Agile Scrum Course

(https://youtu.be/VFQtSqChlsk)

- What is Agile?

Background:

- Before Agile it was difficult for developers to introduce changes in development because of waterfall model.
- Waterfall models doesn’t have the scope to move backwards in the process.
- Each step must be completed first to move to next step.
- End users were not given importance.
- Testing is delayed to step by step model.

Agile was introduced in 2001.

Agile Manifest: A set of principles used in project management and software development.

Principles to follow for making process Agile:

1. Customer satisfaction. (continuous delivery)
2. Welcome change. (easy introduction of change during development)
3. Deliver frequently.
4. Work together.
5. Motivated team.
6. Face to face interaction.
7. Working software.
8. Constant pace.
9. Good design.
10. Simplicity.
11. Self-organization.
12. Reflect and adjust.

Advantages of Agile:

- Continuous interaction with clients.
- Improved transparency to clients.
- Delivery of output is predictable.
- The cost of project is predictable.
- Allows for changes to be refined and reprioritized.
- Client can provide priority of features.
- Project broken into smaller units with high quality development.

Types of Agile Methodology:

1. XP (Extreme Programming)
   When to use: changing requirements
   To handles risk due to new tech.
   Small team.
2. Kanban
   When to use: when work arrives in unpredictable manner.
   Deploy work immediately without waiting for other features.
3. Lean
   When to use: apply to any sector where there is waste of any form.
4. Scrum
   When to use: when cross functional teams are working on product development, when work is split into more than one 2-4 weeks iteration.
5. Crustal
   When to use: focuses on strengthening team communication, focuses on continuous integration, active user involvement.

Definition:

- Agile is a set of methods and practices that focuses on iterative development.
- Requirements and solution are obtained thanks to self-organizing cross functional teams collaborating.
- Teams size 7+1 or 7+2 members.

- What is Scrum?

    - Daily scrum – 15 minutes meeting daily, tell what is completed and what needs to be completed.

History:

1986 scrum introduction.

1995 early version of agile was created.

2001 agile alliance founded.

2002 certification added.

2006 Scrum Inc. created.

2009 scrum.org created.

2010 first scrum guide published.

Definition:

- Framework that enables team to work together.
- Teams can: learn from experiences, self-organize working on problems, to reflect on their victories, their losses to improve.

Benefits of scrum:

- Project deliverables in efficient manner.
- Time and money used efficiently.
- Project divided into smaller units called sprint.
- Works best for fast moving projects.
- Meetings provide great visibility.
- Constantly involves feedback from customers and clients.
- Making changes based on feedback is easy.
- Individual efforts of the team members are given focus.

Scrum team involves:

1. Product Owner.
2. Scrum Master.
3. Scrum team.

1. Product owner:
    a. Primarily responsible for maximizing the ROI.
    b. Determine the product feature.

    c. Prioritize the activities.

2. Scrum Master:
    a. Helps team learn and apply scrum.
    b. Closely works with team.
    c. Helps in removing obstacles in activities.

3. Scrum team:
    a. Collection of all the individuals.
    b. Team members have different capabilities.

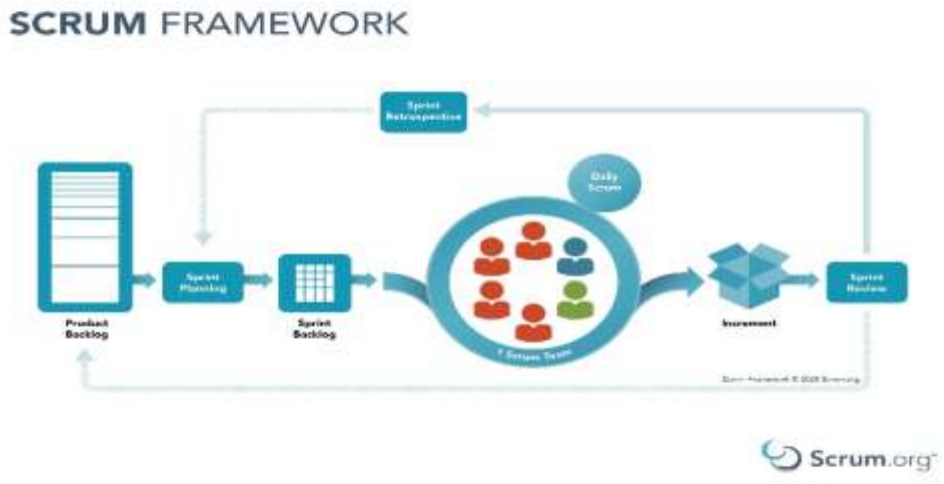
Scrum Artifacts:

1. Product backlog:
    a. List of new features, changes to be made to existing features, bug fixes, changes to infrastructure and several other activities.

