SC-T-303-HUGB - Software Engineering

Project Overview

Reykjavik University August 21, 2019

1 Introduction

During the Software Engineering course in the Fall 2019 term, you will develop a software system in a group of students. The focus of the project is on using a specified **process** throughout the project duration, and to **use systematic software engineering practices**. Due to the short course duration, the project covers some parts of software engineering more than others. Specifically, we will use an agile process, close to SCRUM, and write substantial tests for our code (mainly on unit level). In addition, the first project part will have a focus on requirements and modelling, while the last project part has elements of software architecture and design.

Organisational details are covered in the project introduction video, available on Echo360.

2 Process

You will follow a process similar to Scrum, dividing the work into five iterations, so-called **Sprints**. Each sprint is 2-weeks long, starting with the first sprint on 22nd August, during the first lab session.

As a starting point, read the Scrum introduction *Scrum in 5 minutes* (pages 4-9 only) under the following URL: `https://issuu.com/softhouse/docs/scrum_5min_eng_131210`. The article describes the regular Scrum process, to which we make a number of adaptations here to fit the scope of the course:

1. The TA is your Product Owner (PO)
2. The Product Backlog is created by the team, not by the PO (but with input from the PO)
3. To describe Product Backlog items, we will use the User Story format (more during Lecture 2, or under `https://www.atlassian.com/agile/project-management/user-stories`). During Sprint planning, user stories should then be broken down into smaller tasks.
4. Daily stand-ups are done during meetings (aim for twice a week), not on a daily basis
5. The first sprint follows a different structure. Here, the aim is to form an understanding of the domain and to populate the product backlog with user stories.
6. The Scrum Master role will be rotated between group members throughout the project duration. The role will be randomly assigned (by the instructor), as there are more group members than sprints.

The weekly Lab/Dæmatímar slots will be used to meet your TA/PO. After each sprint, you will discuss your reflections and plan for the coming sprint with the TA during the Lab session (in a shorter meeting, approx. 20 minutes). Make sure to prepare, so that you cover all relevant information during this short slot.

The scope of the sprints differs in this course. Roughly speaking, the first sprint will be a planning sprint, sprints 2-3 will be used to implement a basic backend, while sprints 4-5 will be used to develop a frontend (and adapt the backend if necessary). Sprint 1 is discussed in Section 5, while instructions for the remaining sprints will be published on Canvas at a later point in time.

3 Project Domain

To allow for comparable complexity, all groups will use the same topic. The aim of this project is to create a software system (at least a prototype) to support **Sharing Economy**. According to Wikipedia, in the sharing economy, "individuals are said to rent or 'share' things like their cars, homes and personal time to other individuals in a peer-to-peer fashion.". The system developed by all groups is intended to support this way of sharing.

There is a large space of features and solutions that can be devised to address this goal and anything surrounding it. Groups will, supported by the TA, explore this solution space and come up with their own focus. Therefore, each group's solution will differ.

The project is intended as a **bespoke** project with the TA serving as a customer representative. This means that the TA has the ultimate say in what features should be prioritised, which features are desirable, and which ones are not. The groups are, however, encouraged to discuss with the TA - as in a real project, the customer might not see all the opportunities and could therefore potentially be convinced by argumentation!

4 Solution Domain

Each group will implement the system in **Python**. That is, the system will be intended to be used on a desktop PC/Laptop. Potential web or mobile applications are out of scope (based on the course prerequisites).

To allow the system to be structured into sub-systems, we will be using **WebSockets** to communicate between sub-systems. WebSockets allow message passing between different systems, and therefore enable numerous design and deployment alternatives for the final system. For instance, the system could be structured as a basic client-server architecture, with the server containing all the necessary functionality to enable sharing of goods between multiple clients. The client part would then contain the frontend, and would interact with the server by sending messages over WebSockets. Deployment will not be part of the course, so it is enough if your system runs on a local computer.

From Sprint 2 on, the **unittest** framework will be used to implement functional tests.

From Sprint 4 on, a rudimentary frontend will be implemented using **PySimpleGUI**, while the existing functionality will be evolved at the same time (e.g., refactored to support the frontend).

