

Software Engineering TDT4140

Group 26: Project Plan

Sindre Hansen Vegard Helgesen Hesselberg
Eirik Rismyhr Stian Sørli

Spring 2017

Contents

1 Problem	2
1.1 Opportunity	2
1.2 Stakeholder	2
1.2.1 Persona	2
1.3 Requirement	2
2 Solution	3
2.1 Deliverables	3
2.1.1 Software	3
2.1.2 Infrastructure	3
2.1.3 Data	3
2.2 Work	4
2.3 Team	4
2.4 Way of working	4
A Product backlog	5
B Activity plan	6
C Risk assessment plan	8

1 Problem

1.1 Opportunity

For many years, university education has used the same format with frontal lectures, assignments and written exams. This way of teaching is outdated and poses several problems.

We want to change this using the possibilities that computers and information technology provide. By doing this, we believe that we can facilitate discussions and knowledge sharing between the students, and make the courses more interactive. This could make understanding the curriculum easier for the students.

1.2 Stakeholder

- A lecturer at a university or a college.
- A student participating in a class or a lecture at a university or a college.

1.2.1 Persona

A lecturer for a large group of student who wants to engage the students in discussions. He wants to be able to automatically divide students in groups where they will discuss predefined questions. To see what the students answered he wants to collect the answers from the groups.

1.3 Requirement

- The program we are to create has to be able to run as a standalone program or in tandem with other digital platforms.
- The solution has to manage online sources of input data.
- In order to be a functional bot, it must be able to process the online input data and generate output based on the input.
- For the ease of end users, the program must have a functional graphical user interface.

- When the program is finished, the bot will be distributed as open source software.
- While not a requirement, it is encouraged to use other known APIs and make use of open source software.

See appendix [A](#) for the initial product backlog.

2 Solution

2.1 Deliverables

2.1.1 Software

A functional roBOT, able to connect end users together in a chat room and interact with them based on the users' input. The roBOT will also gather all of the users' input as a chat log, which will be automatically sent to the lecturer and/or saved by the students.

2.1.2 Infrastructure

We are aiming for a web-based solution that will be able to run in both browser and on smartphones, although the smartphone solution is not highly prioritized.

At this stage we aim to implement the front-end web application in Django and use Tornado as our web server. We will also need a database for storing messages etc., most likely a MySQL database.

2.1.3 Data

Throughout the development of the software, we will test the application among the userbase to get feedback on its functionality and use. This will be documented so we can see how the software and our vision of the software changes over time, as well as giving us insight in how the userbase responds to our product. There will be delivered documentation on the entire software engineering process, both as a requirement for the TDT4140 course, but also for our stakeholders' sake. There will be a user guide, including how to set up all necessary frameworks and how to properly use the roBOT, so that when

this software is distributed, it'll be possible for potential users to actually utilize it.

2.2 Work

See appendix B for an activity plan with estimated working hours attached to each task.

2.3 Team

Member	Role	Responsibility
Eirik Rismyhr	Developer	Back-end
Sindre Hansen	Developer	Front-end
Stian Sørli	Team leader	
Vegard Helgesen Hesselberg	Developer	Tests and testing

2.4 Way of working

We will use git for version control of active development. For project related communication we will be using gitter, which is directly connected to github. Our git repository can be found on github [here](https://github.com/sindrehan/TDT4140-Project)¹. We plan on meeting every tuesday from 10-14 as well as wednesday from 10-14.

Quality assurance will be done as we go, trying to work in pairs as much as possible. This way everyone will get a wider knowledge of the project as a whole instead of it being focused on one person. As the project goes on we will write tests as we discover what needs to be tested.

¹<https://github.com/sindrehan/TDT4140-Project>

A Product backlog

ID	Story	Estimate	Priority
T1	As a lecturer I want to be able to create a "Super-room" where I can see all ongoing conversations connected to the current lecture.	8	1
T2	As a student I want to receive a code from the lecturer that I can enter into a website and be placed in a chatroom with other students.	10	2
T3	As a lecturer I want to be able to read the chat logs so that I can monitor the discussions.	3	3
T4	As a student I want to be able to save the chat log to my personal device.	3	4
T5	As a lecturer I want to be able to create topics beforehand so that the groups have something to discuss.	4	5
T6	As a student I want to be able to request a new topic if the first one is completed or too difficult.	5	6
T7	As a student I want to be able to send a message directly to the lecturer.	7	7
T8	As a lecturer/TA I want to be able to supervise chat rooms.	8	8
T9	As a student I want to be able to use a laptop <i>or</i> a smartphone for chatting.	4	9

B Activity plan

Release due date	Tasks	Description	Est. resource (h)	Actual resource (h)
23.01.17	Ideation report	Understanding the purpose and goal of the project	-	32
03.02.17	Project Plan, Project Backlog	Pre-project documentation, gain an overview of the project	-	46
10.02.17	Requirement and Architecture		32	-
24.02.17	Sprint 1	Get back-end up and running	64	-
10.03.17	Sprint 2	Must have the ability to create and join chatrooms by now, basic application functionality in place	64	-
24.03.17	Sprint 3	Expanding functionality of the roBOT, adding required features, phone functionality	64	-
07.04.17	Sprint 4	Polish, bug removal and front-end fixing. Testing	64	-
27.04.17	Final report	Documentation and preparation for the final presentation	64	-
	Total		334	-

C Risk assessment plan

Risk	Means to prevent	Action and responsible
Schedule slips and project cancellation	Daily meetings to check progress and short release cycles.	The programmers are responsible for dropping user stories if the schedule slips. The group leader leads the meetings and makes sure the group delivers the product on time.
Business change, cost of changes	Take in new user stories in the planning meeting before each sprint. Automatic unit tests, no errors in final product.	The customer is responsible of informing the project manager when the requirements change. The project manager has to decide which stories to include in the sprint.
Defect rates	Automated unit tests, customer tests.	The programmers are responsible of creating unit tests which check that the code does what it is supposed to. The customer is responsible for testing the product during development and providing feedback to the developers.
Business misunderstanding	Work closely with the customer.	The developers have to understand and analyze the needs of the customer, and divide these into user stories. The customer is required to give feedback to assure that the project is moving in the right direction.
Technology	Sharing of knowledge and pair programming.	The programmers work in pairs to share knowledge and quality check the code. They inform the project manager if any problems occur, and he sets up code reviews and training sessions.
Staff schedules	Weekly schedule planned in the start of the project	It is the project managers' responsibility to create a schedule that works for everyone and make sure that everyone shows up on time. If any problems occur he should inform the the customer.

