Assignment No. 05

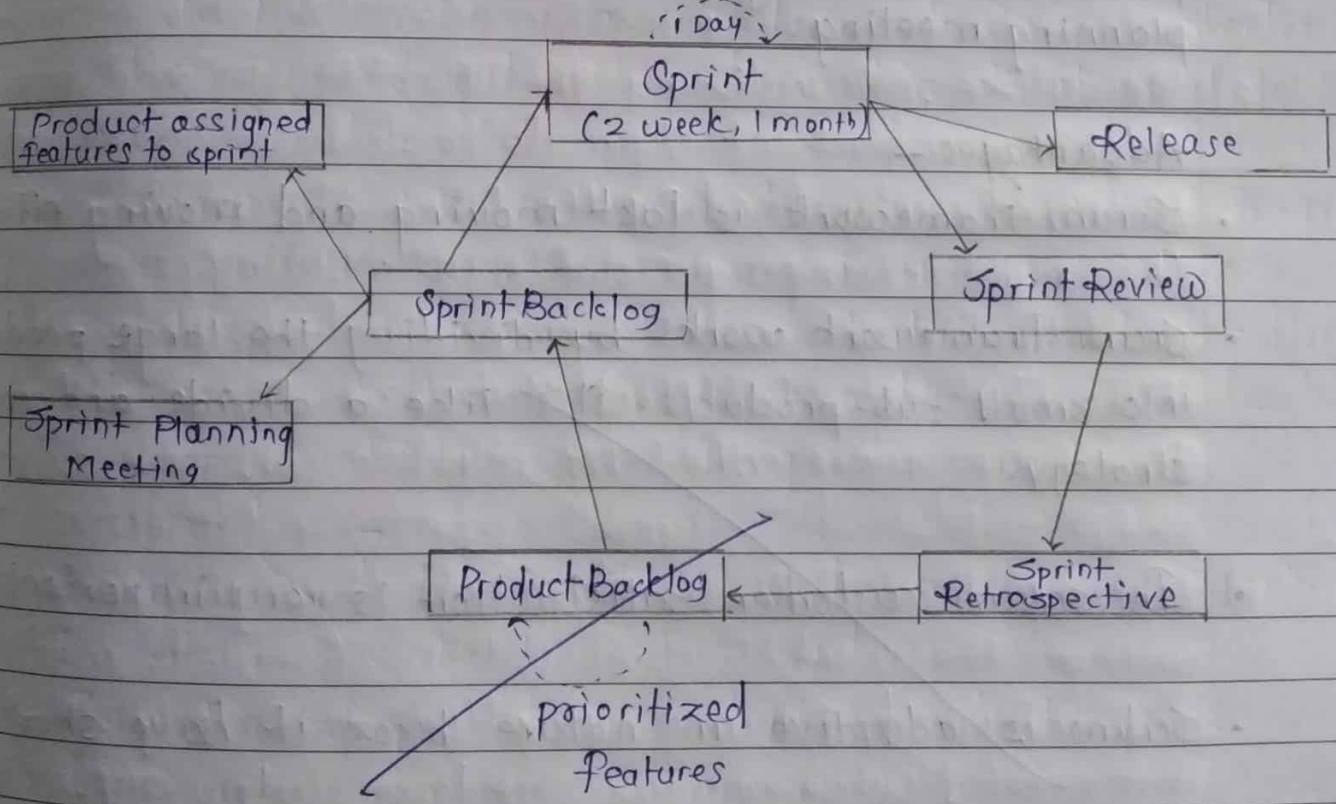
Q1. Describe Agile Scrum Process lifecycle with neat diagram.

Scrum is the type of agile framework. It is a framework within which people can address complex adaptive problem while productivity & creativity of delivering product is at highest possible values. Scrum uses Iterative Process.

Salient features of scrum are:

- scrum is light - weighted framework
- scrum emphasizes self - organization
- scrum is simple to understand
- scrum framework help the team to work together.

Lifecycle of scrum:



Sprint: A sprint is a timebox of one month or less. A new sprint starts immediately after the completion of the previous sprint.

Release: When the product is completed, it goes to the release state.

