Chapter 3: Introduction to Agile methodology

Agile methodology is an approach to the project management which helps to respond to the unpredictability of building software through incremental, iterative work cadences, known as sprints. This methodology was developed to deal with situation where the waterfall model fails. The biggest drawback of waterfall model is that it assumes that every requirement of the project can be identified before any design or coding occurs. This may always be applicable for the development of an automobile on an assembly line, in which each piece is added in sequential phases. However it may or may not be applicable for software development. For example, for a BAU (Business as Usual) projects where the software is already in use for a long time, waterfall model is the best method to implement any changes requests because amount of uncertainly is very less compared to developing a new product. However for developing brand new software, waterfall model is not an ideal choice as the amount of uncertainty in terms of requirement and user quality expectation. It might be the case that end product is not exactly what user has expected due to mismatch of requirement understanding between user and developer. It might also be the case that a team might have built the software it was asked to build, but, in the time it took to create, business realities have changed so dramatically that the product is irrelevant. In that scenario, a company has spent time and money to create software that no one wants.

Agile development methodology provides the opportunity to assess the direction of a project throughout the development lifecycle. It does it through an iterative cycle to build and test followed by an assessment by the user/business until they are satisfied with the product. Thus by focusing on the repetition of abbreviated work cycles as well as the functional product they yield; agile methodology could be described as iterative and incremental.

Agile methods should not be confused with the Spiral method in any case. Spiral method force you to plan for all the iteration in the beginning where as Agile gives you the flexibility to plan only which you surely know and leaves the rest of planning for next iteration. In Spiral methodology since you have planned for all the iteration in the beginning, the number of iterations are fixed. But in Agile, you can have as much iteration required producing the final product as the planning is dynamic.
Main Stages in Agile project:

- **Define initial requirement:** Business will come up with a high level requirements to the project. It is not necessary to have all the requirement finalized and clear in this stage but business have a good idea about the facilities or product they want. For example business want a resource management product but may not be clear whether they want a desktop application or web based one.

- **Prepare high level use cases:** Based on the initial business requirement, project business analyst will have further discussion with client to have a deeper understanding to come up with high level use cases. These use cases will be discussed with the clients to finalize the requirement at a higher level.

- **Prepare High Level Plan:** Once the high level business cases are approved and finalized, Project Manager will prepare a high level plan. Normally the initial plan has a short planning horizon and mainly covers the plan till the first prototype completes.

- **Begin Iteration:** This phase is where all the teams involved in the project start their planning and designing phase. For example, development team may produce Mid Level Platform Design and Test team may produce Master Test Plan.

- **Build and Unit Test:** Development test build and unit test the prototype in this stage. If it is not the first iteration then they do the enhancement to the prototype of the last iteration as per the refines use case.

- **System Test:** Testing team performs the testing of the prototype as per their test cases and test plan prepared in previous stages. If it is not the first prototype, the test team refines their test case and test plan according to the refined use cases and perform testing. Regression testing may also be required to prove that functionality tested in earlier iterations are still working properly.

- **User Evaluation:** Once the prototype passes the system test, representatives from the client evaluate the prototype against the requirements and the use cases. They also access if they want more functionality to be added to the product as the initial requirement gathering stage may not have collected the entire user requirements or were not clear at that time.

- **Are more changes required?** This is not a stage but a decision point in the project life cycle where client decides whether they need more changes to the products or not based on the evaluation done in the last stage. Prototype may go into one more iteration if client want more changes to the product. If they are happy with the product and are ready to accept this as a final product, then the product may go the final round of system test.
• **Refine Requirement /Use case:** If client decides to make propose changes after the User Evaluation stage, requirements and in turn user cases need to be refined as per the change request. Development and testing team will use these changes use cases to refine their design and test plan.

• **Final System Test:** Once the customer approves the prototype as the final product, a final system integration test might be required to check the integration of the product with client environment and other application. User also might want to do a final assessment of the product with all the other application integrated to it.

• **Product Release:** Once final test is done, product is ready to be released and deployed at the client location. Training and demo might be required for the wider user group along with the user manuals.

**Advantage and Disadvantage of Agile Methodology:**

**Advantages:**

• **No Detail requirement needed:** You don’t need to have the entire requirements finalized to start the development work. Build and Test can start as soon as initial high level requirements are available.

