AGILE METHODOLOGY AND BANKING DOMAIN

INTRODUCTION

Agile methodology in the investment domain includes conforming the principles of agile to create financial spreadsheets that meet the specific needs and challenges of making investments. It is about regularly adding new features to software, encouraging different teams to work together, and making sure the customer benefits the most. In investing, Agile helps us to quickly respond to market changes, deliver new features fast, and make customer happier by listening and making regularly. Important practices include preparing for challenges, practicing regularly and making changes regularly. Important practices include preparing for challenges, practicing regularly, improving skills and testing ability with co-workers. This method allows banks to quickly deal with changes in how they are supervised, what customers want and advance in technology. It also ensures that software is transferred to the best quality.

MOSCOW RULES

Moscow is a prioritization technique used in the agile project management. The terms stand for "Must-have, Should-have, Could-have and Won't have". It helps the teams to prioritize features and requirements based on their importance.

- **Must-have:** These are the essential features that are necessary for product to fulfill its primary purpose. They are non-negotiable and must included in the first release.
- **Should-have:** These features are important but not critical for initial release. They can be included if time and resources are available.
- **Could-have:** If these features are there in product it would enhance the product but are not essential. They can be considered for the future releases.
- **Won't-have:** These are features that are explicitly decided not to include in the current release of the product.

Next, we discussed about customer value, customer feedback and stakeholders play a crucial role in product development.

- **Customer value:** It refers to the benefits that customer receives from using the product or service relative to its cost. Prioritizing the customer value help in the creating products that are relevant and competitive with the target market, leading to higher customer satisfaction and loyalty.
- **Customer Feedback:** It provides valuable insights into how customers perceive and interact with the product. It helps in the identifying the areas for improvement and uncover new opportunities. Feedback can be done through surveys, user interviews, usability testing and analytics.
- **Stakeholders:** Stakeholders are the individuals or groups with an interest in the success of the product, including customers, investors, internal teams and partners. Involving stakeholders early in the process ensures that product meets business objectives and other constraints which still delivering value to the customer.

Roles and Responsibilities

In the product development there includes many roles such as product owner, product manager, scrum team and scrum master. Each one has its own responsibilities. such as

- **Product owner:** The product owner is responsible for defining the product vision and gathers, refines and communicates requirements to the development team. They prioritize the product backlogs based on the business value and customer needs, ensuring that most valuable features are delivered first. Product owner makes critical decisions regarding to the product such as release plan and scope change.
- **Scrum Team (Development Team):** Teams consisting of individuals with diverse skills (such as Developers, UX Designers, Architects, Quality Assurance, Testers) who collaborate to deliver increments of working software. The team is self-organized and responsible for deciding how to accomplish the work and meet the sprint goals. They commit to delivering the sprint backlog items and work collaboratively to achieve to achieves sprint goals within timebox. The team reflects on their processes and practices during the sprint retrospective, identifying areas of improvement and implementing changes to enhance the productivity and quality.
- **Scrum Master:** They keep track of deliverables to achieve the sprint goals within the timebox and ensures that teams needs are met in terms of infrastructure readiness. The

scrum master identifies and resolves impediments that hinders the teams progress, working to eliminate blockers and improve team velocity. They facilitate communication between the development team and stakeholders ensuring transparency and managing expectations.

The key aspects of agile is to break down the features into epics, user stories, story points and tasks. Steps as follows,

- **Identify Epics:** Identify high level features, known as epics. These are too big to implement in a single sprint.
- **Break down Epics into User stories:** After identifying epics break them down into smaller and manageable chunks called user Stories. User Stories represents individual pieces of functionality from end user's perspective. Each user story should follow the

  "As a [Type of user],
  I want to [perform action],
  So that I can [see result]"
- **Estimate user stories with story Points:** Assign story points to each user story to estimate its relative size or complexity compared to other user stories. Story points based on the factors such as complexity.
- **Prioritize User Stories:** Once the user stories are estimated, prioritize them based on their importance and value to the product and its customer's needs.
- **Break Down User Stories into Tasks:** For each user story, break down the work into smaller tasks or sub tasks. They should be completed in single sprint.

We have learnt some important terminologies such as,

- **Time Scale:** Timescale refers to the duration within which task is planned to be completed.
- **Time Box:** Fixed duration of time allocated for a specific activity.
- **Sprint:** Sprint is a time-boxed iteration during which a development teams work to deliver a set of features.
- **Planning poker:** It is used to estimate the effort or size of user stories in project.
- **Sprint Backlog:** Sprint Backlog contains the list of tasks selected by the development team to be worked on during the specific sprint.

Next day we learnt more about agile which we use while developing a product.

Release Plan

A release plan is document that outlines the schedules and scope for delivering the a specific set of features or enhancement to a product. It provides a roadmap for coordinating and managing the release process, including the timing of release and content of each release.

