Issue Tracker Express

Sprint Retrospective 1

CPSC 3720

University of Lethbridge

Team Aegir

Lorenzo Conrad
Mathew Richards
Steven Deutekom

October 28, 2019

1 introduction

We have completed our first sprint for Issue Tracker Express. Below we reflect on the experience in order to improve our process for the next sprint. We will give some details of our initial plan for the first sprint. Then we will show what we actually got done. This will be followed with some description of a few of the major problems we encountered during the sprint. Then we discuss a plan for the second sprint and describe what we will try to do differently this time around.

2 Sprint Plan

The intention for our first sprint was to get a very minimal working product[1]. We wanted to be able to add an Issue and to see an overview of all the issues that have been created while the server is running. We also hoped to have the ability to add and list users.

In order to view these things we planned to get a simple website working to allow a user to create both issues and users. The website was supposed to have a home page that would for the moment give some information on the system. We would have a few links to allow navigation to the Issue and User pages. On these pages we wanted to have a list of essential information for the Issues or Users. There would also be a link on each page to allow addition of new objects.

[1] See our proposal for a more information on what features we intended to complete in the first sprint

The code for our system we wanted to have the basic classes for Issues, Comments, and Users to be setup, documented, and fully tested. We also wanted to find a good way to organize classes for a system that would hold the objects for these classes while the server is running. We wanted to create a server class that would hold the system and to handle requests.

We planned to have weekly meetings to discuss our progress. At these meetings we could each explain how we were doing with our individual tasks. We also talked about the next steps we could take and what our schedules looked like. At these meetings we could take notes on our potential designs and keep ourselves on track. We also gave ourselves the opportunity to help each other solve any problems that may have arisen.

3 Sprint Results

At the end of our first sprint we did not complete all of our goals. We were able to get a simple website working. It has a home page that describes the project a little bit. We were also able to add an issue page that shows all the issues currently in the system. There is a button to add an issue. When it is clicked a new small window opens with a form for submitting an issue. For now all this does is submit a test issue to the server when the submit button is clicked. Currently, the contents of the form do not change the object submitted.

Our system does include classes that we expected to get implemented for this phase. Each class is mostly documented. We were able to test the Issue, Comment, User, and IssueSystem classes for our project. They are currently mostly tested. We were not able to properly unit test the Server and Handler classes for reasons explained in the next section.

We did manage to meet frequently, at least once a week. We discussed how we wanted to design the system. We took some notes and posted them to slack so that we would be able to refer back to them later. It was helpful to get together and discuss the project, but we did not manage to meet in the lab to work on code together very often.

4 Problems Encountered

One of our main issues during this first sprint was being busy with other assignments and midterms. We all had a couple midterms during this sprint and several assignments. This made it hard to get together as often and put in very much work on the project.

We also ran into some problems with the way the Linux environment worked with our code. We were

unable to run our tests with the g++9.1 compiler that is required to work with restbed. This meant any class that referenced restbed could not be unit tested and therefore could not be built with TDD. We communicated this to the right people, but it may take some time to resolve.

Learning how to get JavaScript to send requests to our server was also a little bit troublesome. We ran into an issue specifically with post requests where we were able to send the request ok, but could not read the response. This was never solved. Because the server was at least receiving the requests this was not the end of the world yet, but it will need to be fixed during one of the next sprints. Because we don't know JavaScript very well yet, it was hard to know if these problems were caused by our client or the server. And without proper unit tests for the server we could not confirm that the server was in fact working as expected in every way.

5 Plan For Next Sprint

During our next sprint we are going to try to finish the endpoints that allow us to work with issues and users. We hope to be able to add, get, update, and remove Issues. We also hope to get the same functionality done for Users. Since the server and system are mostly built we do not expect that this will be too hard. Most of the work will come from creating handlers on the server and the client.

We want our client to properly update custom Issues and Users in by the end of the next sprint. This will probably mean that for now we do not worry too much about the look of our web page. We will need to focus on the functionality and learning how to make things work in the best way with our HTML and JavaScript.

It will also be desirable for us to finish any missing documentation. This will include documenting any new classes or methods that we implement. It would also be really helpful to be able to figure out why we cannot compile our tests with g++9.1 and find out if there is a good fix.

6 What To Do Differently

We will try to meet a little more often in the lab to actually work through things together. This will help us when we run into problems with the system. We can collaborate and solve these problems. We also have a tendency to spend too much time chatting about things that are not relevant to the project. This is good for team moral, but it can be an obstacle to productivity. We will try to keep focused during meetings, and when we get off track point it out and re-focus.

We will try to be better at communicating our technical issues to the people that can help us. This may mean asking questions on slack or talking to Dr. Anvik. It is good to try to solve our own problems, but sometimes the issue is not something we can fix. In these moments it is better to find help so that the project can continue. And so that others are not inconvenienced by the issue. In any company we may work for finding issues should be shared with others so that the issues are resolved before they affect too many people.

If we cannot get the testing issues resolved easily it may be possible to write an adapter class for restbed. This way our server and handler classes can use the adapter instead. Then we could unit test these classes without needing g++9.1. Any where we need to test the adapter we can just mock it out.

Most importantly we need to do a better job of budgeting our time. We may have chosen too much for our first sprint as we did not anticipate so many issues getting the server set up with a simple web page. However, we also did not plan our time out really well around other assignments. Even how we plan time for our other class assignments makes a big difference in when we can meet, and how much we can get done on this project.

7 Conclusions

We are relatively pleased with our progress with this sprint. We did not get as much done as some of the other teams, but we tried to organize our project in a good way and that caused us some problems, specifically with POST requests. Some issues with the system also slowed us down. The positive was that we were able to recognize that we would not finish all our planned features in time. We then scaled back out plan so that we got only the basis for each part of our system completed. While it was not perfect to start the lessons learned will help us do better in the future. Also, hopefully some extra thought for our design will mean that we can add new features quicker and easier and already know what problems we might face and potential solutions. We feel that we learned many things about working as a team and attempting Agile practices. This learning will help us in the future with our project, and when we eventually get jobs at software companies.