Team Project

Each team will be using the agile method scrum to develop a calculator. The instructor will act as the Product Owner. Each team should attempt to follow the agile manifesto and/or scrum as closely as possible. Sprints will start and end on Mondays. We will use class time to do sprint reviews, sprint retrospectives, sprint planning and daily scrum. Code will be submitted via github. The calculator can be developed in the team's desired programming language (i.e. java, python, C#). Teams will use ZenHub for issue/story tracking. In class time will be used to perform sprint meetings. A majority of sprint execution (i.e. coding) will be done outside of class but feel free to do this work in class if time persists. Class time will be used as follows:

Monday

- "daily scrum"
- Sprint review
    - each team will demo any progress they have made and talk about what they accomplished
- Sprint Retrospective
    - Teams will answer the following questions
        - What went well?
        - What could have gone better?
        - What do we want to try next?
        - What puzzles us?
- Sprint planning
    - The Scrum team will review what items will be completed with the product owner

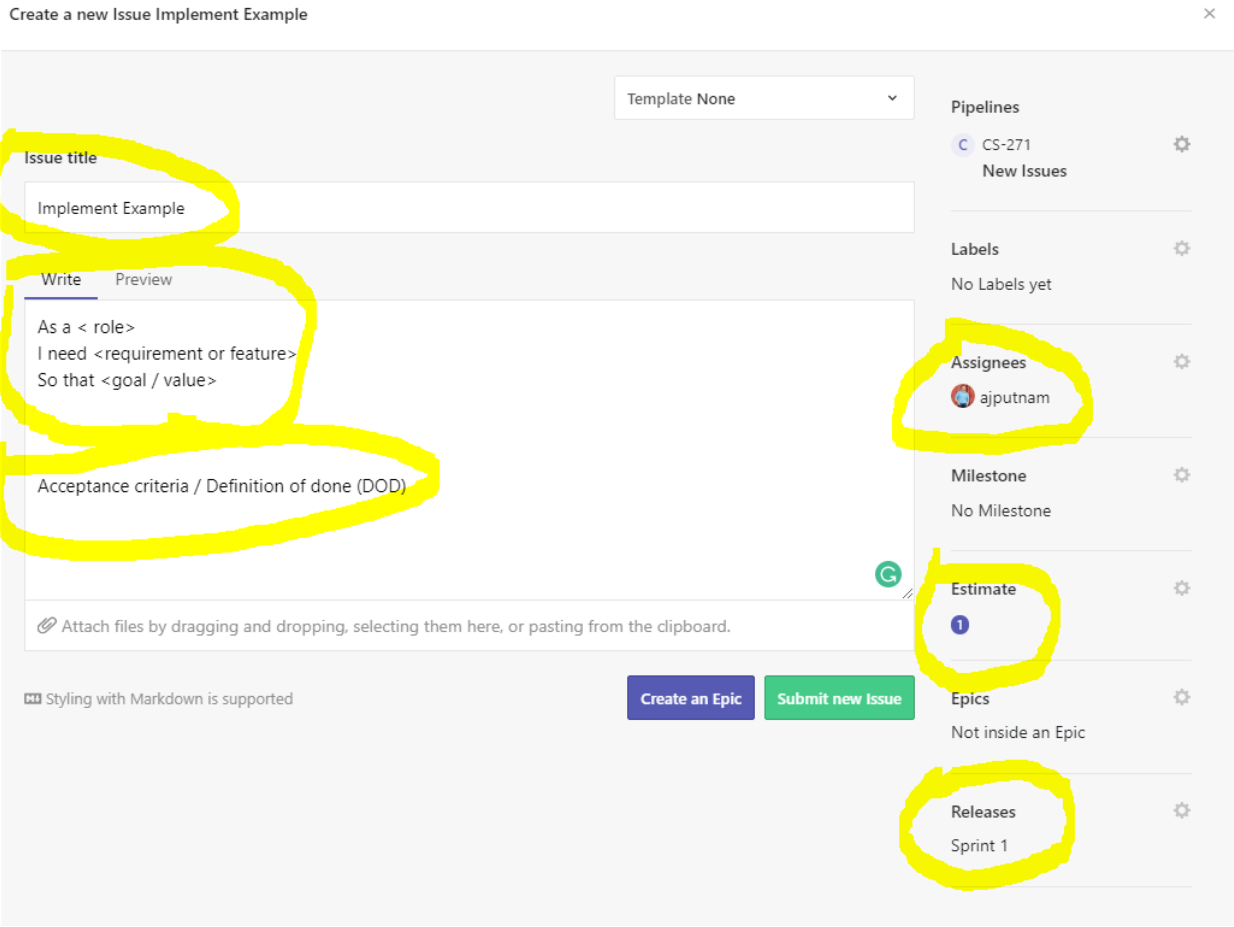
Tuesday

- "daily scrum"
- Backlog grooming
    - Switch estimation games each week.
- Sprint execution
- Product Owner questions

Sprint Schedule:

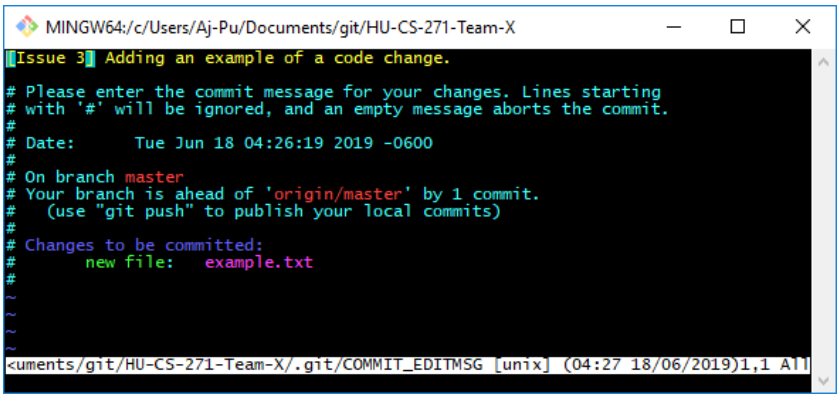
Sprint 1	17th – 24th
Sprint 2 (defect fix)	24th – 1st
Sprint 3	1st – 8th

Creating an Issue (yellow sections are required)



Git commit example.

Each git commit should be associated with a task in github/zenhub. The commit message should start with [Issue #] followed by a description of the change. If the task requires multiple code changes/pushes reuse the same issue.



Project Backlog:

- Get the results from a mathematical expression (i.e. 1+5/4)
- Support: add, subtract, multiply, divide, exponential, factorial
- PEMDAS support
- Support the following functions
    - Factor(n)
    - Sqrt(n)
- Invalid user input (error handling)
- Matrix Math (add, subtract, multiply)
- Fractional support (1/3 + 1/3 = 2/3)

Project requirements

- Each story/feature should have at least 1 test associated with it
- Each team member will be required to create and complete at least 1 **task** and code change per sprint
-

Sprint Rubric

Creating stories from the "backlog". Breaking stories down into tasks and assigning them to a team member and "committing" to them	25
Submitting code with proper git commit message and tagging.	25
Attending and participating in team meetings	25
Overall team performance, code quality, tests and "product"	25