

CS 314 Agile User Stories Pre-Class Assignment

Pre-Class Tasks:

1. Complete pre-class readings. (Estimated time: 20 min)
2. Complete the pre-class preparation. (Estimated time: 10 min)
3. Take the on-line quiz before 11:55pm on Sunday, Feb 7. You may take the quiz multiple times.

Overview

This pre-class work is meant to help you complete the following user stories:

1. As a student in CS 314, I want to understand the Scrum process and how user stories fit into it so that I can practice the process to be a better developer.
2. As a student in CS 314, I want to understand how to accurately size user stories so that my team can commit to a number that can be completed in one 2- week sprint.
3. As a student in CS 314, I want to understand how to decompose a small user story into tasks with accurate time estimates so that I can get all the work done in a 2- week sprint that I committed to.

Pre-Class Readings and Video:

- Agile Scrum and User Story slides: 'CS314SP2016-ScrumUserStories.pdf'. This pdf file contains NOTES pages. The slides have several notes that are important. Please study these.
- Story point estimation article: <https://msdn.microsoft.com/en-us/library/hh273055>. Read up through the definition of "Planning Poker". This article gives a good example of relative estimation. Copyright statements require it be accessed via the URL link.
- Rally.com short video on user stories: "Chalk Talk: Writing Good User Stories"

Pre-Class Preparation:

In Assignment 2, you will be changing the AdventureGame code to enable other types of user interfaces to be added. Dr. Bieman will be providing some skeleton class code that you will have to adapt and integrate into the existing game code. The existing code will have to be restructured somewhat in order to do this. Assignment 2 has two user stories. One of these is the following:

User Story 1: As a game player, I want to use a GUI so that the Adventure game is easier to play.

Acceptance criteria:

- The game uses a GUI for all commands previously available as text commands.

This user story is large, so it needs to be split. We have created two split user stories for it. These are:

Split story A: As a CS 314 student, I want to refactor the Adventure game design so that it is easier to change the user interface.

Acceptance Criteria:

- The main branch game source code structure matches Dr. Bieman's class design.
- Meet 'Definition of Done' (DoD), below.

Example Tasks:

- (1hr) Infrastructure: create git branch, add new code (new game classes, BreezySwing.jar), set up build path in Eclipse, etc. Result is that all new code is merged into master and ready for everyone to work on.
- (1hr) Study class diagram and code and figure out what code needs to be moved to match class diagram. Change code accordingly.

Split story B: As a CS 314 student, I want to implement Dr. Bieman's model-view separation design and get the game working with this GUI so that a player can use the GUI to play the game.

Acceptance Criteria:

- The game uses a BreezySwing GUI for all commands previously available as text commands.
- Meet 'Definition of Done' (DoD), below.

There are 2 parts to the definition of done. One part relates to the “delivered” user story and the other part relates to the development process you need to use to implement the story. These are:

Definition of Done (DoD):

For all user stories:

- GitHub main branch code runs with no warnings and a user can access a GUI to execute all commands that were possible from the command line version
- All code changes are reviewed before being merged onto the main branch (see [A1, tutorial 5](#) for instructions on the GitHub Pull Request Code Review System you must use for reviews)
- All code changes are merged into the main branch
- All code is unit tested and all tests pass

For CS314 development:

- All work is done on individual branches with frequent commits and pushes to GitHub before being merge onto the main branch (do frequent commits and pushes to prevent losing work, and do pushes so others can use your work)
- Commit messages include user story and task
- All user stories and point estimates are in GitHub
- All tasks and time estimates and actual time taken are in GitHub
- The overview.txt file for A2 has all required items in it

One in-class activity you will undertake is to estimate the size of the split user stories using the Poker Planning game. To help you with this task here are some of Dr. Bieman’s notes about Assignment 2:

“The GUI makes use of the BreezySwing package, which simplifies the design of the GUI. To run the GUI version you need to run "AdventureGameView.main". You will discover that the GUI interface is not connected to the game. CaveGameView displays the GUI, and the GUI buttons make the right calls to AdventureGameModelFacade. However, all of the methods in AdventureGameModelFacade that should implement the buttons are empty.

Your job is to connect AdventureGameModelFacade to the rest of the system, finish implementing AdventureGameView (add the GUI code needed to get the "Grab an Item" and "Drop an Item" buttons to work) and get the whole system working.

Surely, there are some other modifications required to implement the design change. For example, you will need to move some of the functionality (and code) in the original class AdventureGame into class AdventureGameModelFacade. Lots of the code in AdventureGame deals with the command line interface interactions with the user. This code is greatly simplified with a GUI.

Key problems you will find are (1) finding all of the output statements hidden in the code, and (2) writing the dialog GUI code. You may need to add an additional TextArea to the GUI to display information other than the room descriptions.

You may use your own creativity to add more interesting textual descriptions of the rooms. The descriptions should be consistent with the layout in the instance diagram. Use good taste in your descriptions --- do not use offensive language. Keep the room numbers in your descriptions to make it easier to test.”

Read and understand the above material, and think about how you will:

1. Estimate the story points of the 2 split user stories (A and B).
2. Decompose the B split user story into tasks and estimate the time it will take to complete them.