Agile Development
UNIT -2
Subject: Software Engineering (2160701)
Semester: 6th (BY)

Q.1. what is Agile? Why is Agile important?

Agile is a set of *practices, values*, and *principles* for software product development. In software product development, we think about "methodologies," "activities," "interactions," "results, work products or artifacts;" we think about "processes" that we use to organize the work:

- documents
- meetings and reviews
- diagrams and models
- coding and user documentation standards

The core ideas in Agile Development:

- Adaptive
- Iterative/incremental
- People-oriented

Adaptive means that the teams and the process should be flexible in the presence of "rapid-fire change".
Iterative and incremental means that Agile Development produces working products in stages – a growing set of "completed and working software".
People-oriented means the team organization and processes will support good people, who are the most important ingredient to project success.

Agile Development as a "software development framework" says:

- keep things small
- deliver partially-completed software frequently
- talk to the customer often
- write more code than documentation
- everyone on the team learns together

There are many Agile practices:

- short time boxed iterations
- continuous integration
- daily unit testing
- regular retrospectives
- direct communication between developers and the customer or a customer surrogate
- a single list of features and tasks
- short-term estimation of development tasks
- information radiators
- refactoring

Q.2. What are the basic principles of Agile Development?

AGILE PRINCIPLES :>

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	9. Continuous attention to technical excellence and good design enhances agility.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	10. Simplicity--the art of maximizing the amount of work not done--is essential.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	7. Working software is the primary measure of progress.	11. The best architectures, requirements, and designs emerge from self-organizing teams.
4. Business people and developers must work together daily throughout the project.	8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

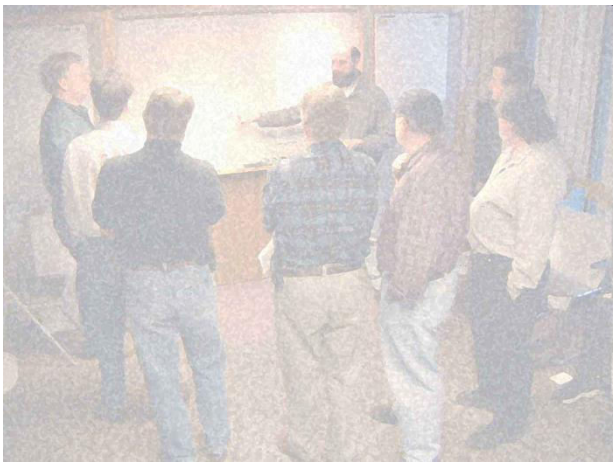
Q.3. Explain other Agile Development tool.

In this course, we will discuss the Scrum methodology

- Scrum has been around since the early 1990s
- The structure of Scrum is very simple (3 roles, 3 meetings)
- Scrum is not as “extreme” as some other methodologies

What is a Scrum?

- It is a meeting with attitude – good teamwork is necessary



The Scrum presentation is short and simple:

- Scrum iteration process
- Product Backlog
- Roles: Team Member, Product Owner, and Scrum Master
- Project estimation and iteration estimation
- Daily Scrum Meeting

- Management
- Retrospectives

Scrum is designed to organize the work of a single cross-functional team

The team will do software product development this way:

1. Iteration planning – create a plan for one iteration
    - Select next features or sub-features to deliver (choose from highest priority items), define and estimate tasks, negotiate scope of the delivered product
2. Iteration execution – implement the items in the plan
    - Fill in missing requirements, design, code, integrate/build, and test the modules needed in the plan
3. Deliver the results of the iteration – give a demo

- Steps 1 – 3 will be executed many times – based on the Release Plan
- Each cycle is a fixed-length timebox:
    - Always end each iteration on schedule, even if it isn't complete
        - (Don't say – "we can finish everything in this iteration in 2 more days". Just deliver and run the next iteration planning meeting.)
    - The team learns to make good short-term estimates – so over time, most of the iterations will deliver as expected

Scrum Elements:

