

**Introduction**

This guide contains the process flow for Team Gold to navigate the SCRUM framework during the final project. It will ensure success in delivering value for the customer and keeping the team on track to complete this project in a timely manner. Should any process change, this guide will be updated so that all internal stakeholders are continuously on the same page and informed of all current processes. This guide is meant to provide clarity on all SCRUM processes as defined by Team Gold.

**Sprints & Definition of Done**

The timebox of each sprint will be one week in duration. Sprints will start on Mondays and conclude on Sundays. Subsequent sprints will begin immediately at the end of the previous sprint. Sprint tasks will be assigned at the start of each sprint, with the exception of new tasks which may arise throughout the current sprint. Should there be any tasks not completed during the initially assigned sprint, it will roll over to the next one. However, every effort should be made to ensure timely completion of all tasks.

Team Gold will meet with Instructor Babb each Monday at 4:30 pm CST. After each meeting, Team Gold will then hold a Sprint Planning session to agree upon a plan of execution for any new tasks that are assigned during the instructor meeting. During the Spring Planning session, the product backlog will also be reviewed and the goal for the sprint will be finalized. In addition, all tasks will be estimated at this time.

Team Gold’s “Definition of Done” will include the following checklist items as outlined in figure 4.1 in Rubin’s Essential Scrum Guide:

Table

Description automatically generated

**Requirements and User Stories**

Epics, features, and user stories will be a part of the Story Mapping and documented within Azure DevOps. User/sprintable stories will capture user needs, include acceptance criteria/conditions, and fall underneath each feature in Azure. Features will capture higher-level user needs and fall under larger Epics. Sprintable stories will be more granular yet display multiple levels of abstraction in relation to each feature. Stories should follow Wake’s INVEST criteria and be Independent, Negotiable, Valuable, Estimatable, Size Appropriate, and Testable. In terms of format, stories will include the user role, what the user wants to achieve, and why the user wants to achieve this goal.

Both non-functional and functional requirements will be a part of the functional requirements gathering process. Requirements must be detailed and thoughtful; however they will be considered a living document and can change throughout the development process. Any changes must be documented accordingly.

**Product Backlog**

* PBI’s and why
  + Working login screen
    - The admin and professors will need to login in order to access the necessary information
  + The ability to add courses
    - Professors need to add which courses they teach while the Admin needs to add which courses are being evaluated to the system
  + Work on the ability to logout
    - Users need to be able to logout for security reasons and also so others can login
  + Ability to Register
    - New users need to be able to register in order to access the website and its capabilities
  + These are a few of the PBI’s that we will have. We are unable to provide change, defects, and technical improvement PBI’s due to the fact that we have not started building the website yet.
* How will we establish a definition of ready?
  + We will establish a definition of ready using the following checklist
    - The item is small enough to be completed in one sprint
    - All members understand what must be done in the current sprint in order to complete item
    - Item must have a clear value to the website
    - There are no external or internal factors prohibiting from use completing the item
* How will we conduct grooming?
  + All members of our team will have an active hand in grooming the PBI, however the product owner (yet to be determined) will have to final say so. Each member should contribute 10-15% of their time each spring into grooming. We will most likely conduct grooming before we start moving PBI’s into sprints.

**Estimation & Velocity**

* What processes will you adopt for PBI estimation?
  + After a deep analysis of all the different methods of PBI estimation, our team will adopt the Planning Poker Method
* What metric will you use for PBI estimation?
  + We will be using Ideal Days as I feel this is the simpler of the two units to implement
* How will you do task estimation?
  + We will do task estimates as a team. We will do the estimates during sprint planning, and they will be sized in ideal hours
* How will you track velocity and adjust the team's activities back on velocity?
  + We will track velocity by seeing how many PBI’s we can complete each sprint. Which in turn will help us understand when we will be done. We will adjust the velocity by reexamining our product backlog and adjusting how many PBI’s are in each sprint.

**Technical Debt**

Identifying Technical Debt will go hand in hand with our sprints as we work through building the permissions and different pages and forms that will be accessed. We will come across code that needs to be adjusted and “tweaked” for the sake of having our page run smoothly. During that time of the initial build, the code could have looked correct and ran okay, but once we started to progress in the site’s development, that code could become irrelevant and need to be adjusted accordingly. When it comes to breaking down debt in our code, we will not be looking to “pay down the interest”, but we will be looking to pay down the debt and taking care of the issues as they arise.

In an attempt to avoid technical debt, team Gold will be keeping up to deadlines as much as possible. As expressed in the previous section about sprints, after our weekly meetings with Professor Babb, we will be meeting as a group to determine our weekly roles and deadlines for each task to make sure that stress is limited and we have time to present rough drafts to our professor for any potential feedback and critiques that we can use to update our code before the next meeting time.

**Scrum Planning Principles**

For planning and development purposes, these are some of the principles we have already implemented into our process so far:

* Upfront planning should be helpful w/o being excessive
* Favor smaller and more frequent releases
* Correctly manage the planning inventory

So far we have used these principles to plan the progress of our project up to this point. Principles that we are going to be implementing into our process after reviewing this chapter are:

* Keep planning options open until the last responsible moment
* Focus more on adapting and replanning than on conforming to a plan
* Favor smaller and more frequent releases
* Plan to learn fast and pivot when necessary

These principles will be vital to the success of not only our project, but the functionality of our group. Learning to keep planning options open until its due date allows for us to give each idea and thought time to develop, and to be refined as needed. Focusing on adapting as the code grows will allow us to be more open to shifting course, as well as seeing new angles to adjust our code to produce a more full and complete end product. If we focus more on small and frequent wins or releases, then the tasks ahead won't seem as daunting or scary for lack of a better word. And if we plan on learning on the fly and adjusting where needed, then as we progress through the code, not only can we see new life in our product, we will also learn new ways to code and create.