• **Early benefit to the user/business:** Following Agile methodology gives an early view to the user about how the final product might look and behave. This helps them into finalizing the user requirements. It might also happen that the prototype delivers some of the requirements which use may want to use and evaluate till the final product is delivered with all the functionalities.

• **Face to face communication:** Agile method give more emphasis on having the face to face communication between the user/customer and project team to make sure there is no room left for any kind of confusion in understanding requirement and inputs.

• **Less time to market:** Using Agile method, final product is delivered to the customer in least possible time.

• **Less cost to customer:** It saves cost for both customer and supplier as resources are used for less time.

• **High Quality:** Since customer is involved in all the stages of software development, means the quality of final quality if high resulting in highly satisfied customer.
Disadvantages:

- **Smaller Planning Horizon:** Since Agile projects have smaller planning horizon meaning that project is started without detailed planning, there is always a chance that initial project effort and cost estimation may not be correct. This might result in multiple changes to estimation in subsequent estimation.

- **Lesser design and documentation:** Since build and test starts early, there is always a chance that proper designing and documentation may take a back seat.

- **Need clear customer vision:** As customer input is required in all iterations, it is very necessary that customer should have clear vision of end product. Project can easily lose its direction if client have only vague idea of the product they want.

- **Necessity of experienced and senior resources:** Since Agile method is more about less planning and more decision making, it is absolutely necessary to have experienced and senior resources in the team or experienced resources mentoring new resources.
Various Agile Software development methodologies

Some well known Agile Software Development methods are:

1. **eXtreme Programming:**

   eXtreme Programming works on the principle of Simplicity, Communication, Feedback and Courage. The application is designed such that it is adaptive to changing requirement and is simple. Lot of client interaction and communication in team are key factors which help in development of adaptive application.

2. **Scrum:**

   In Scrum, a list of all the deliverables for the project, named backlog, is identified and is continuously updated. Project is divided into small units named sprint, lasting 2-4 weeks generally. Deliverable, which is visible and usable increment, is decided for each sprint. Active communication in the team and follow-up using the backlog help and ease the development.

3. **Agile Modeling:**

   Agile Modeling is used for modeling and documentation of software systems. It involves the software values, principle and practices which lead to software development in more flexible manner.

4. **Adaptive Software Development (ASD):**

   Adaptive Software development works on the principle of continuous adaptation. It uses repeated series of speculate, collaborate and learn cycles to develop the application.

5. **Crystal Clear and other Crystal methodologies:**

   Crystal family of methodologies is people-centric and focuses on enhancing the work of people involved. The logic is to start with small tasks and then build them up into larger ones. The overhead activities are reduced and efforts are diverted to work building. It’s a collaborative process involving tracking and iterations.
6. **Dynamic Systems Development Methods (DSDM):**

DSDM is based on Rapid Application Development (RAD) and can be used for projects which have tight schedule and budget. DSDM works on pareto principle-80/20 rule i.e. 80% of the business benefit comes from 20% of the system requirements. Therefore DSDM starts implementing this first 20% of system requirements to meet 80% of the business needs. And then it progresses to cover rest of requirements.

7. **Feature Driven Development (FDD):**

The main purpose of FDD is to deliver tangible, working software repeatedly in a timely manner. The development process involves 5 activities: Develop Overall Model, Build Features List, Plan by feature, Design by feature, Build by feature. Milestones are identified and progress is tracked on basis of the achievement of milestone.

8. **Lean software development:**

Lean software development is based on lean manufacturing principle. The principles followed are:

1. Eliminate waste i.e. anything that does not add value to customer.
2. Amplify learning by doing and testing things rather than documenting them and them implementing.
3. Decide as late as possible thus keeping all the possible the options open as long as possible in the life cycle.
4. Deliver as fast as possible. It works on basis of the fact that sooner the end product is delivered, sooner is the feedback received and sooner it is implemented in next iteration.
5. Empower the team by keeping people motivated.
6. Build integrity in by keeping the code simple and avoiding repetitions.
7. See the whole so the problems can dealt at application level, product level, instead of dealing with them at minute levels.
9. **Agile Unified Process (AUP):**

AUP is simplified version of Ration Unified Process (RUP). It distinguishes Development Release iterations from Production Release iterations. It works on building up a model, implementing, testing and deploying it and then followed by configuration management. The focus is on the covering the high-value activities rather than covering every possible thing