THREE Roles

- Product Owner
- Scrum Master
- Team Member

THREE Meetings

- Planning (Release & Sprint)
- Daily Scrum
- Sprint Review

THREE Lists

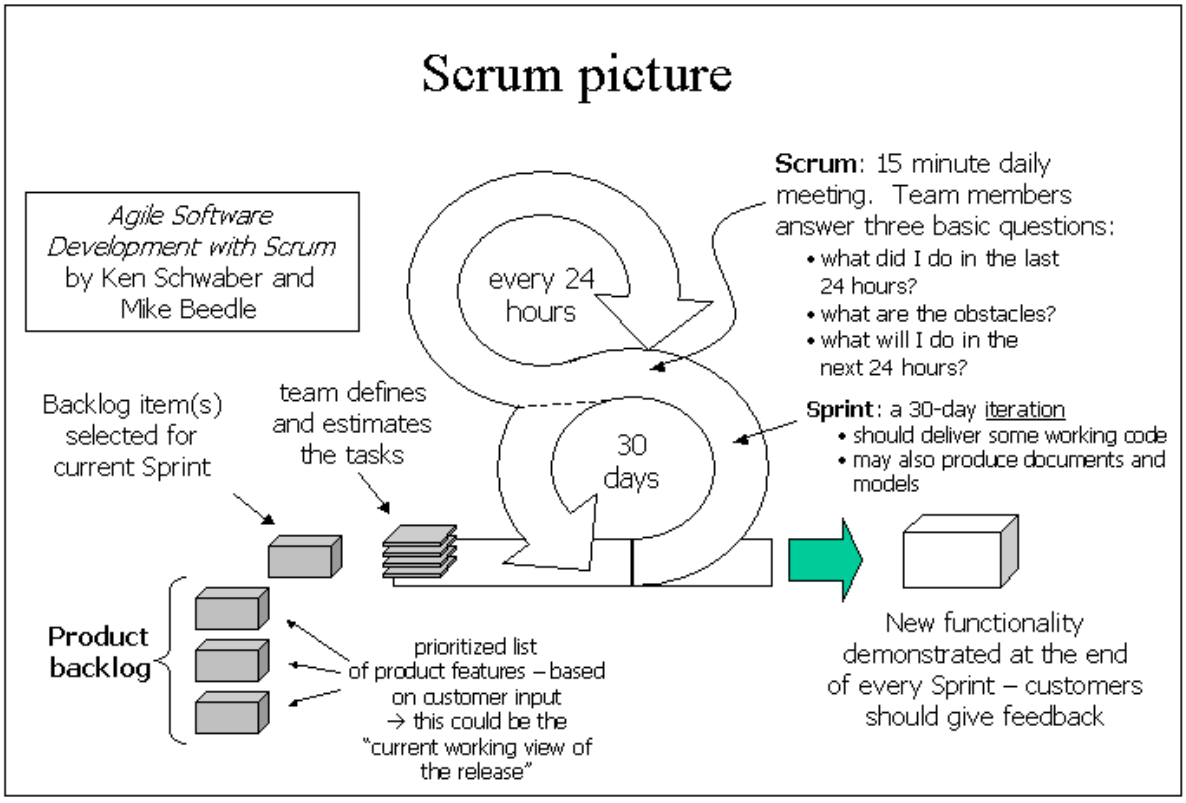
- Product Backlog
- Spring Backlog
- Impediments List

The *Product Backlog* is the set of all features and sub-features that you know you need to do to build the product

- This is the "plan" for multiple iterations
- The items in the Product Backlog is ordered by priority – value to the customer
    - you want to deliver some value to the customer in each iteration, so you put the most important things early
- It is OK to add things to the Product Backlog at any time

A Scrum iteration (called a *Sprint*) contains a list of tasks and work product outputs that will be done in a 4-week* timebox

- At the beginning of the 4 weeks, each team member has a pretty good idea of what they will be working on
- Management should not add new work product outputs to the Sprint – any new items should be added to the Product Backlog instead
- If new work items are important enough, they will get done in the next 4 week iteration



The Product Backlog – managers and customers use it to set the working agenda of the development team

- Managers and customers work with Product Owner to set the priority of each item
- Development team estimates the size/effort for each item
- Even if the managers and customers don't like the estimates, they are not allowed to change them

Product Owner

- Responsible for the ROI
- Available for the Team during the whole product development period
- Gets answers to all requirements questions
- Talks with customers and understands their priorities
- Keeps the Product Backlog current

Scrum Master

- Scrum rules guardian
- Coach the team
- Removes impediments
- Prevents outside interference during an iteration
- Scrum Master is both a teacher and a referee

The Scrum Team has two kinds of "once-per-iteration" meetings:

- An Iteration Planning meeting at the beginning of each Sprint
- A Sprint Review meeting at the end of each Sprint

In addition, the Scrum Team has one daily meeting: the Daily Scrum

- Daily Scrum is 15 minutes – no longer
- Everyone is supposed to speak:
  "This is what I did yesterday."
  "Here is what I am planning to do today."
  "These are the obstacles in my way."
- No problem solving in the meeting – everything is taken offline later.

Q.4. Short Note: Extreme Programming (Agile Development Process Model)

- Extreme Programming (XP) takes an 'extreme' approach to iterative development.
  - New versions may be built several times per day;
  - Increments are delivered to customers every 2 weeks;
  - All tests must be run for every build and the build is only accepted if tests run successfully.
- Incremental development is supported through small, frequent system releases.
- Customer involvement means full-time customer engagement with the team.
- People not process through pair programming, collective ownership and a process that avoids long working hours.
- Change supported through regular system releases.

Maintaining simplicity through constant refactoring of code

The extreme programming release cycle: