CSC301H5 - Introduction to Software Engineering

Final Project

University of Toronto, Mississauga Campus
Winter 2023
Due: Feb 10th, 11:59 PM EST

Sprint 1

PREFACE

Great job! You have successfully completed Sprint 0 and you are now on the first stage of developing your new project! We hope you have fun and gain lots of valuable experiences while bringing your project to life.

We're all here to help you achieve both of these goals. In class we talked about how an Agile team works on a software development project: the process, the roles, etc. The project is where you put all this knowledge in practice.

This document outlines how your work will be assessed and what is expected from Sprint 1 of your project.

> **Important: Branches**
> **Do not delete your remote branches** on your repo until sprint marking has concluded, otherwise we will be unable to mark that work and you will receive a 0 for that portion.

NOTABLE INDIVIDUALS FOR YOUR AGILE PROJECT

- **Product Owner**, your product owner will be the course staff.
- **Scrum Demo TA.** At the end of each sprint, you will be demoing your software online, during your tutorial day. The exact timings for these demos will be announced later.

PROJECT PLANNING

Include all aspects of your project plan. Among other artifacts, we will definitely want to see:

- **RPM.md.** The release planning meeting document RPM.md, document the Release Planning Meeting. This document has to indicate clearly the release goal, the scope of the project (at least in terms of epics/key features) and the participants.
- **sprint1.md.** Once you complete the Release Planning Meeting, you should also do the Sprint 1 Planning Meeting documented in sprint1.md. This document must clearly indicate the sprint goal, all stories for this sprint clearly identified, team capacity recorded, participants are recorded, decisions about user stories to be completed this sprint are clear, tasks breakdown is done.

PROJECT TRACKING ON TRELLO

See marking scheme for the full breakdown on what we expect to see for your project tracking on trello

STANDUPS

Standups are a way for you to communicate with your team what you've worked on, what you will be working on and any blockers you encountered.

For your project, standups/team updates are done on the CSC301 Discord server within your team's #standup channel.

Each standup should follow this format:

```
[Standup Date] - Sprint # Standup #
1. What did you work on since the last standup?
2. What do you commit to next?
3. When do you think you'll be done?
4. Do you have any blockers?
```

Each person is required to post a minimum of 3 standups per sprint and **ALL** standup updates must answer the necessary questions and are of good quality.

We encourage you to do more than three to ensure team communication and to resolve blockers. In addition, they should be spaced out.

DOCUMENTATION

Accompanying your code should be documentation. This is crucial in the workplace as it smoothes the onboarding process and provides a handy resource for developers to share their knowledge.

You may choose to have a folder of text files to serve as your documentation, but there are many technologies out there that provide documentation tooling that you may find useful for your project and worth consideration, including but not limited to:

- Github Wiki
- ReadTheDocs
- Storybook
- Bit
- Swagger UI

In addition, you should be updating your README as appropriate based on the changes in your codebase.

SYSTEM DESIGN DOCUMENT

The System Design document must be prepared in a format that we can read (PDF, MS-Word, md, html) and it must be stored in the same folder `doc/sprint1`.

You will likely learn and use a new technology in the course of the project. It is therefore likely that the system design you provide here will undergo major changes as you work on the project. Don't panic! Get together and spend some time brainstorming. You will also receive feedback on this from me and the TAs.

- Include a high-level description of your classes using CRC Cards: what they are, what their responsibilities are, and what is the interaction between them.
- You can use the following template for the CRC Cards:

Class Name:	
Parent Class (if any): *List the parent class if applicable* **Subclasses (if any):** *List all the subclasses separated by a comma*	
Responsibilities: • *List responsibilities*	**Collaborators:** • *List all your collaborators*

- The description of system interaction with the environment should indicate any dependencies or assumptions made about the operating environment of the system. E.g. OS, programming language compilers and virtual machine, DB's, network configuration, etc.
- Describe the architecture of the system, that is the most abstract view of how your system is divided into components and how those components are interconnected. The architecture should be described with a diagram showing components and how they are

related (or equivalent in words). Beware of designs based on a large number of components, they may signal a design that is overly complex.

- The system decomposition should relate the system architecture to the detailed design, to identify the role of each component in the higher-level architectural view. Description of strategy for dealing with errors and exceptional cases (e.g. invalid user input, network or external system failure) that might arise in the use of the software. For anticipated errors and exceptions, a summary of how the software will respond in these situations.

- The following tools may be useful in generating your software architecture diagram:
    - Draw.io/mxGraph - http://www.draw.io
    - LovelyCharts - http://my.lovelycharts.com/
    - Cacoo - http://cacoo.com
    - Draw Anywhere - http://www.drawanywhere.com/
    - Creately - http://creately.com
    - Diagrammr - http://www.diagrammr.com/
    - Grapholite - http://grapholite.com/
    - Gliffy - http://www.gliffy.com
    - LucidChart - http://www.lucidchart.com

DEMO WITH THE TA

Please make sure to stay within the scope identified during the planning meeting. You should be able to demo your software at the end of sprint 1..

During the demo, all team members must be present. The TA will mark the attendance for everyone on the team and ask you to show your working software. The TA should be able to use the software to the extent of the feature(s) you have already installed. You should demo **at least three** features to your TA to receive full marks.

Afterwards, you should conduct your Sprint 1 retrospective meeting. Appoint a notetaker and document your observations about sprint 1 in a document named `SR1.md` in `doc/sprint1`.

You should record:

- The participants in the meeting

- Unfinished tasks and group them into stories; add them to SR1.md in the form of new user stories.
  Update your PB.md and save the updated copy in doc/sprint2 (yes, sprint2 - prepare for the next sprint!)

- What are practices that you should continue during next sprint
- What are some new practices that you might want to use during next sprint
- What are (if any) harmful practices you should stop using during next sprint
- What was your best/worst experience during sprint 1

MARKING EVALUATION

Sprint 1 is worth 18% of the final project mark. Please review the marking scheme for the full mark breakdown and items that are needed for submission.

Additionally, there will be a peer evaluation submission on quercus. The peer evaluation must be submitted before 11:59pm on the last day of your sprint (when the sprint is due).