2. Sprint Backlog:
    a. Sprint refers to short period of iteration.
    b. Helps in delivering software in frequent manner.
    c. A subset of the product backlog contains tasks that the team aims to complete to satisfy the sprint goal.

3. Product Increment:
    a. Combination of all product backlog completed in sprint and previous sprint.
    b. Outcome should be in usable condition, even if it is not intended to be released.

Scrum Frameworks:



Scrum Board:

- Physical or virtual tool to visualize items in sprint backlog.
- Shows activities that needs completion.
- Usually accessible to all members.
- Slots are to-do, in progress and done.

1. Agile Project Management:

What it is?

1. Flexible approach to building a project.
2. Project is broken down into several stages of sprints.
3. It works on delivering sections of project or mini projects.
4. No central control of project manager as it was in traditional approach.

Agile development cycle:

1. Product backlog
2. Sprint backlog
3. Sprint
   a. Planning
   b. Design
   c. Testing
   d. Release
4. Final product

Why Agile project management?

1. High product quality
2. Customer satisfaction
3. Reduced risk
4. Better and faster return on investment

Principles of Agile project management:

1. Satisfaction of customer by delivering the project fast with less number of errors.
2. Decrease the amount of time between planning and delivery phase.
3. Managers and developers work together to increase productivity.
4. Changes introduced can be taken up during development phase.
5. Better coordination among team members.
6. Monitoring progress of project and finding solution.
7. Trust and support must be there to complete the project objective.
8. Face to face conversation must be done.
9. Keep the solutions to the problem simple.
10. Scrum tools must be used for simplicity.

Steps in agile project management:

1. Project planning:
    a. Feasibility study
    b. Development of scope
    c. Time estimation.
    d. Breaking the project into sprints.
2. Roadmap creation:
    a. List of activities created.
    b. Steps to achieve is decided.
3. Release planning:
4. Sprint planning
5. Daily meetings
6. Sprint review and retrospective

Agile Project Management Frameworks:

1. Kanban framework: Kanban board is used for visual representation.

2. Scrum framework
3. Hybrid framework: combination of agile and non-agile methodology
4. Lean framework

Companies using agile:

- IBM
- Cisco
- Microsoft
- AT&T
- Philips
- Samsung

1. Agile User Stories

What are user stories?

- Agile tools that provides the users with simple, natural language explanation of one or more features written from the end users perspective.
- Doesn’t goes in details.
- Building blocks for epics and initiaties.

  Epics:

  - Group of user stories.

  Initiatives:

  - Combination of multiple initiatives.
- User stories are recorded on project management sofwares.

Advantages:

1. Delivers high quality content.
2. Eases collaboration with team members.
3. Helps understand users better.
4. Improves transparency.

5. Reduces risks.
6. Supports iterative development.
7. Focuses on vocal communication.

Investing in User Stories:

INVEST is a concept that helps in creating meaningful user stories.

1. **I**ndependent:
   Stories should be independent of one another so that they can be developed and delivered separately.
2. **N**egotiable:
   Stories should be discussable and should be open for negotiation.
3. **V**aluable:
   Stories must ensure there's value being added to the customers.
4. **E**stimable:
   The stories must be estimable and can be divided into tasks.
5. **S**mall:
   Stories shouldn't be too big and should be completed in about 40 hours or 3 to 4 days.
6. **T**estable:
   They should have an acceptance criterion that can be tested to check if they fulfill the customers need.

How to write user stories?

The 3C's of user stories:

1. Card:
   A card provides a written description of the story. This is used for planning and estimation.
2. Conversation:
   This represents discussion between users, team, product owners, etc. it helps build a shared understanding.
3. Confirmation:
   These represents conditions that need to be satisfied to ensure that the story meets all requirements.