Groups are free to use any additional libraries/frameworks. However, this has to be a consensus-based decision, where all group members agree and are able to use the technology. Also, interfaces between sub-systems have to use WebSockets only. WebSocket communication between the frontend and backend cannot be replaced by another technology. To be sure, it is recommended to discuss choice of additional technology with the TA.

For simplicity, we exclude authentication and authorisation. While the system shall support multiple users, you do not need to consider how to ensure that users are indeed logged in, and that the system has a reasonable level of security.

Similarly, we exclude persistence. It is accepted to keep all data in memory, and to start up the system with sample data (e.g., a number of standard users, entries, etc.). This simplification is again related to the fact that the course on databases is not required to attend this course.

5 Sprint 1

The first sprint takes the role of a planning sprint, intended to give the group an overview of the project and some time to start up the work. No implementation will take place during the first sprint, unless individual group members want to experiment with the technology (e.g., try out WebSockets).

The first sprint has a number of deliverables:

1. Set up GitLab[1] repository
2. Project overview report
3. Product Backlog on GitLab
4. Decision Protocol

For deliverable 1, you are supposed to set up a **repository** on GitLab. The repository has to be named **T-303-HUGB 2019 Group [groupNumber]**, where groupNumber is the number assigned to your group. The repository has to be set to private, include all your group members, and **all** the course TAs and the instructor[2] (giving TAs/instructor the Reporter role is sufficient - we don't need write access). Each repository has a **src** folder (where all the source code will be placed), and a **docs** folder (where notes and protocols will be placed, sorted by sprint). For Product and Sprint backlog, we will be using GitLab's inbuilt issue tracker and boards feature. You can fork the sample project[3] that already contains the relevant folders, as well as a fitting board with one sample user story.

The **project overview** document will be a textual document (with pictures/diagrams as needed). In this overview document, you should sketch the system in its context. Try to answer the following questions:

- What is part of the system? What is outside?
- What relevant stakeholders/personas exist?
- What features do we envision? (Note: This is not a promise that you will implement all of these features, just a brainstormed list of potential features).

The project overview shall contain a **domain model** (see lecture 3), and a number of scenarios (see lecture 2) that illustrate the use of the system. To arrive at feasible scenarios, you can use regular brainstorming, other elicitation techniques covered in lecture 2, and anything you have learned in T-216-GHOH. Note that most detailed elicitation techniques (e.g., interviews or observations) will take too much time and resources to conduct them in the scope of a single sprint.

The **product backlog** in your repository shall be filled reflecting the scenarios in your project overview. Create issues on GitLab that follow the User Story format (see lecture 2) and assign them to the **Open** list

[1] `http://gitlab.com`

[2] GitLab user names: grischal, olgakristjans, ballioli, gabriels17, bjornbrynjar

[3] `https://gitlab.com/grischal/hugb2019template`

on your repository GitLab. The user stories shall follow the format **As a [persona], I want to [feature], so that [benefit]**, and should relate to your scenarios (that is, all user stories together should support the scenarios in your project overview). Additional explanations or rationales to a user story can be added as comments to the issue that represents the user story.

Finally, you should for this sprint (and for all following sprints) keep a documentation of your meetings. For sprint 1, this means documenting all important decisions that you made as a group. This does not need to be excessive - just document important things that help us understanding where your group is headed.

You do not need to consider economics in your project. Assume that the customer pays for the project and does not expect a concept of how revenue is generated from the system. Neither do you have to take into account how design decisions would affect operating costs.

Submission and Assessment

Sprint 1 deadline: Wednesday, 4th September, 23:59
For each sprint, there is a group and an individual part. For the group work, we look at your GitLab repository only, considering the latest state before the deadline. It is your responsibility as a group to make sure all relevant files are committed on time. Any late commits will be ignored.

For Sprint 1, make sure the following parts exist:

1. A GitLab repository, with correct name, that is both private and shared with all TAs and group members

2. Project overview in the docs/sprint1 subfolder

3. Decision protocol in the docs/sprint1 subfolder

4. Product backlog as a number of issues assigned to the Open label in GitLab

The individual part is submitted through Canvas (assignments are named by Sprint). It consists of a peer assessment of all group members (including yourself). If a group member forgets submitting this part, his/her view is not taken into account - he/she effectively loses the chance to raise concerns regarding team work.