Sprint Review: If the product still has some non-achievable features, it will be checked in this stage and then passed to the sprint Retrospective stage.

Sprint Retrospective: In this stage quality or status of the product is checked.

Product Backlog: According to the prioritize features the product is organized.

Sprint Backlog: Sprint Backlog is divided into two parts product assigned features to sprint and sprint planning meeting.

Advantages –

- Scrum framework is fast moving and moving efficient
- Scrum framework works by dividing the large product into small sub-products. It is like a divide and conquer strategy.
- In scrum customer satisfaction is very important.
- Scrum is adaptive in nature bcoz it have short sprint

Q.1. Describe kanban methodology with kanban principle and practises.

→ Kanban is a popular Agile Software Development Methodology. It is basically a signaling device that instructs the moving of parts in a 'pull' production system, developed as part of the TPS (Toyota Production system). Kanban is about envisioning the existing workflow in terms of steps. These steps can be creating on the whiteboard.

Principles of Kanban:-
Kanban is based on four key principles :

1) Start with existing process : It is a change management method that starts with the existing process. Changes are done in the system in incremental and revolutionary ways. Unlike, Scrum, there is no specific process or roles defined in Kanban.

2) Agree to continue evoluttionary and incremental changes: After starting with existing process, the team must agree on continuous, incremental and evolutionary changes. The changes should be small and incremental. Rapid and substantial changes may be effective but they will be subjected to larger resistance as well by the team.

3) Admire current roles, processes, responsibilities & titles: Though kanban suggests continuous incremental changes in the process, it respects current roles, responsibilities, and job titles. This helps the team to gain confidence as they get started with Kanban

4) Leadership at all levels: Kanban does not expect leadership from a specific set, rather the actions of leadership at all levels in the organization, are very encouraged.

Kanban Practices:-

The following are the six core kanban practises:

1) Limit WIP: Limiting Work-In-Process (WIP) implies that a pull system is executed on either parts or the cohole workflow. It (PULL system) coill act as one of the key stimuli for incremental, continuous & evolutionary changes to the system. Limit WIP assigns explicit limits to the number of items that may be in progress at each workflow state.

2) Visualize: Visualizing the workflow and making it visible is important so as to know how work proceeds. Without understanding the flow of work, incorporating the right changes is difficult. Usually, a card coall coith columns and cards is used to visualize the flow of work. Different states or steps within work-flow are represented by columns on the card coall.

3) Manage Flow: Flow of work through every state coithin the workflow should be observed, measured and informed. By managing the flow vigorously, the incremental, continuous and evolutionary modifications to the system can be assessed to have negative or positive effects on the system.

4) Improve collaboratively, Evolve experimentally: Kanban encourages small incremental, continuous and evolutionary changes. Whenever teams have common understanding of concepts about work, process, workflow and risk, they are more likely to be able to form a shared understanding of a problem and suggest enhancement actions that could achieve a consensus.

5) Implement Feedback Loop: Early feedback from clients and the pull system are important in Kanban. If we get feedback from different stakeholders and processes, it will help to eliminate risk and optimize the delivery process.

6) Make policies explicit: Until the mechanism of a proces is not made clear, it is difficult to hold a debate & discuss ways to improve it. Without a clear understanding of how work is truly done and how things actually work, any conversation of complications tends to be anecdotal, emotional and subjective. With a clear understanding, it is possible to hold a more rational, empirical, objective discussion of issues. It is more likely to Facilitate consensus around improvement suggestions.

Q.3. Difference b/w^n Scrum and Kanban.

	Scrum	Kanban
Origin	Software development	Lean manufacturing
Ideology	Solve complex problems while delivering valuable products	Use visuals to improve work flows and processes.
Practices	Sprint planning Sprint Daily scrum Sprint review Sprint retrospective	Visualize the flow of work Limit work in progress Manage flow make process policies explicit Implement feedback Loops improve experiment.
Roles	Product owner Scrum master Development team	No formal roles
Metrics	Velocity	Cycle time throughput

Q.4. Short Note-

A. Story Mapping-

1. Story mapping or user story mapping is a technique used in product discovery; outlining a new product or a new feature for an existing product.