Life cycle of user stories:

1. Pending:
   User stories in basic form are created after communication with the user and the project team. These act as a reminder for further discussion.
2. To-do:
   After discussion with stakeholders, user stories that need to be addressed are decided and put into sprints.
3. Discussion:
   At this stage user confirm the requirements and acceptance criteria. The end-users are shown a preview of upcoming features.
4. Developing:
   Once the discussion is complete, the team would be able to design and implement features to fulfill user requirements.
5. Confirming:
   The end-users confirm the story. Features are confirmed through testing environments/ alpha version and the acceptance criteria.

6. Finished:
   The user story is completed at this stage. For new requirements or new features, a new story must be created.

2. What is Scaled Agile Framework (SAFe)?

Need of Scaled Agile:

1. Scalability of agile was difficult for large scale projects.
2. Aligning the entire organization and managing commitments.
3. Unrealistic expectations with respect to the speed of delivery.

Scale Agile Frameworks:

1. Large Scale Scrum (LeSS).

2. Scaled Agile Framework (SAFe)
3. Disciplined Agile Delivery (DAD)
4. Scrum at Scale
5. Nexus Scrum
6. Spotify model

Scaled Agile Framework:

1. The Scaled agile framework or SAFe, consists of a collection of principles, best practices and processes that would enable large organization to adopt agile methodologies.
2. It would help them deliver products and services of high quality.
3. SAFe is best suited for complex projects that involves several large teams at the project, program, and portfolio levels.
4. The current version of SAFe, version 5.0 has several core competencies based on which it can help large organizations.

SAFe Core values:

1. Alignment: It is required to keep up with change, competition and geographically separated teams.
2. Built-in quality: This ensures that every component, as well increment, maintains quality throughout the development cycle.
3. Transparency: Transparency enables teams to rely in each other, build high-performance programs and meet reasonable commitments.
4. Program execution: SAFe's focus is on working systems and business outcomes.

SAFe Competencies:

1. Team and technical agility:
   a. Agile teams are high-performing and cross functional.
   b. Business solutions are built by business and technical teams.
   c. Delighting customers with high quality output.
2. Agile Product deliver:
   a. The customer is at the center of organization's product strategy.
   b. Development is based on cadence, and releases on demand.

   c. Exploring, integrating, deploying and innovating continuously.

3. Enterprise Solution Delivery:
   a. Building big systems with the help of lean.
   b. Co-coordinating and aligning the entire supply chain.
   c. Evolving the live systems continuously.
4. Lean Portfolio Management:
   a. Aligning the strategy, funding and execution.
   b. Optimizing operations along the portfolio.
   c. Decentralized decision making powered by lightweight governance.
5. Organization Agility:
   a. A lean agile mindset is created across the enterprise.
   b. Business operations are leaned out.
   c. Opportunities and threat are addressed quickly.
6. Continuous learning culture:
   a. Everyone learns and grows together.
   b. Exploration and creativity are a very important part of the organization.
   c. The continuous improvement of solutions, services and processes is everyone’s responsibility.
7. Lean agile leadership:
   a. By modeling desired behaviours, inspire others.
   b. Align words, actions and mindsets to lean-agile principles and values.
   c. Lead change and guide others.

Advantages of SAFe:

1. Enable de-centralized decision making.
2. Eases collaboration across cross-functional teams.
3. Ensures decision are made with strategic objectives in mind.

Disadvantages of SAFe:

1. Additional layers of oversight make it resemble the waterfall approach.

2. The top-down approach can limit understanding of the software lifecycle and cause bad planning.
3. Longer planning cycles and roles that are fixed in development cycles.

3. Agile VS Waterfall

Software development lifecycle:

1. SDLC refers to the process of designing, developing and testing high quality software.
2. The aim of the SDLC is to create best quality software keeping in mind the customer’s expectation and deadlines.
3. SDLC defines tasks that have to be performed at each step in the software development process.

SDLC stages:

1. Planning
2. Defining
3. Designing
4. Building
5. Testing
6. Deployment

Waterfall Model:

1. The waterfall methodology is the earliest SDLC approach to software development.
2. Each phase needs to be completed for the beginning of the next phase. The next phase depends on the deliverables of the last phase.
3. It is a breakdown of project activities into several sequential phases.
4. The waterfall model follows the top-down approach, going from gathering requirements to the step of maintenance.

