



# Agile & Scrum

## Practices

Duration: 15 minutes

Q&A: 5 minutes by the end of the lecture

# What is Agile ?

A **group** of software development methods in which **requirements** and **solutions evolve** through collaboration between **self-organizing, cross-functional** teams<sup>1</sup>

# The Agile Manifesto

- **Individuals and interactions**
  - over Processes and tools
- **Working software**
  - over Comprehensive documentation
- **Customer collaboration**
  - over Contract negotiation
- **Responding to change**
  - over Following a plan

# What is an Agile Software Practice?

The practice centers around sprints, that are time-boxed periods of experimentation and development. While a sprint is in progress, the team goals remain fixed.

At the beginning of each sprint, the team re-evaluates objectives and sets sprint goals which account for any new changes in customer needs or product expectations.

# Scrum

Scrum is a framework for Agile teams. It comes with a set of roles, processes, ceremonies, and tools that enable teams to collaborate in a fast-paced iterative development cycle.

# Scrum

1. **Scrum roles**
2. The Sprint
3. Scrum artifacts

# Scrum Roles

1. The Product Owner
2. The Scrum Master
3. The Team Member

# Scrum Roles

## 1. The Product Owner

- a. **Keeper of the product vision**
- b. Represents "business" & end users
- c. Owns the product backlog
- d. Prioritizes tasks



# Scrum Roles

## 1. The Product Owner

- a. Keeper of the product vision
- b. Represents “business” & end users**
- c. Owns the product backlog
- d. Prioritizes tasks

# Scrum Roles

## 1. The Product Owner

- a. Keeper of the product vision
- b. Represents "business" & end users
- c. Owns the product backlog**
- d. Prioritizes tasks

# The Product Backlog

Deliverables	Status	Owner	Points
<b>Product Backlog</b>			
Homepage: autocomplete / connect to journeyview	Not Started		3
Get autocomplete in navbar on journeyview	Not Started		5
Refactor json importer to change dates to integers	Not Started		2
JSONImporter: Deal with duplicates	Not Started		2
Pass headline back	Not Started		3
JourneyView: styling	Not Started		3
JourneyView	Not Started		5
- create path via client router to profile view	Not Started		1
- change to horizontal bar charts	Not Started		2
- make bars clickable	Not Started		5
- upon click, get profiles info (including id, name, img url)	Not Started		
- serverside: refactor getStats to return an array of objects	Not Started		5
- front-end: refactor existing chart-drawing code to accept array of objects instead of number	Not Started		2
- Profile thumbnails appear next to bar charts	Not Started		
- create thumbnail models & collection	Not Started		2
- create profile thumbnail views and collection	Not Started		2
- render thumbnail collection view to journey view	Not Started		3
- upon click of profile thumbnail, take user to profile view using profile ID	Not Started		2
Profile View			
- create Profile Model	Done	Derek	2
- Make get request to return all info related to the profile	Done	Derek	1

# The Product Backlog

We use a number of tools within Scrum - the Product Backlog is the first we'll look at

- List of desired deliverables for the product.
- Includes: features, bug fixes, documentation changes, anything else
- Ever changing list that is tended to regularly by the Product Owner
- Items at the top of the list are closer to production and are much smaller and more well defined, further down they might be less defined and larger, at the bottom you'll find 'nice to have' features. Advantage: effort is never wasted on writing specs on features that never get developed.
- Strict priority - not High, Medium, Low - but priority by strict ordering - think single-file line of stories waiting to be worked on.
- Can exist as a wall, spreadsheet or anything the PO and team decide on.

# Scrum Roles

## 1. The Product Owner

- a. Keeper of the product vision
- b. Represents "business" & end users
- c. Owns the product backlog
- d. Prioritizes tasks**

# The Sprint Backlog

Deliverables	Status	Owner	Points
<b>Sprint #7</b>	<b>Wednesday, June 1</b>	<b>Points Done:</b>	<b>18</b>
Homepage: autocomplete	In Progress	John	3
JourneyView styling	In Progress	David	3
Frontend: comment code	Not Started	Melanie	1
Serverside: Logic for bucketing degrees	Done	David	3
Server: comment code	Done	Melanie	1
Total			25
<b>Product Backlog</b>			
Homepage: autocomplete	Not Started		3
Get autocomplete in navbar on journeyview	Not Started		5
Refactor json importer to change dates	Not Started		2
JSONImporter: Deal with duplicates	Not Started		3
Pass headline back	Not Started		1

# Scrum Roles

## 2. The Scrum Master

- a. **Scrum expert and advisor**
- b. Facilitator & Coach
- c. Impediment bulldozer
- d. Not the boss!

# Scrum Roles

## 2. The Scrum Master

- a. Scrum expert and advisor
- b. Facilitator & Coach**
- c. Impediment bulldozer
- d. Not the boss!



