Agile Methodology and Banking Domain

Introduction:

Agile methodology is an iterative and incremental approach to software development which involves customer collaboration and customer feedback. Agile methodology can be used to develop banking products or applications. It mainly focuses on the customers by gathering feedbacks and making changes to the product based on what customer needs and by responding quickly to the market needs, so that one can create solutions that really meet their needs. It encourages different departments to work in coordination, improve their skills and develop a good working environment with the coworkers. Agile is flexible as it helps to adapt to changing needs of the customers, follow the rules and regulations, minimize the risks associated. It also focuses on the working software, thereby getting rid of paper works and other extra steps to make things proceed smoothly while using the resources in a better way.

MoSCoW method:

It represents Must have, Should have, Could have, Won't have. It is a method which is used to prioritize the features based on the importance and impact they have on the success of the project.

- **Must-have** : These are the elements which must be implemented in the project for it's success and these elements are critical and absolute in order for the core functionality to work.
- **Should have** : These are close to must haves but are not inevitable for the success of the project and can be delayed in case of limited resources.
- **Could have** : These elements are not as critical as must haves and should haves. They are the elements which can be considered if we have enough time and resources.
- **Won't have** : These are the elements which are unsuitable and are excluded from the project.

Outcome : Outcome of a project is the results, deliverables, or the objectives that are expected upon the completion of the project.

Customer value :

If the outcome of a project in some way is benefitted to a customer then it can be said that the product adds value to the customer. It is not only the product feature but also quality, cost, user experience, usability. The value added to the customer is most critical to assess the success of a project. It can be enhanced by gathering customer feedback.

Customer feedback :

It is the information provided by the customer about the product based on their interaction and experience with product, service, etc. If there is a negative feedback from the customer then it gives us the opportunity to understand what modifications are needed in the project. If positive feedback is received then it gives assurance about the strengths of the project.

The development of a project encompasses different roles and responsibilities, they are :

1. **Product Manager :** Product manager is the one to whom the Product Owner reports.

2. **Product Owner :**
    - The responsibility of a Product Owner is to define the requirements for a project, plan the entire strategy for the product, define the priority for each task.
    - Product Owner is the one who engages with development teams and ensures that customer needs are fulfilled.
    - Makes high level decisions during development process to ensure that user stories align with the product vision.

3. **Scrum Master :**
    - A person who ensures that the needs of scrum team is met in terms of infrastructure readiness, deliverables.
    - Scrum Master removes the obstacles that hinder the scrum team's performance thereby ensuring that the principles are followed and high value products are delivered.

4. **Scrum Team :**

   It is a team of developers and testers which includes Quality Assurance, UX Designers, Architects, Deployment Team.

   - Scrum Teams are self managed teams who are empowered to make their own decisions and manage their own work without the need for direct supervision. They deliver the work that has been planned.
   - Developers are responsible for building an actual product by writing the codes.
   - Quality assurance is involved in testing and verifying the product before the release to ensure that specified requirements are met and defects are addressed.
   - UX Designers create interface designs in such a way that they are user-friendly.
   - Deployment team is responsible to ensure that the product is correctly installed and configured so that it can be released.

In the development of a project, the project features or the requirements defined by Product Owner must be divided into hierarchical components to organize and manage work.

1. **Epics :**
   - Epics are large and complex elements that encapsulate a significant feature.
   - They are too complex that they cannot be completed in a single sprint.
   - So they are often divided into user stories. Product Owner is responsible for breaking down Epics into stories.
2. **Stories :**
   - User Stories are smaller units than epics and provide more detailed requirements.
   - They represent a specific and important functionality from the end user perspective.
   - No story should span sprints and it should be delivered as a part of a sprint as it is small enough.
   - Format for the story is :

     As a <user>

     I want to <perform action>

     So that I can <see result>

   The user stories can be estimated with **Story Points** :

   - Story points are usually estimated based on factor such as complexity.

- If there are new requirements or stories that need to be delivered then the stories need to be prioritized and de-prioritized on discussion with the PO so that the stories with higher priority can be delivered early.
- The stories are further broken down into tasks by the individual team members.

Some of the important terminologies that we have come across are:

1.Time box : Time box is a fixed duration in which a task is performed.