Phases:

1. Requirement gathering
    a. Requirements related to the development of the project are gathered in this phase.
    b. The requirements are documented so that they can be easily accessed in the future.
2. System Design
    a. Requirements gathered in the previous phase are used in this phase for creating the system design.
    b. The design helps in preparing overall system architecture.
3. Implementation
    a. The system is developed in small parts known as units.
    b. This unit are tested for their functionality and they are referred to as unit testing.
4. Integration and Testing
    a. The units developed in the previous phase are tested and then integrated into the system
    b. After integration the entire system in processed for any fault or failure.
5. Deployment
    a. Once the functional and non-functional testing are completed the product is deployed into the customer environment.
6. Maintenance
    a. The issues are fixed and new versions are released for enhancing the performance.

What is Agile Methodology?

1. It promotes continuous development and testing all through the software development life cycle through iterations.
2. The process ensures rapid delivery of all the components of the project that are completely functional.
3. The project is divided into sprints, and these sprints have a defined duration with a list of deliverables, that are planned at the beginning of each sprint.
4. The work can be reviewed by the project team and the customer with the help of daily builds and sprint demos.
5. The entire process has a very high level of customer involvement throughout the project.

6. The agile software development life cycle includes five steps, beginning from concept building to the retirement process.

Phases:

a. Initiation: It refers to the discussion of the project vision and justification of ROI.
b. Planning: The team gets together and begins to identify all the aspects of the project. They tend to find out how to start developing the project.
c. Development: Once the requirements are finalized the work of development begins. A high quality working project is delivered in several sprints.
d. Release: Release or production phase is when the product or projects is deployed and is now being used by end users. The stages are monitored for bugs and defects.
e. Retirement: The last stage or retirement states a newer release of the product after some minor software changes are made or bugs are resolved.

4. Difference

Agile Model	**Waterfall Model**
Agile model is a continuous iteration life cycle model for developing and testing a software.	Waterfall model is a linear sequential life cycle model for developing and testing a software.
Agile model is a flexible way of building a software.	Waterfall model is a rigid, structured way of software development.
Agile model is highly collaborative for yielding better output.	Waterfall model is least collaborative and follows a sequential approach.

In Agile the entire process of development is divided into sprints.	In Waterfall model the software development is broken down into several phase.
In Agile changes may be made even after initial planning.	In Waterfall the development requirement cannot be changed once the development has been started.
In Agile the software development process is a collection of many different projects.	In Waterfall the software development is completed as a single project or deliverable.
In agile testing is performed in same iteration as programming or building.	In waterfall testing phase comes after build phase.
The Agile model strictly focuses on making an easy approach for team interaction and customer satisfaction.	The Waterfall nmodel focuses on the progression of the steps. It has two important elements products and processes.
Agile is good for projects where requirements are expected to change and evolve.	Waterfall is good for projects that have well defined requirements and changes are not expected.
The agile process focuses on customer satisfaction, hence involving them throughout the development phase	The waterfall approach is more of an internal process, hence not requiring the participation of customers.

5. Agile VS Scrum

Agile	**Scrum**
Agile is a set of principles that's iterative and incremental in nature.	Scrum is an implementation of the agile methodology.
Suited for projects involving a small team of experts.	Used in projects where the requirements are constantly changing.

Project head takes carte of all the task and its vital for the project.	There’s no leader. The scrum master and the team addresses the issues. It involves cross functional, self-organizing teams.
Changes cannot be handled frequently.	Changes can be handled frequently.
Requires frequent deliveries to the end users.	With sprints, builds are delivered to client for feedback.
Face to face interaction takes place between cross-functional teams.	Daily stand-up meetings help with collaboration.
The design and execution is simple.	The design and execution can be innovative and experimental.

6. Scrum VS Kanban:

Scrum	**Kanban**
The process is divided into time-constrained iterations.	It is event driven instead of time boxed.
Release takes place after each sprint.	Releases takes place in the form of continuous delivery.
Changes cannot be made during the sprint.	Changes can be made at any time.
Velocity is the default metric for planning and process improvement.	Lead is the default metric for planning and process improvement.
Cross-functional teams are a mandatory requirement.	Specialist teams are required, and cross-functional teams are optional.

Items cannot be added in between an iteration.	New items can be added if there is capacity available.
Has 3 major job roles: Product owner, Scrum master, Scrum team.	Specific job roles haven’t been set up.
The scrum board needs to be reset after each sprint.	The Kanban board stays persistent throughout the project.
It is better suited for long running projects.	Kanban works better for projects that are expected to finish in a short period.