

# **Prioritizing and Estimating Exercise**

# Overview

You now should have a set of user stories, and a story map organizing them into themes.

Our last deliverable is a set of prioritized stories (10) and associated acceptance criteria and story points.

# Prioritizing Stories

We went over MOSCOW in the lecture (see [slides](#)). Use the story map to figure out the Must-Have requirements (likely the ones you have higher up the list of tasks).

Our aim is to build the `walking skeleton' that will cover the MVP for our project.

Spend 10 minutes choosing these. You should anticipate spending most time debating between should-have and must-have requirements.

You probably should separate the must-have stories for Assignment 4 (backend stories) from the Assignment 5 (front end/interface stories). I recommend a GitHub label for this.

# Estimating Story Points

Your next task is to assign the Must-Have stories - the ones you build in the next sprint - a story point value. Story points are a rough list of how hard the task is to finish.

There are several approaches to this. In this lab, we will use a simple version of planning poker. Each person gets 7 numbers to choose from to assign: 0, 1, 2, 3, 5, 8, 13.

# Point values and effort

The numbers reflect "effort" to implement that story. 0 would be a story that is already done or doable in 5 minutes (maybe flicking a feature toggle). 13 is a story that would take the entire sprint (3 weeks) to finish.

You have 3 weeks for half the stories, and I estimate you spend 20 person-hours a week on the project = 60 person-hours. You might look at story points as "one person-hour".

If you have 5 stories at 13 points each, you have not broken your stories down enough.

# Procedure

For each of the Must-Have Stories,

1. pick a narrator to read the story out (sit in a circle). Then
2. make sure people understand what the story is about (simple technical clarifications).  
Finally,
3. in unison, each person should say their chosen number (e.g. Inderjit says "5", Mohamed say "8"). If everyone agrees, move on to the next story.
4. If there is disagreement, discuss to reach consensus. Move on when everyone agrees the value.
5. The scribe should write the choice down (on the GitHub card for that story).

# Things to watch for

- one voice dominating all the discussions and/or
- the one guy who has "done this all before" low-balling all the estimates (it's usually a dude!)
- lack of shared understanding on what the story implementation entails
- imposter syndrome: feeling like you don't have the skill to implement any of the stories
- technical choices (React, Vue.js, ) that strongly influence story effort
  - you may want one story to be "investigate if <technology X> will work"
- stories need to be implemented, but the assignment(s) itself has other deliverables: test plans, BDD docs, CI configuration, etc. These are not part of the 10 stories but **STILL REPRESENT EFFORT!**