Emerging Programming Paradigms

**SCRUM**

Scrum is an *agile framework* which is used when the project is big and complex, the amount of work cannot be estimated and the requirements change heavily due to stakeholders feedback.  
Scrum team contains about 5-10 people which includes product owner, scrum master and the development team.

**Product owner** - He acts as a bridge between the stakeholders and the rest of the scrum team. He explains all the requirements and the clarify the queries related to them based upon the stakeholders need.

**Scrum master** - He manages the development team and removes all the outside challenges and dependencies to make sure team can work efficiently.

**Development team** - They are the actual people who creates the product. They are not separated by some specific role like tester, designer, programmer, etc. All of them work together to complete the tasks at hand.

First the whole project is divided into *user stories* which are then listed out based on priority upon discussion with product owner. At the start of each sprint the development team decides what all tasks they can complete in one sprint (*Sprint* is short fixed duration of about a month in which a some workable product is created). Each day there is daily scrum which is a short meeting to set the goals for each day and remove any hurdle if any. At the end of sprint there is a *sprint review meeting* to show what they have achieved/completed during this sprint.

**LEAN DEVELOPMENT**

Lean development is an *agile framework* which is actually adapted from the manufacturing industry which used the lean development process as a way to optimize production and assembly lines to minimize waste and maximize customer value.  
Lean development empowers the development team to make decisions. The team first delivers the bare minimum version of its product to the market. From the customers, users, stakeholders feedback they understand *what is needed in the product and what is not*. Based upon this feedback they reiterate. This makes the production more efficient and the production time is reduced, hence reducing the cost.  
This is an excellent approach as it removes all the features and related activities which will not be needed in final product. But it makes the project heavily rely on the development team, so the organisation must trust the team for this approach to work.

**EXTREME PROGRAMMING (XP)**

Extreme programming is an *agile framework* that aims to produce a higher quality software and a higher quality of life for the development team. It is used when the project has ever changing requirements, risks involved due to limited time or new technology.

The five values of XPare :   
**Communication -**Face to face is considered to be the best way of communication. The team can use aid of white board or other drawing mechanism.   
**Simplicity -** The design of the system must be kept as simple as possible addressing all the requirements and not covering for future extensions.   
**Feedback -** The experienced developers should use their previous feedbacks while solving problems and creating the design.   
**Courage -** Every team member should have courage to raise any organizational issues, etc. which is hindering their individual or team performance.   
**Respect -** Each team member should respect each other in order to communicate to create a better working environment and work more efficiently.

The core of XP is various practices which helps to maintain the above five values. Some of these practices are - sit together, energized work, peer programming, test first programming, continuous integration, etc.

**ADAPTIVE SOFTWARE DEVELOPMENT (ASD)**

Adaptive software development is evolved from RAD ( Rapid Application Development ) practices. It was published by John Highsmith in 2000. Adaptive Software Development practices provide ability to accommodate changes and are adaptable in turbulent environments with products evolving with little planning and learning. It is a cyclical and involves three phases - Speculate, Collaborate, Learn.

**Speculate** - In this phase project is initiated and planning is conducted. The team while speculating plans the project goals while acknowledging the uncertainty in complex problems.

**Collaborate** - To create complex applications they is need for large amount of information to be collected, analyzed and then applied to the problem. In order to do that there is a need for proper communication between team members. The team should work jointly to make decisions, share knowledge and experience.

**Learn** - This part is also very important in the lifecycle of software development. The team need to enhance their knowledge to keep up with the team and provide solutions to complex and new problems. Team can use practices such as - Technical reviews, focus groups, project postmortem.

All the three phases are important and necessary for the success of the project.

**FEATURE DRIVEN DEVELOPMENT**

Feature driven development is an agile framework where the progress of the project is measured based upon the features that are completed. The features here are not really the product features but user stories as in SCRUM.   
FDD is used when working for a large corporation on a big scale project. It heavily relies on chief or lead developers as opposed to collective decision making in other agile frameworks like SRUM, XP, etc.

FDD has a five step development process -

1. Develop an overall model
2. Build a feature list
3. Plan by feature
4. Design by feature
5. Build by feature

The first real-world application of the Feature Driven Development methodology was on a 50-person software-development project for a Singapore-based financial institution. The simple 5 step process allows for rapid development but it is not efficient for smaller projects. Pre - defined development standards helps in completing tasks quickly.