2. The result is a story map: All the user story arranged in functional groups. This keeps your eye on

the big picture. While also providing all the details of the whole application.

3. Story maps were first introduced by Jeff patton in 2005. The main idea behind story maps is that single-list product backlogs are a terrible way to organize and prioritize the work that needs to be done.

4. A richer structure is necessary. A user story map is a powerful tool that enable an agile team to groom their product backlog and plan the product releases more effectively.

5. A user story map captures the journey a customer takes with the product including activities and tasks they perform with the system.

6. Creating a story map collaboratively ensures team members are on the same page from the start of page through to ongoing development of new releases.

B. Test - Driven Development -

1. Test - Driven development (TDD) is a development practise that involves writing automated tests before writing the code. This helps to ensure that the code meets the requirements and reduces the likelihood of defects.

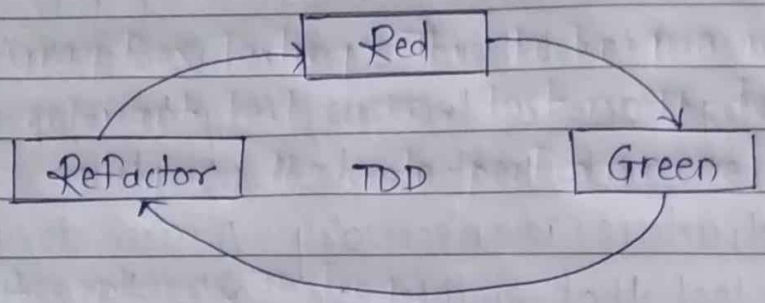
The following sequence steps are generally followed:

1. Add a test - Write a test case that describe the functio completely. In order to make the test cases the developer must understand the features and requirements using user stories and user cases.

2. Run all the test cases and make sure that the new test case fails.

3. Write the code that passes the test cases. Run the test cases.

4. Refactor code - This is done to remove duplication of code. Repeat above mentioned steps again & again



Red - Create a test case and make it fail
Green - Make the test case pass by any means.
Refactor - Change the code to remove duplicate/ redundancy

A. Pair Programming-

1. Pair programming involves two developers working together on the same code. This helps to improve. Code quality, share knowledge and reduce the likelihood of the defects.
2. Pair programming is a programming method in which two people work together on a single program.
3. The first person is the "Driver", who writes the code the other person is the "Navigator" who reviews each line of code as it is typed, checking for errors. They exchange their roles on a regular basis.

There are three pairing variations -

1. Newbie-Newbie pairing can sometimes give a great result. Because it is better than one solo newbie. But generally, this pair is rarely practiced.

2. Expert-Newbie pairing gives significant results. In this

pairing, a newbie can learn many things from expert, and expert, and experts gets a chance to share his knowledge with Newbie.

3. Expert-expert pairing is a good choice for higher productivity as both of them would be expert, so they can work very efficiently.

Advantages:
1. Reduce errors
2. Better workflow and focus
3. Improves morale
4. Mutual and continuous learning
5. Team union

B. Unit Testing:
1. Unit testing is a type of software testing that focuses on individual units or components of a software system.
2. The purpose of unit testing is to validate that each unit of the software works as intended and meets the requirements.
3. Unit testing is typically performed by developers, and it is performed early in the development process before the code is integrated and tested as a whole system.

There are 3 types of unit testing techniques-

① Black-Box Testing: This testing technique is used in covering the unit tests for input, user interface, and O/P parts.

② White-Box Testing: This technique is used in testing the functional behavior of the system by giving the input

and checking the functionality o/p including the internal design structure and code of the modules.

③ Gray Box Testing: This technique is used in executing the relevant test cases, test methods, test functions and analysing the code performance for the modules.

C. Acceptance Testing:

1. Acceptance testing is a method of software testing where a system is tested for acceptability.
2. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.
3. It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorized entities to determine whether to accept the system or not.

Types of acceptance testing:

1. User acceptance test (UAT)
2. Business acceptance testing (BAT)
3. Contract acceptance testing (CAT)
4. Regulations acceptance testing (RAT)
5. Alpha testing
6. Beta testing.

24/3/23