Learn To Code

Agile, User Stories, and Agile Project Management

Student Workbook #1-C

Version 3.0 Y

Table of Contents

Module 1

Agile Software Development

Section 1–1

Agile Software Development

Agile

- **Building software is complex and not just because writing the code is difficult!**
    - Keeping track of changing priorities, who is working on what, and what tasks need to be completed before others can be started is difficult

- **Software developers and product managers have come up with a number of excellent tools to make it easier to plan projects to keep everyone on the same page**

- **Agile is a structured and iterative approach to project management and product development**

- **With an Agile approach, we focus on being open to change during product development and respond to that change without going off the rails**

Waterfall: The Bad Way!

- **The term Waterfall was coined as an example of what not to do.**
  - Yet, a lot of companies do just this
- **In Waterfall, project management looks like this:**
  - Everything is planned upfront
  - All features are developed from the start
  - All features are tested at the end
  - The product is shipped to the customer
- **But what happens when:**
  - You have a poorly articulated vision of the end product
  - Project requirements changes during development?
- **It's often a *long* time before stake holders see anything that works**

Agile Philosophy : The Manifesto

- **The *Agile Manifesto* is a document that defines the core values behind the Agile philosophy**
  - Its goal is to help development teams work more efficiently and sustainably
- **As a class, let's go look at the Agile Manifesto**
  - `https://agilemanifesto.org/`
- **Discussion: What do these mean values to you?**

Agile Philosophy : The 12 Principles

- **Agile's "founders" came up with the Agile Manifesto be defining 12 principles that would help organizations be more flexible, responsive, and adaptive to changes**
- **As a class, let's go look at the 12 Agile Principles**
  - `https://agilemanifesto.org/principles.html`
- **Discussion: What do you think these help software development teams?**

The Scrum Framework

- **Scrum is framework that helps teams work together**

- **It encourages teams to:**
  - self-organize while working on a problem
  - learn through their experiences
  - reflect on their successes and failures in order to continuously improve

- **Scrum team members belong to one of three roles:**
  - product owner
  - scrum master
  - development team member

- **Scrum is just one of the many iterative and incremental Agile software development frameworks which are used widely in software development**
  - However, it is very popular and will be examined here

Sprints

- **Scrum relies on delivering working software in short cycles called Sprints**
  - The use of sprints is at the very heart of Scrum and other Agile methodologies

- **A sprint is a short period during which a scrum team works to complete a specific amount of work**
  - Often, sprints are 2-3 weeks long

- **Sprints enable a team to:**
  - Get fast feedback
  - Continuously improvement
  - Rapidly adaptation to change
  - Accelerate delivery

- **Sprint planning is important! You must decide on:**
  - what is the sprint goal
  - how long the sprint is will last
  - where you team will start

The Daily Standup

- **When we talk about Agile, we are referring to a methodology to plan and complete projects**
  - In order to do this, there are a number of rituals teams participate in to help the Agile process keep moving
  - They include the ***daily standup*** and the ***retrospective***
- **During the sprint, team members meet daily at the *daily scrum* or *standup***
  - It is an important part of the day!
- **It is here team members to talk about:**
  - their successes
  - the issues they face
  - their blockers
- **They may reach out and ask for help**
  - Or help might be offered
- **The scrum master usually facilitates the daily scrum**
  - However, it is the development team's responsibility to run the standup

- **Questions are often:**
  - What did you accomplish yesterday?
  - What do you plan to accomplish today?
  - Where are your blockers?
- **It is very important that team members be comfortable surfacing the problems they are encountering in order for the process to work**

Retrospectives

- **It's important to reflect back on what you've done in order to better prepare for the future**
  - *Retrospectives* are one way to do this reflection
- **A retrospective usually happens at the end of a sprint or large project**
- **Some of the common forms they take include:**
  - Start/Stop/Continue
  - Happy/Sad/Confused
  - Positives/Deltas
- **No matter how you do your retrospective, the goal is to hear from multiple people in the group and derive action items for your next sprint or project**

Research: Learn More

- **Take a little time to get a better sense of what the Scrum implementation of an Agile process looks like**
  - Scrum Alliance: Why Scrum?
    - `https://www.scrumalliance.org/about-scrum`
  - Scrum Roles
    - `https://www.agile42.com/en/agile-community/agile-info-center/scrum/scrum-roles`
  - Scrum Glossary
    - `https://www.scrum.org/Resources/Scrum-Glossary`

Module 2

User Stories

Section 2–1

User Stories

User Stories

- **If we have an idea that's going to change the world, how do we get started?**
- **If we want millions of users to love our new social media platform, how do we get there?**
- **Tackling large problems is impossible without a plan**
  - We need to break apart large project into smaller, manageable tasks
  - We do this by creating User Stories
- **The goal of User Stories is to focus on features with the user in mind**
  - Being able to write concise user stories will make projects easier to accomplish and make collaboration with multiple developers easier

What is a User Story?

- **What is a user story?**

  Format

  ```
  - (role) can (action) -

  ex: As a (role) I want (something) so that (benefit)
  ```

  Example

  As a ***web site user*** I want a ***navigation bar*** so that ***I can switch between pages***

- **Recommended Reading and Examples:**
  - Mountain Goat Software - User Stories
    - https://www.mountaingoatsoftware.com/agile/user-stories
  - GSA
    - https://tech.gsa.gov/guides/user_story_example/

Prioritizing User Stories

- **Before we start to work on a project during a sprint, we want to prioritize our stories**

- **To help prioritize our stories, we may think about the following questions:**
  - What is time-sensitive and needs to happen now?
  - What will add the most value to the project and get it closer to a shippable product?
  - What work can happen in parallel?
  - What work depends upon another story being finished?

- **There is no "right" way to prioritize your tasks but the goal should be to get a *minimum viable product* (MVP) as soon as possible**
  - The MVP is a version of a product with just enough features to be usable by early customers
  - These customers then provide feedback that is used for future enhancements

Module 3

Agile Project Management

Section 3–1

Agile Project Management Tooling

Tooling

- **Once you have a number of stories, you need some way to track:**
  - who is working on what
  - what has yet to be completed
  - what has already been done

- **There are a number of tools that will help you manage these user stories, including:**
  - Asana
  - Trello
  - GitHub Projects

Issue Tracking

- **These tools all help us with something called *issue tracking***
  - Issue tracking is really just a specialized to-do list.

- **This to-do list has an extra layer supporting a review process for the task**

- **Instead of a task just being completed, you may have several states indicating where the task is in the workflow**

- **Usually, the last step in the workflow indicates that the finished work has been approved**
  - This helps prevent problems from slipping through the cracks by ensuring that there are at least two sets of eyes on every resolution

Coming Soon

- **Discussion of GitHub project boards**
- **Examples of capstone user stories**