Module 3:

Team Organization and workflow Management (Scrum methodology)

Agile software Development Methodology

- **Software Engineering is now a Team Sport**
    - Now is Post-superhero Programmer where a programmer can go into his room or workspace and after 6months or more come up with a large complex working software.
    - Rising bar of functionality/quality => cannot do SW breakthrough alone
    - Successful Software (SW) career => programming chops AND plays well with others AND can help make team win".

> "***There are no winners on a losing team, and no losers on a winning team.***" Fred Brooks, Jr

- **How then do we organize High performance software engineering teams?**
    - **Agile Development is the Answer**
        - **What Agile Development is**
        - **Agile Manifesto 2001**

"We are uncovering better ways of developing SW by doing it and helping others do it. Through this work we have come to value" Authors of Agile methodology

- Individuals and interactions over processes & tools"
- Working software over comprehensive documentation"
- Customer collaboration over contract negotiation"
- Responding to change over following a plan"

That is, while there is value in the items on the right, we value the items on the left more."

1. Agile Life Cycle

- Embraces change as a fact of life: continuous improvement vs. phases"
- Developers continuously refine working but incomplete prototype until customers happy, with customer feedback on each Iteration (every ~1 to 2 weeks) "
- Agile emphasizes Test-Driven Development (TDD) to reduce mistakes, written down User Stories to validate customer requirements, Velocity to measure progress"

2. Examples of Agile Methodologies

Different methods following Agile principles have originated within the last two decades. They include:

- Scrum Methodology
- Kanban Methodology
- eXtreme Programming (XP) Methodology
- Crystal Methodology
- Dynamic Systems Development Method (DSDM)
- Feature Driven Development (FDD) Methodology
- Among them, Scrum is arguably the most popular and extended one.

Among them, Scrum is arguably the most popular and extended one.

3. SCRUM Methodology

- Scrum is an agile way to manage a software development project.
- It was proposed by Jeff Sutherland and Ken Schwaber in 1995 for an effective agile software development.
- In SCRUM methodology, the software project is divided into sprints - which are time-bound cycles that typically last between 2 and 4 weeks, and which are repeated over and over until the project is finalized
- The Scrum method in its original form prescribes a Sprint length of 30 calendar days and requires the Team to meet every day for a 15-minutes Daily Scrum meeting. However, because of scheduling problems and other course commitments it was impossible to expect students to work on a project every day. We resolved this problem by prescribing two Daily Scrum meetings per week: in the presence of SCRUM Master and Product Owner.

- In Agile Methodology, “Zero Sprint” refers to the first step that comes before the first sprint. So, it is more like a pre-step to the first sprint. Thus, Zero Sprint would include a host of activities that are to be completed before starting a project, including setting up the development environment, preparing Product Backlog (requirements not prioritized), Sprint Backlog and other such tasks that are usually done before beginning the actual development process.
- Sprint retrospective meeting to discuss what did and did not go on well and how to ameliorate the process.

4.1 Scrum: Team Organization

• “2 Pizza” team size (4 to 9 people)"
• “Scrum” inspired by frequent short meetings"
– 15 minutes every day at same place and time"

4.2 Scrum: Daily Scrum Agenda

o Answers 3 questions at “**daily scrums**”:"

1. What have you done since yesterday?"

2. What are you planning to do today?"
3. Are there any impediments or stumbling blocks?"

- Help individuals by identify what they need

4.3 Scrum: roles

- Team: 2-pizza size team that delivers SW"
- ScrumMaster: team member who "
  – Acts as buffer between the Team and external distractions"
  – Keeps team focused on task at hand"
  – Enforces team rules (coding standard)"
  – Removes impediments that prevent team from making progress"
- Product Owner: A team member (not the ScrumMaster) who represents the voice of the customer and prioritizes user stories"

4.4 Scrum: Resolving Conflicts

• eg. Different view on right technical direction"

✓ **1st list all items on which both sides agree "**
– vs. starting with list of disagreements"
– Discover closer together than they realize?"

