# Building Requirements for Semester Project

# Spring 2017 CS4320/7320 Software Engineering

### **Group Assignment: Semester Groups**

## Problem

* Need: A system for easily viewing and parameterizing metrics about Open Source Projects on GitHub.
* Purpose: To create a system which allows potential contributors, project owners, corporations, and other OSS project stakeholders to understand the health of various GitHub Projects. Different projects will have their health indicated in different ways. For example, a long running project with 100’s of contributors will be interested in different health indicators than a new project with less than 10 contributors. Additionally, the indicators should be highly visual.
* Client Base: GitHub users, open source project stakeholders, corporations involved in open source.

## Deliverable

Create a wiki page in the GitHub repository for your group. Create a page called group\_x on the wiki. Place your requirements in a single wiki page that is linked to that page.

Your GitHub repository is a fork of the main repository for the project where your stakeholders live, located here: <https://github.com/OSSHealth/ghdata/tree/dev>

Then complete the steps below, presenting the information in an organized fashion. When you are finished, one of you submit a link to your group wiki page to Canvas. Note the details under Step 4 regarding other document requirements.

Your “language of construction” for all text based work in your repository is “markdown”. You should familiarize yourself with this page so that you can use the markdown syntax easily. <https://daringfireball.net/projects/markdown/syntax> …

The measure of success for “.md” files (which are different than the wiki in this assignment” is, “do they render in a markdown preview editor like “atom” ( <http://atom.io> )? **\*\*It is not “do they render on GitHub”?\*\*** This is because GitHub’s markdown syntax is not exactly the same as most people use, and I would rather require you to learn a method that is not GitHub dependent.

Step 1

* In this exercise, you will create use cases for the semester project, which is based on extending the work in the repository “ghdta” in the OSSHealth organization. Your first step is to find a software program that you can use to draw use case diagrams. You’re welcome to use any program for this assignment. Here are a few suggestions:
  + yEd: [http://www.yworks.com/products/yed (Links to an external site.)](http://www.yworks.com/products/yed" \t "_blank)
  + Microsoft Visio or Powerpoint
  + Sketchbook: https://[sketchbook.com/?locale=en (Links to an external site.)](http://www.sketchbook.com/?locale=en" \t "_blank)
  + Violet UML Editor: [http://alexdp.free.fr/violetumleditor/page.php (Links to an external site.)](http://alexdp.free.fr/violetumleditor/page.php" \t "_blank)
  + Draw IO : [http://www.draw.io (Links to an external site.)](http://www.draw.io/" \t "_blank) (recommended by Jeremy)
  + We will also accept neatly drawn hand diagrams, but make sure you know how to write use case diagrams.

Step 2

* Familiarize yourself as individuals and as a team with the OSSHealth/ghdata project by reviewing the existing wiki, and possibly downloading and attempt run the /dev branch. Seek information about requirements in the slack channel #OSSHealth. This is your chance to get feedback and ideas directly from Stakeholders.

Step 3

* You should prepare a sufficient number of use cases to serve the purpose of scoping your team’s semester project. This may not be the final version of the requirements. We are beginning with use cases because they are high level at the outset, and form a solid foundation for keeping your team centered on how to get the work done. I suggest the following steps:
  + Complete step 2
  + Make a list of actors for the use cases
  + Make a list of tasks/actions that constitute the use cases
  + Create use case diagrams and short descriptions
  + Ask questions of your stakeholders using the slack channel
  + Iterate on the use cases; possibly changing some, dropping some and adding some.
* Your team is responsible for several use cases. ***Each individual should contribute to three different use cases*,** through either independent design, design collaboration, or careful peer review of a finished diagram.
  + Carefully plan responsibilities for design and peer review so each individual works on three different use cases.
  + For 5-person teams, build a minimum of 5 ***Use Cases***, and for 6-person teams, build a minimum of ***6 use cases.***
* Each use case should contain the following elements:
  + **Title** (active verb phrase, states main goal)
  + **Description** (This may be several paragraphs. Context is important. You are describing the use case in some detail, and since many of the use cases will involve users changing parameters on data visualizations, you should be exceedingly clear about this type of thing.)
  + **Triggers** (What prompts the use case to start?)
  + **Actors** (Who is involved?)
  + **Preconditions** (This includes things like “data loaded”. Or, project is flagged as “of interest”; etc.)
  + **Main Success Scenario** (Goals) (What does it look like when the user’s work is successful in the system?)
  + **Alternate Success Scenarios** (For a data analysis and “data playing” focused project like this one, there could be several different success scenarios for each use case. “Sees visualization” is **not** a success scenario. “Compares four different projects on “indicator X” and saves “project trackers” for each one could be a success scenario.)
  + **Failed End Condition** (“crashes” is not a failed end condition. “User is unable to discern the difference between two projects because they are similar on the available indicators” might be).
  + **Extensions**
  + **Steps of Execution** (Requirements)
  + A **use case diagram**, following the UML Standard for expressing use cases.

Step 4

* Review each use case as a team.
  + Provide a full list of use case pages. You may want to simply list the use case wiki pages that exist under your main group page at the top. .
  + Each use case should have a title. Each use case diagram should be captioned. You should succinctly describe the actors and activities in the caption. When you caption something “use case diagram” with no further explanation you are wasting both your own time and the reader’s!
  + Each diagram should indicate who worked on it and in what role.
* Your work will be assessed on the completeness, consistency, clarity and notational correctness of your group use cases.
* Your Group Assignment is due in Canvas at 11:59 pm on Thursday, March 16.

Step 5

* Deploy the dev branch of your fork of the main project. If you have questions or things do not work, put an issue on the GitHub repo for the main project, found here: <https://github.com/OSSHealth/ghdata/tree/dev>

Step 6

* Bonus points for any group that deploys its project fork and opens an issue related to something they do not think is working correctly (to the main project that you forked) . … found here: <https://github.com/OSSHealth/ghdata/tree/dev>

***\*\*Remember to work in the /dev branch. That’s where the latest “stuff” lives\*\****