## Write short notes on following

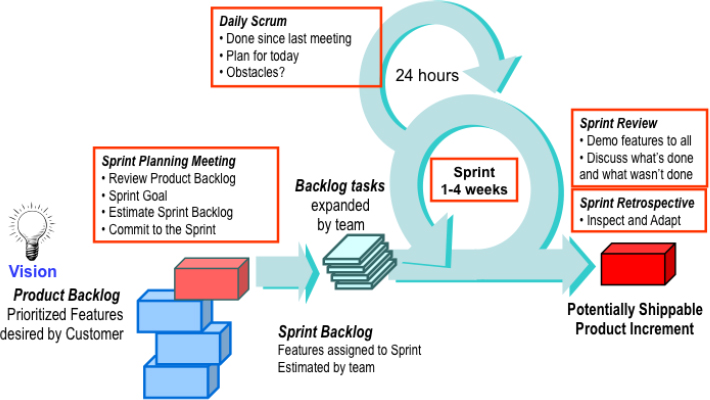
1. **Scrum**

Scrum is an agile methodology designed for teams to guide them in the iterative and incremental delivery of a product. It is often referred to as “an agile project management framework” and its focus is on the use of an empirical process that allows teams to respond rapidly, efficiently, and effectively to change.

Unlike traditional, sequential practices to product development which are idealistic and rely on a rigid algorithmic approach, scrum adopts a heuristic approach which acknowledges human capabilities and self-organization to deal with unpredictability while finding non-trivial solutions to complex problems. This self-organization is often achieved by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines involved. Traditional project management methods fix requirements in an effort to control time and cost; Scrum on the other hand, fixes time and cost in an effort to control requirements. This is done using time boxes, collaborative ceremonies, a prioritized product backlog, and frequent feedback cycles.

A typical scrum project begins with a clear vision provided by the business, and a set of product features in order of importance. A time box commonly referred to as an *iteration* or *sprint*, is the set amount of time that the team has to complete the features selected. Sprints are generally from one to four weeks in length, and that length is maintained throughout the life of the project. After this time in the sprint, the team is protected from interruptions and allowed to focus on meeting the sprint goal. No changes to the sprint backlog are allowed; however, the product backlog can be changed in preparation for the next sprint.

During the sprint, the team checks in daily with each other in the form of a 15-minute meeting known as a scrum. The team stands in a circle and each member states what they did yesterday, what they plan to do today, and what is getting in their way. At the end of the sprint, the teams demonstrate the work they have completed to the stakeholders and gather feedback that will affect what they work on in the next sprint. They also hold a retrospective meeting to learn how to improve.



Some limitations of Scrum are:

* Scrum fails to deliver the same results when the team members are geographically dispersed or part-time.
* Scrum relies on heavy collaboration of the team members which often requires the understanding of skills in other teams. Therefore, it fails when the team members have very specialized skills.
* In Scrum, dividing product development into short sprints requires careful planning. External dependencies, such as user acceptance testing or coordination with other teams, can lead to delays and the failure of individual sprints.
* In Scrum, product increments should be fully developed and tested in a single sprint. Products that need large amounts of regression testing or safety testing (e.g., medical devices or vehicle control) for each release are less suited to short sprints than to longer waterfall releases.

1. **Lean Development**

Lean development refers to the application of lean manufacturing principles and practices to software development. It is based on optimizing development time and resources, eliminating waste, and ultimately delivering only what the product needs. The lean approach is also called the Minimum Viable Product (MVP) strategy, in which a team releases a bare-minimum version of its product to the market, takes feedback from users, and then iterates based on this feedback.

Principles used in Lean Development

* **Waste elimination:** Waste is anything in the software development lifecycle that doesn’t add value to the customer and must be eliminated. E.g. Partially done work, relearning, handoffs etc.
* **Amplification of learning:** This is achieved through the usage of short sprint cycles along with refactoring and integration testing after each cycle. This helps in both the client getting to know about the progress of the project and the teams getting feedback and learning more about the requirements.
* **Delaying of decisions:** As software development is always associated with some uncertainty, better results should be achieved with a set-based or options-based approach, delaying decisions as much as possible until they can be made based on facts and not on uncertain assumptions and predictions.
* **Fast delivery:** Customers value rapid delivery of a quality product. The sooner the end product is delivered without major defects, the sooner feedback can be received, and incorporated into the next iteration.The shorter the sprints, the better the learning and communication within the team.
* **Empower the team:** The people involved with the product must be empowered in the sense that the managers treat them with respect and acknowledge their work and not treat them as resources. The developers should be given access to the customer, the team leader should provide support and help in difficult situations, as well as ensure that skepticism does not ruin the team’s spirit.
* **Build integrity in:** There are two types of integrity in lean development.
  + Perceived integrity: The customer needs to have an overall experience of the system including its advertisement, delivery, deployment, access, intuitiveness, price and its closeness to the use case.
  + Conceptual integrity: The different components of the system work well together with balance between flexibility, maintainability, efficiency and responsiveness.
* **Optimize the whole:** The focus of the product teams should be on the optimal working of the final product and not on the short-term profit optimization of their own part for a system with smoothly interacting components**.**

Advantages of Lean Development

* Streamlined approach allows more functionality to be delivered in less time.
* Eliminates unnecessary activity, and as a result can reduce costs.
* Empowers the development team to make decisions, which can also boost morale.