2. Planning Poker : Where multiple team members estimate based on a scale.

3. Sprint :

- It is a time boxed iteration during which a development team focuses on delivering prioritized user stories.
- A sprint can last up to 2-4 weeks of duration and no changes are allowed in sprints.

4. Sprint Backlogs : List of items which are delivered as a part of sprint.

Release Plan :

Release plan is an overview of schedule and release of a product which includes product roadmap and product backlog.

Some of the terms included are:

Product Roadmap : Visual guide providing overview of the direction for the development of a product along with timeline.

Product Backlog : It contains all the items that the Product Owner wants to be delivered.

Definition of Done : Set of conditions that user stories or tasks must meet to be worked upon and ready for release.

Definition of Ready : It is the set of conditions that user stories must meet to be accepted into a sprint for the implementation.

Process for execution of a project :

1. Sprint Planning :

Where development team collaboratively determines the works to be done during the upcoming sprint. It occurs at the beginning of each sprint and it involves backlog review, estimation, doubt clarification etc.

2. Daily Standup :

It is a 15 minutes catchup for a team to discuss on what are the actions that has been performed, what is to be performed and what are the blockers that came along as team proceeded. If there are any blockers then they are discussed outside the standups.

3. Sprint Review :

It is the demonstration with business user towards the end of sprint. The business user then provides feedback to the team.

4. Retrospective :

It is a meeting held at the end of each sprint to reflect upon What went well for the team during the process, What went wrong and How can we improve the performance.

Testing :

- **User Acceptance Testing :**
  It is a phase in software development where the users perform real world scenarios to ensure that the product meets the user requirements. It is used to identify any issues before the system is released.
- **Load Testing :**
  The system is tested under different loads to check the system stability and identify the weakness or bottlenecks of the system. It is to ensure that there is no performance degradation.

- **Integration Testing :**
  In Integration testing, different components or modules are integrated and made to interact to verify if they work together as intended. It ensures that individual units ,when combined functions properly as a single unit.
- **Edge case Testing :**
  It is testing very unusual cases or testing system's behavior at extreme situations. It ensures that system can handle unusual cases very well and does not produce unexpected results.
- **Continuous Integration and Continuous Deployment :**
  It ensures frequent or regular integration of components and provides building ability in the system to deploy at will or release in demand.

Slicing and Dicing of stories can be done based on :

Independent : Stories should be able to be developed and tested independently.

Negotiable : The stories should be adjustable to the requirements.

Valuable : Each story that we select must add value to the customer.

Estimable : It should be possible to estimate with story points for planning purpose.

Small : Stories must be as small as possible.

Testable : Stories should pass the acceptance criteria and be easily testable.

3 C's of Agile :

- Card
- Conversation
- Confirm

Agile Values:

1. Individuals and interactions over processes or tools.
2. Working software over comprehensive documents.
3. Customer collaboration over contract negotiations.
4. Responding to change over following a plan.

Implementation of Agile methodology in designing a brochure for Banking Domain:

A case study was conducted on designing and developing a digitalized Brochure for Banking Domain. First we divided ourselves into different teams with each team having a Product Owner, a Scrum Master, and a Scrum Team. While the Scrum Master ensured that team needs are met, the Product Owner defined the requirements for the team. The Scrum team first specified 5 epics for the brochure. Then each epic was broken down into five stories. Then brochure was designed by each of the teams. Then daily standups and sprint planning were held to discuss on what needs to be done within a specific timeline. We prioritized and de-prioritized these stories which resulted in each epic having 3 stories.

As a part of the brochure, one epic was developed completely in the first sprint and a demonstration of the same was given. Various feedbacks, suggestions were received and they were taken into consideration so as to satisfy the customer needs. Again standups and review meetings were held to develop a complete product. The product was developed completely by each team with their own unique ideas. Then the retrospectives were discussed and there were few takeaways from the hands-on workshop. At the end, final presentation of the product was conducted.

Conclusion :

In conclusion, applying agile methodology in Banking Domain has proven to be more flexible, responsive, and collaborative rather than using any other models like waterfall model. The iterative nature of Agile helps to adapt to the changing market demands and trends. As we apply agile principles and values the system develops more customer-centric solutions. By breaking down the larger epics into smaller stories, agile strengthens prioritizing of tasks in an efficient manner.