. In other words, “complete the login process” might be considered a feature in the Feature Driven Development (FDD) methodology.

FDD was designed to follow a five-step development process, built largely around discrete “feature” projects:-

Developing an Overall Model : In the first stage, the development team members cooperate together to build an object model of the domain problem. The main goal is to propose a model for the domain area. The Chief Architect follows them and provides guidance.

Building a Feature List: After the development team built an object model, then it is time to identify the features that the user or client values. These features are meant to be the building barriers of the project that help the group members to navigate the processes.

Planning by the Feature: The third stage turns around managing the features and the way the development team tends to implement them. As anticipated, it’s essential to consider the team workload, risks, as well as other important aspects in order to prevent any kind of complex issues from arising.

Designing by the Feature: Everything planned pretends design. Using the knowledge from the first modeling process, the chief programmer selects all the features that the team should develop next and also identifies the domain classes.

Building by the Feature: The last step is to put all the necessary items into action in order to support the design. In other words, once your team developed, tested and inspected the code, it is time to start developing the software.

Strengths: Simple five-step process, allows larger teams to move products forward,leverages pre-defined development standards.

Weaknesses: Does not work efficiently for smaller projects, less written documentation, highly dependent on lead developers or programmers.