# Scrum Roles

## 2. The Scrum Master

- a. Scrum expert and advisor
- b. Facilitator & Coach
- c. Impediment bulldozer**
- d. Not the boss!

# Scrum Roles

## 2. The Scrum Master

- a. Scrum expert and advisor
- b. Facilitator & Coach
- c. Impediment bulldozer
- d. Not the boss!**

# Scrum Roles

## 3. The Team Member

- a. **Active Scrum team member**
- b. Does all of the development work
- c. Self-organizes to deliver user stories
- d. Owns the estimation process
- e. Owns the 'how to do the work' decisions
- f. Avoids 'not my job'

# Scrum Roles

## 3. The Team Member

- a. Active Scrum team member
- b. Does all of the development work**
- c. Self-organizes to deliver user stories
- d. Owns the estimation process
- e. Owns the 'how to do the work' decisions
- f. Avoids 'not my job'

# Scrum Roles

## 3. The Team Member

- a. Active Scrum team member
- b. Does all of the development work
- c. Self-organizes to deliver user stories**
- d. Owns the estimation process
- e. Owns the 'how to do the work' decisions
- f. Avoids 'not my job'

# Scrum Roles

## 3. The Team Member

- a. Active Scrum team member
- b. Does all of the development work
- c. Self-organizes to deliver user stories
- d. Owns the estimation process**
- e. Owns the 'how to do the work' decisions
- f. Avoids 'not my job'

# Scrum Roles

## 3. The Team Member

- a. Active Scrum team member
- b. Does all of the development work
- c. Self-organizes to deliver user stories
- d. Owns the estimation process
- e. Owns the 'how to do the work' decisions**
- f. Avoids 'not my job'

# Scrum Roles

## 3. The Team Member

- a. Active Scrum team member
- b. Does all of the development work
- c. Self-organizes to deliver user stories
- d. Owns the estimation process
- e. Owns the 'how to do the work' decisions
- f. Avoids 'not my job'**



# Scrum

1. Scrum roles
2. **The Sprint**
3. Scrum artifacts

# The Sprint

## Use 2 day sprints

### 1. 1st Sprint

- a. Monday, Tuesday

### 2. 2nd Sprint

- a. Wednesday, Thursday

### 3. 3rd Sprint

- a. Friday, Saturday

# Sprints

## Plan your work in Two-day sprints

Day 1	Day 2
<i>Morning</i> <b>Sprint Planning</b> 30 minutes	

## Activity: Sprint Planning

A meeting at the start of each sprint where the team:

- Prioritizes the backlog
- Evaluates tasks to ensure specificity
- Estimates tasks near the top of the list
- Picks tasks to commit to doing this sprint

Try and include at least one process improvement in each sprint.

# Sprints

## Plan your work in Two-day sprints

Day 1	Day 2
<i>Morning</i> <b>Sprint Planning</b> 30 minutes	
<i>Afternoon/Evening</i> <b>Standup</b> 15 minutes	

# Sprints

## Plan your work in Two-day sprints

Day 1	Day 2
<i>Morning</i> <b>Sprint Planning</b> 30 minutes	<i>Morning</i> <b>Standup</b> 15 minutes
<i>Afternoon/Evening</i> <b>Standup</b> 15 minutes	

## Activity: Standups

Standups are a brief periodic meeting to ensure every team member is aware of what is happening. Answer these questions:

- What did I do ? (What did I complete ?)
- What will I be doing next ?
- What am I blocked on ?

## Activity: Standups

Standups must be brief (<10-15 minutes) usually done standing up.

Never use this time to solve problems. Identify problems, then break out into pairs to come up with a plan to solve blockages.



# Sprints

## Plan your work in Two-day sprints

Day 1	Day 2
<i>Morning</i> <b>Sprint Planning</b> 30 minutes	<i>Morning</i> <b>Standup</b> 15 minutes
<i>Afternoon/Evening</i> <b>Standup</b> 15 minutes	<i>Afternoon/Evening</i> <b>Sprint Reflection</b> 15 minutes

## Activity: Retrospectives

- At the retrospective, discuss why something wasn't accomplished. You will often find that:
  - *Tasks were too vaguely described*
  - *Your commitment was too aggressive*
- Discuss the reasons with your team. Adjust accordingly.
- Retrospectives are not the time to fix problems. This is dedicated time for the team can discuss issues without judgment.

# Scrum

1. Scrum roles
2. The Sprint
3. **Scrum artifacts**

# Scrum Artifacts

1. User Stories
2. The Product Backlog
3. The Sprint Backlog
4. Task Board

## User Stories

It's helpful to describe your new tasks and issues from the perspective of the user. We call these **user stories**. They help us identify the distinct value that each task adds to our product, which helps us to prioritize our work.