✓ **Each side articulates the other's arguments, even if don't agree with some"**
– Avoids confusion about terms or assumptions, which may be real cause of conflict

✓ **Constructive confrontation (Intel)"**
– If you have a strong opinion that a person is proposing the wrong thing technically, you are obligated to bring it up, even to your bosses"

✓ **Disagree and commit (Intel)"**
– Once decision made, everyone needs to embrace it and move ahead "
– "I disagree, but I am going to help even if I don't agree."

4.5 Scrum: Workflow management

➢ At the beginning of the project, the roles are assigned and the team is assembled.

➢ A meeting is held between the **Product Owner, the user, the Scrum Master and the Development team** to extract the project Product backlog.

- The Project **Product Backlog** is a list of User Stories. User Stories are self-contained entities which define a functionality required by the customer for the project or the service. They can be grouped into a component which define a higher-level functionality of the system that cannot be delivered in a single sprint, and which are normally divided into smaller User Stories.
- For every User Story, a series of Acceptance Criteria are agreed between the customer (user) and the Product Owner (occasionally they might also be written between developers and Product Owners to increase transparency).

➢ A Sprint kick-off meeting is held between the **Product Owner, the Scrum Master and the Development team.** During the Sprint kick-off meeting the team decides which User Stories will be implemented during the sprint. The selected User Stories compose the **Sprint Backlog.**

4.6 Scrum: Product Backlog and Sprint

- ***Product Backlog***

ID	*Requirement (Description or User Stories)*	*Acceptance Criteria*	*Priority*	*Initial Estimate (in hours)*	*Adjustment Factor*	*Adjustment estimate (in hours)*
1	As a **user** of the system, I should be able to **create an account** in order to **have access to the system functionalities**	When a user of the system creates an account, they are provided with an interface with various options that enables them perform functionalities based on their role in the system	1	8	1.5	12
2	As a **fire fighter unit**, I should be able to **get notifications of a fire disaster** in order **to be able to respond**	When a fire fighter logs into system, they can receive emergency fire request/Notification	2	15	1	15

- **User stories**
  - Who ?
  - What?
  - Why?

As a [type of user], I want [an action] so that [a benefit/a value]

Acceptance criteria are a set of functionalities that a product must met in order to be accepted by the user.

- Helps you do verification and validation

Properties of a user story

- testable
- Clear and concise
- Easy to understand
- Provide a customer's perspective

How to write an Acceptance Criteria

- ➢ Scenario(explain the scenario)
- ➢ Given (how things begin)
- ➢ When(action taken)
- ➢ then (outcome of action)

Priorities and estimation of priority techniques

- **T-shirt: work is categorized according to the size of a T-shirt:**
  - Xs(extra-small)
  - S (small)
  - M(medium)
  - L (large)
  - Xl (Extra Large)

Effort Estimation

We are good at comparing things together rather than absolute estimation
Fibonacci sequence: 1 2 3 5 13
Its how nature lays itself out, curve of brain, shape of petals of flowers
It follows a pattern called the golden ratio
This number are far enough for us to easily see the difference
Using this way enables us to collect opinion on the size of a task as a group.

- **Planning poker**
  - use cards
  - pick user story
  - team votes by picking on the cards

    if they are 2 cards apart then we take the cards and just average them and move to the next user stories

    if they are more than 2 cards apart, the high and the low cards will discuss with the team why they selected their given value. The team will need to revote again
  - discuss
  - vote again

NB: They should not be an expert estimator for the team. But rather the entire team that will be doing the work should be responsible for the estimates, because they are the ones who know themselves

- ***Sprint Backlog***

Release	***Sprint***	***ID of User Stories***	***Period***
Release 1: REST API Gateway	Sprint 1	1,2,4	10th March – 5th April
	Sprint 2	3,5	6th April – 30th April
Release 2: Mobile App	Sprint 3	6,8	1st May – 21st May
	Sprint 4	7,9	24th May – 15th June

Concepts:

- Pair programming
- TDD (Test Driven Development) and BDD(Behavior Driven Design)
- Version control

Scrum Summary

- Basically, self-organizing small team with daily short standup meetings"
- Work in “sprints” of 2-4 weeks"
- Suggest members rotate through roles (especially Product Owner) each iteration