WEEK-1

🡪Introduction to Agile philosophy:

* Agile is an iterative approach to software development that emphasizes flexibility, interactivity, and transparency using small, cross-functional teams
* The Agile Manifesto describes the core values of Agile:

- Individuals and interactions over processes and tools

- Working software over comprehensive documentation

- Customer collaboration over contract negotiation

- Responding to change over following a plan

* The waterfall approach is a structured, step-by-step process that can lead to problems not surfaced until later in development
* Extreme Programming (XP) advocated an iterative approach that valued simplicity, communication, feedback, respect, and courage
* A Kanban system is characterized by visualizing workflow, limiting work in progress, managing and enhancing flow, making process policies explicit, and continuously improving a process
* Working in small batches means delivering something useful quickly
* An MVP is the cheapest/easiest thing you can do to test a hypothesis and learn
* Behavior Driven Development makes sure you are building the right thing
* Test Driven Development makes sure you are building the thing right
* Pair programming enables you to discover defects earlier and increase your code quality
* The Gherkin syntax, which comes from a company called Cucumber, is a single syntax that developers and stakeholders can understand: given some set of preconditions, when an event occurs, then some outcome is observed.

🡪Intro to Scrum methodology:

* Scrum is a methodology that follows the Agile philosophy
* The Scrum management framework:

- Provides structure through defined roles, meetings, rules, and artifacts

- Prescribes small, cross-functional, self-organizing teams

- Uses fixed-length iterations called sprints

- Produces a potentially shippable product increment with every iteration

* The product owner represents stakeholders, articulates the product vision, and decides priorities, requirements, and readiness to ship
* The scrum master coaches the team, promotes a cooperative environment, shields the team from interference, and unblocks impediments
* The scrum team is small, dedicated, co-located, cross-functional, and self-managing
* The scrum team negotiates commitments with the product owner — one sprint at a time
* The scrum team has autonomy regarding how to reach commitments
* Scrum produces a product backlog, a sprint backlog, and a completed — or done — increment
* The Scrum events are sprint planning, daily Scrum meetings, sprints, sprint reviews, and sprint retrospectives

If done well, Scrum can:

* Increase employee productivity and happiness
* Improve product quality
* Reduce time to market
* Enhance stakeholder satisfaction

🡪Organising success:

* How you are organized can affect the systems that you build.
* Giving teams autonomy leads to motivated teams who can execute faster and build better systems
* Not adopting Agile across your organization can lead to operational bottlenecks
* Many companies adhere to their waterfall planning and call it Agile
* Simply doing iterative development is not Agile unless you are being responsive to changes and delivering value often.

WEEK-2

🡪Planning to Agile:

* Planning everything at the beginning of a project can lead to missed deadlines
* Iterative planning allows for course corrections and more accurate estimates
* Placing existing people in new roles without the proper training will lead to failure because Agile roles need a different focus and new priorities
* To be successful, the Agile mindset must be adopted by user management
* A kanban board is a way of tracking plan items needing to be done, items in process, and completed items
* A kanban board is made of multiple pipelines
* Work moves from left to right on the kanban board as it is completed.

🡪User-stories:

* A user story documents a person requesting a function to achieve a goal.
* Using a template helps ensure that stories are complete.
* Defining "done" helps minimize misunderstandings.
* Use the INVEST acronym to remember the qualities of a good user story: independent, negotiable, valuable, estimable, small, and testable.
* Epics can be used to capture big ideas.
* Story points are a metric used to estimate the difficulty of implementing a given user story.
* Story points are relative, like T-Shirt sizes.
* You must agree on what "average" means.
* You should never equate story points with wall-clock time.
* A product backlog is a ranked list of all unimplemented stories.
* Stories high in the ranking should have more detail than those that are lower.
* Create stories using the "As a", "I need", "So that" template to ensure everyone understands who it benefits and the business value it provides

🡪The planning process:

* It is the product owner's responsibility to maintain a groomed backlog
* Backlog refinement is used to order the product backlog and make stories sprint ready
* You start refinement by triaging new issues
* Large stories should be broken down until they are small enough to fit in a sprint
* The goal of backlog refinement is to get the backlog ready for the sprint planning meeting
* It is the product owner's responsibility to present the sprint goal
* It is the development team’s responsibility to create a sprint plan
* A sprint plan is created by moving stories from the product backlog into the sprint backlog until the team's velocity is reached

WEEK-3

🡪Executing the plan:

* You need to keep the kanban board updated so that everyone knows what you are working on
* It is important to always work on the story with the highest priority that you have skills for
* Working on more than one story at a time may lead to neither story being finished at the end of the sprint
* The daily stand-up occurs every day for 15 minutes
* Topics not related to the stand-up should be addressed after the meeting
* Each person should be prepared to answer the three stand-up questions:

- What did I accomplish the previous day?

- What will I work on today?

- What blockers or impediments are in my way?

🡪Completing the sprint:

* A burndown chart shows the measurement of story points completed vs story points remaining for a sprint
* Burndown charts can be used to show progress and forecast the team's probability of achieving the sprint goal
* A sprint review is a demonstration of the features that have been implemented during the sprint
* Feedback from stakeholders is critical to help shape the future of the product
* The backlog is updated based on feedback
* A sprint retrospective is a time to reflect on how the sprint went
* The sprint retrospective is attended by the scrum master and the development team
* The team must feel comfortable to speak freely
* A sprint retrospective must result in changes to improve the next sprint
* Three questions are answered on what went right or wrong:

- What went well? (keep doing)

- What did not go well? (stop doing)

- What should we change for the next sprint?

use🡪Measuring Success:

* High performing teams use metrics to continually improve
* It is important to be sure that the metrics you use are actionable
* You should take a baseline before trying to measure change.
* It is important to give developers credit for unfinished stories
* Unfinished stories should be split into new stories to complete the work in the next sprint
* Each sprint milestone should be closed when completed to reflect the velocity of the sprint
* You should create a new milestone for the next sprint
* The top 4 actionable metrics can be used to improve your team’s overall performance

- Mean Lead Time (How long does it take from idea to production?)

- Release Frequency (How often can you deliver changes?)

- Change Failure Rate (How often do changes fail?)

- Mean Time to Recovery (MTTR) (How quickly can you recover from failure?)

* Anti-patterns to avoid when practicing Scrum include:

- No real product owner

-Teams are too large

-Teams are not dedicated

-Teams are too geographically distributed

-Teams are siloed

-Teams are not self-managing

* The Scrum health check provides guidelines for measuring a healthy scrum team