Sprint Planning

## Know the end from the beginning

* When the sprint ends
* How many hours you have to work on this project this sprint
* Who is going to be here on what days (vacation / other class priorities)
* Pick a metric to evaluate each other on during the retrospective
* For sprints 2 & 3: List 1 item from your retrospective that you said you were going to do better at that you are going to focus on this sprint.
* Assign a ScrumMaster for this sprint
  + List all team members on this document (indicate who was not here)

## Fill out details for each story

* Make sure all previous stories in the “Done” column are archived in your GitHub project
* Each story that you bring in has a description with:
  + Size estimate by the team
    - Small (< 1 day)
    - Medium (1 day)
    - Large (2 days)
    - Any larger than this should be broken down into smaller tasks
  + Description of what is in scope, what’s out of scope
  + Acceptance criteria in the description stating what it will look like when it’s done
  + Tasks that will be accomplished to complete the story
    - These tasks can be created in the GitHub project as notes (but then convert them to issues so that you can assign a person)
    - Assign a size estimate to each task (S, M, L)
* Stories and tasks should not be assigned to a specific person (unless carrying over from a previous sprint)

## Artifacts

* Stories / tasks are created and on the sprint backlog
* Create a spreadsheet graph for burndown by totaling up the size estimates and setting that as your amount of work left to do. (commit and push to /docs/planning/SprintXBurndown.xlsx)
* Sprint planning document (with the top section information to /docs/planning/SprintX.docx)
  + Attach a screenshot of your Sprint Backlog after planning

Sprint ends: 11/22

Time

8 – 10 hours each

Metric

Point system for each issue completed

What do we want to get done?

Unit tests

CSS

Process the live bid

Return the item winner list

return item history for admin

list all item history

User winning total

admin view of all user winnings and total amount.

What to do better.

unit testing.

communicating.

Scrum Master: Brock Francom