Where we learnt about,

- **Product Backlogs:** The product backlog is a prioritized list of all features, enhancement, bug fixes and other work that needs to be done to a delivery of product.
- **Product Roadmap:** A product roadmap is a strategic document that outlines the vision, goals and direction of product over time.
- **Swim lanes:** It provides a structured way to organize and categorize information, tasks based on specific criteria.
- **Definition of Ready (DOR):** It is a checklist of criteria that product backlog item must meet before the development team starts working on it.
- **Definition of Done (DOD):** It is set of conditions that a product backlog item must meet to be considered and ready for release.

The process of execution includes,

1. Sprint Planning
2. Daily Standups
3. Sprint review
4. Retrospective

Sprint Planning

Sprint planning is a collaborative meeting that marks the start of new sprint. Its purpose is to define what work will be accomplished during the sprint and how that work will be achieved. The entire scrum team including the product owner, scrum master and development team, participate in sprint

planning. During sprint planning the team select the stories from product backlog to work on during the sprint. They define sprint goal, breakdown user stories into smaller tasks, estimate efforts and create sprint backlogs.

Daily Standup

The Daily Standup is short, time-boxed meeting held every day during the sprint. Its purpose is to synchronize the team's activities, discuss progress, and identify any blockers are there. Duration of meeting is 15-30 minutes. The Daily standup involves the development team and facilitated by the scrum master. Each team member answers three questions:

1. What did I accomplish yesterday?
2. What will I do Today?
3. Are there any obstacles preventing me from making progress?

The focus is on quick updates and identifying any issues that need to be addressed.

Sprint Review

The sprint review is held at end of each sprint to demonstrate the work completed during the sprint stakeholders and gather feedback. During the sprint review, the development team presents the completed user stories or features to stakeholders. They discuss what was done, what was not done and any changes to the product backlog based on feedback received. The sprint review is an opportunity for stakeholders to provide input and shape the direction of the product.

Retrospective

The retrospective is a meeting held at the end of each sprint to reflect on the team's processes and identify opportunities for improvement. It includes scrum team and scrum master. During the retrospective the team reflects on sprint, discussing what went well, what went wrong, what could be improved and any action item for next sprint. They identify and prioritize improvements and create a plan for implementing them in future sprints.

Next, we discussed about testing,

- **User Acceptance Testing (UAT):** UAT validates that the software meets the requirements and expectations of end user and stakeholders. It involves executing test cases that

stimulates real world usage scenarios to determine whether the software behaves as expected and satisfies user needs.

- **Load Testing:** Load Testing assesses the performance and scalability of a software application under expected and peak load conditions.
- **Edge Case Testing:** It involves identifying and testing scenarios that are unlikely to occur under normal conditions but have the potential to cause unexpected errors.
- **Integration Testing:** Integration testing is a software testing technique that verifies the interaction between different components or modules of a software as expected. It focuses on identifying and addressing issues related to the integration point between components, such as communication protocol, data exchange formats and dependencies.

Continuous Integration and Continuous Deployment (CICD)

CICD are practices in software development processes aimed at improving efficiency, quality and speed of delivering software applications. Continuous integration is the practice of frequently integrating code changes into a shared repository. Continuous deployment is practice of automatically deploying code changes to production. It aims to streamline the release process and deliver new features and updates to users quickly and frequently.

3 C's of Agile

1. **Card:** User stories are written on index cards or digital cards and serves as placeholder for conversations about requirements.
2. **Conversation:** Conversation refers to the ongoing dialogue and collaboration between team members and stakeholders to discuss and refine requirements.
3. **Confirmation:** Confirmation refers to the acceptance criteria associated with each user story, which defines condition that must met for the story to be considered complete and ready for release.

Slicing and dicing of Stories can be done based on

1. **Independent**
2. **Negotiable**
3. **Valuable**
4. **Estimate**

5. **Small**
6. **Testable**

Values of Agile are,

- Individual interactions over process and tools
- Working software over comprehensive documentation.
- Customer collaboration and contract Negotiation.
- Responding to change over following the plan.

Practical Implementation of Agile Methodology in Designing a Brochure for Banking System

A case study was conducted on designing and developing a digitalized brochure for banking system. First, we divided into the groups and each group member into product owner, scrum master and scrum team. We started with identifying the Epics for the brochures. We gathered a set of epics and listed out them based on the MOSCOW Rule. Next, we breakdown epics into stories with Story format and then into tasks, we done the sprint planning. Scrum team which is self-managed divided work among themselves. While designing daily standup was done to keep track of progress, what to do next and whether they are facing any problems. After each sprint, sprint review was done with stakeholders and feedback is taken from them. If there are any feedback, we correct it accordingly. If stakeholders are satisfied, then we start with the new sprint. After our sprint review, we received feedback that was not handy, and we made the changes in design and designed according to the feedback received. After Retrospective was done to discuss in which and all areas need to be improved.

Conclusion

In conclusion, Agile methodologies emphasize a collaborative and iterative approach to software development, with a focus on delivering value to the customer continuously. The principle and practices discussed, including the Moscow principles, roles and Responsibilities and scrum team, provide the framework for teams to effectively plan, execute and delivery of product. By embracing Agile principles and practices teams can enhance collaboration, transparency and responsiveness leading to more successful and satisfying outcomes for both development teams and end-users alike.