User stories follow a common format:

**"As a *kind of user*, I want something  
so that *motivation for that thing*."**

## User Stories: Examples

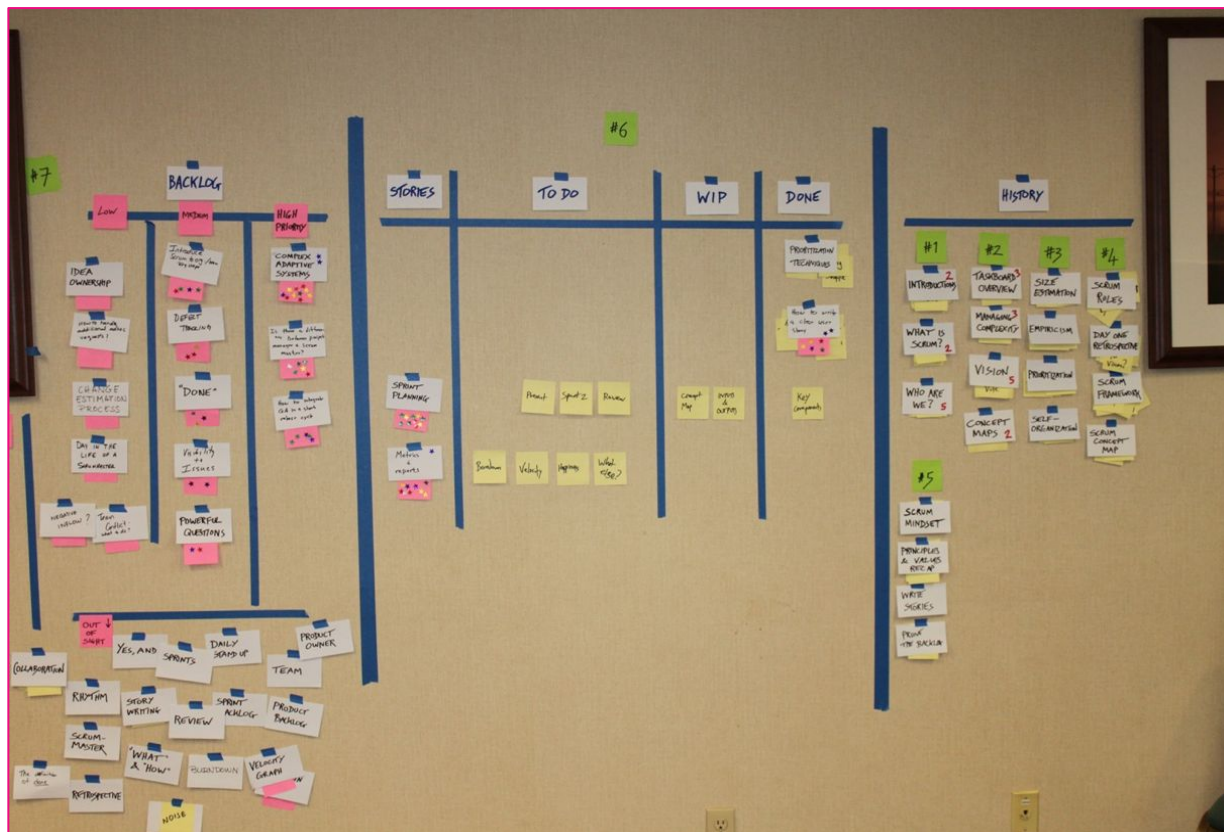
- As a manager, I want to see my employee's calendars so I can know when they're available for meetings.
- As an student, I want a map showing where my classes are so I know where to go on the first day of school.
- As a moderator, I want a list of user-flagged posts so I can ban all the trolls.
- As an admin, when I edit this user, I want to be able to reset their password.

# The Backlog

The Backlog is a prioritized list of all things you need to do in this project.

The top of the Backlog is often referred to as the Sprint Backlog. It represents what you are doing now (the current sprint).

# Task Board





## Task Board



# Ticket Types

Make use of at least the following 4 types of tickets:

- User Stories
- Bug Tickets
- Implementation Tickets (features)
- Chore Tickets (technical debt, refactor, etc)
- Process Ticket (optional)

## Ticket Types

By putting your tasks into different ticket types, the team gains visibility into how they are spending their time.

Aim for a balance of features, bugs, chores and optionally process improvements.

# Task Estimation Process

During sprint planning, estimate the complexity of each task by:

- Assigning a numerical value to each task which represents its complexity
- Use Fibonacci Sequence (1, 2, 3, 5, 8) or a linear sequence (1,2,3,4)

## Task Estimation Process- Why?

- Holding yourself accountable to commitments is the number one way to get better at being an engineer.
- Each sprint is an opportunity to hone your skills of task specification, estimation and knowing what you are committing yourself to.

A solid pink horizontal bar with rounded ends, located in the top right corner of the slide.

# That's it