Disadvantages of Lean Development

* Heavily depends on the various teams involved, making it not as scalable as other frameworks.
* Depends on strong documentation, and failure to do so can result in development mistakes.

1. **Extreme programming (XP)**

Extreme Programming is a software development methodology designed to improve the quality of software and its ability to properly adapt to the changing needs of the customer or client. The extreme programming model recommends taking the best practices that have worked well in the past in program development projects to extreme levels. XP advocates frequent product releases in short sprint cycles which improves productivity and introduces checkpoints at which new customer requirements can be adopted.

XP acknowledges that changes are an inevitable and desirable aspect of software development projects, and should be planned for, rather than laying out rigid requirements or rules.

Extreme programming has been described as having the following ***practices*** :-

* 1. Pair Programming: Involves two developers sharing a workstation. This facilitates a quick code review and better solutions to problems faced while development.
  2. Planning Game: Planning game is a meeting that occurs once per sprint, typically weekly. Its purpose is to guide the product into delivery. It’s divided into :-
     1. Release Planning: Determines the requirements included in the release and its delivery. Customer reps are included in this meeting.
     2. Spring Planning: Plans the activities and tasks of developers.
  3. Test-driven Development: Software development process that relies on the repetition of a very short SDLC where requirements are converted to specific test cases, and code is improved to pass those cases.
  4. Continuous Integration: Practice of merging all developers code to a shared branch several times a day.
  5. Refactoring: Restructuring existing code without affecting the external behaviour.
  6. Small Releases: Frequent releases of live functionality of the existing code.
  7. Coding Standard: Set of rules that the teams agree upon and adhere to during the entire SDLC.
  8. Collective code ownership: The teams take collective control and responsibility for the code. This means that each team member has edit access to the full code.
  9. Simple Design: Code is refactored to make complex code simpler and more understandable.
  10. System metaphor: Naming convention for classes and methods for making its functionality obvious to the entire team.
  11. Sustainable pace: A practice that sets a limit to the amount of effort that a developer puts in during a short period of time. (40 hour week)

1. **Adaptive Software Development (ASD)**

Adaptive software development is a software development process that replaces the traditional waterfall model with a recurring series of speculate, collaborate, and learn cycles. This facilitates continuous learning and adaptation to the changes in the project due to requirements or market needs.

Phases in ASD :-

1. **Speculation**: The project is initiated and the planning of requirements and impact analysis is conducted. The project plan uses project initiation information like project requirements, user needs, customer mission statement etc, to define a set of release cycles that the project wants.
2. **Collaboration**: Collaboration encourages the team members to communicate and work as a team but it emphasizes individualism as individual creative thinking plays a pivotal role in creative thinking. Complex applications require that a large volume of information be collected, analyzed, and applied to the problem. This results in diverse knowledge requirements that can only be handled by team collaboration.
3. **Learning**: The team members might underestimate the amount of knowledge required about the technology and the domain which may not lead to the desired result. Learning is achieved through practices such as technical reviews, project retrospectives and customer focus groups. Thorough reviews are done after each sprint in which both the developers and the customers examine their assumptions and use the direction of each development cycle to learn the direction of the next.

Advantages of ASD:

* Focused on the end users, which can lead to better and more intuitive products.
* Allows for on-time and even early delivery.
* Encourages more transparency between developers and clients.

Disadvantages of ASD:

* Demands extensive user involvement, which can be difficult to facilitate.
* Integrates testing into every stage, which can add to a project’s costs.
* Emphasis on rapid iterating and continuous feedback can lead to scope creep.

1. **Feature Driven Development**

Feature driven development is an Agile framework and an iterative and incremental software development process that organizes software development around making progress on features. These features don’t exclusively include the features in the final product but includes the features mentioned in the user stories as well. The primary objective of FDD is to deliver tangible, working software repeatedly in a timely manner while adhering to the Principles behind the Agile manifesto.

Activities/phases in project development with FDD:

* 1. **Develop overall model**

The project starts with a high level walkthrough of the scope of the system and its context. Then, detailed domain models are created for each modelling area by small groups and presented for peer review. One or more of the proposed models are selected to become the model for each domain area. Domain area models are progressively merged into an overall model.

* 1. **Build feature list**

Knowledge gathered during the initial modeling is used to identify a list of features by functionally decomposing the domain into subject areas. Subject areas each contain business activities, and the steps within each business activity form the basis for a categorized feature list. Features should not take more than two weeks to complete, else they should be broken down into smaller pieces.

* 1. **Plan by feature**

After the feature list is completed, the next step is to produce the development plan and assign ownership of features (or feature sets) as classes to programmers.

* 1. **Design by feature**

A design package is produced for each feature. A chief programmer selects a small group of features that are to be developed within two weeks. Together with the corresponding class owners, the chief programmer works out detailed sequence diagrams for each feature and refines the overall model. Next, the class and method prologues are written and finally a design inspection is held.

* 1. **Build by feature**

After a successful design inspection for each activity to produce a feature is planned, the class owners develop code for their classes. After unit testing and successful code inspection, the completed feature is promoted to the main build.

Advantages of FDD:

* Simple five-step process allows for more rapid development
* Allows larger teams to move products forward with continuous success
* Leverages pre-defined development standards, so teams are able to move quickly

Disadvantages of FDD:

* Does not work efficiently for smaller projects
* Less written documentation, which can lead to confusion
* Highly dependent on lead developers or programmers