# Annex 1: Software Project Plan

In following annex, the organizational aspects of the chatbot’s development will be examined. More precisely, the software development process is described as well as the tools that were used to manage the process, followed by a detailed examination of the course of the project.

The second part of the annex examines the project’s viability, including the calculation of involved costs and possibilities for making profit.

# 2. Project Management

## 2.1 Scrum

The project’s management is inspired by *Scrum1*, an agile software development framework which facilitates managing tasks in teams. The Scrum framework is based on the experience that most projects are too complex to be planned completely in the early stages and therefore provides an agile and iterative alternative by structuring the project in smaller iterations.

However, the applied process during this project is only loosely based on the Scrum framework since there are several concepts of the original framework that were not applied. Scrum defines roles which describes the different members of a group (*Scrum Master*, *Project Owner* and *Development Team*) as well as several artefacts organizing the team’s interaction. Due to the nature of the project not having a development team in the proper meaning of the word, but only a single person realizing the bachelor thesis, only some artefacts were applied. The most important applied concept is the *Sprint*, a two-week time slot which is used to structure the project. At the beginning of each sprint, a *Sprint Planning* is realized between the author and the coordinator of project. The project’s coordinator can be seen as a *Product Owner* in the Scrum framework, prioritizing tasks and guiding the project’s direction.In the *Sprint Planning*, it is decided which tasks are going to be worked at during the two-week sprint. At the end of each sprint, a *Sprint Review* takes place where the development team, meaning the student, presents its results to the *Product Owner*.

## 2.2 Managing the project in GitHub

In order to facilitate the project’s management, the online project hosting tool *GitHub* with *ZenHub* as an extension was used. *ZenHub* provides several useful collaboration features such as a board to visualize tasks as well as an overview of the remaining workload and velocity. The tasks are defined in GitHub’s *Issues* where tasks can be named, described and estimated. Story Points are used to estimate the workload of each issue. In this project, one story point is seen as the equivalent of 2 – 3 hours of work. The iterations are planned and defined using *Milestones* which the different issues are assigned to.

## 2.3 Time Planning

The Kick-Off Meeting took place at 5th December 2016, where the elemental concepts of the project were discussed. Due to the fact that this project was held in the course of the semester parallel to usual classes and exam periods, there were some time breaks between the two week sprints in which the development on the project paused.The project consists of 11 iterations, the last one ending a day before the submission date of the thesis, the 2th July, 2017.

The following section gives an overview of the iterations during this project, finally using Burn-Down-Charts and velocity tracking to visualize the project’s progress.

### 2.3.1 Iteration #1 (30/12/2016 – 17/01/2017): Planning and Research

The first iteration mainly centered on setting up the project's infrastructure, including the establishment of the repository structure and the virtual machine set up. First research steps were taken by getting familiar with the used geo-information system.

### 2.3.2 Iteration #2 (17/01/2017 – 07/02/2017): Further Research

In this iteration, the knowledge of the used geo-information system was deepened by investigating how to extract essential tourist information from the database. On the other hand, a chatbot library was chosen after comparing different possible candidates. Besides, the project's documentation will be extended by describing theoretical concepts, used tools and the project's objectives.

### 2.3.3 Iteration #3 (07/02/2017 – 21/02/2017): Recommender System and Design

In this iteration, recommender system libraries were examined to find out which one was the most suitable for this project. Afterwards, the system's basic architecture was designed. On top of that, the project's documentation was extended.

### 2.3.4 Iteration #4 (21/02/2017 – 07/03/2017): Mockups

First mockups were implemented in which the design of certain components of the application were tested. At first, the interaction of the application's chatbot layer with its environment was set up, using the Telegram Bot API and the API.AI HTTP API.

The recommender component was examined in more depth, meaning that a recommender library was finally chosen and then used to mock up a POI recommender mechanism.

### 2.3.5 Iteration #5 (10/03/2017 – 24/03/2017): Recommender System Mockup

This iteration centered on the design and first implementation of the application's recommender system. The library Apache *Mahout* is used to implement a content-based filtering recommendation mechanism. The iteration includes the design of user and item profiles as well as the investigation of how to retrieve the data from *OpenStreetMap* into the recommender system.

### 2.3.6 Iteration #6 (03/04/2017 – 17/04/2017): Collaborative Filtering & Requirements

In this iteration, the collaborative filtering recommender was designed. Additionally, the project's requirements were examined as well as user-chatbot interaction.

### 2.3.7 Iteration #7(18/04/2017 – 02/05/2017) Conversational Interface Design

In this iteration, the chatbot's conversational interface was designed by modeling a conversation flow graph. The resulting knowledge was explained in the Theoretical Concepts chapter of the documentation. On the other hand, user ratings were generated for the collaborative filtering algorithm.

### 2.3.8 Iteration #8 (04/05/2017 -18/05/2017) Conversation Flow Implementation

The previously specified conversation flow was modeled in API.AI and the application's chatbot layer was implemented to handle the conversation. On the other hand, the recommender component was finished and documented.

### 2.3.9 Iteration #9 (18/05/2017 – 01/06/2017) Conversation Flow Refinement and Testing

The conversation flow was refined to improve the conversation between user and chatbot. The rating mechanism was implemented. Some performance tests for the recommendation mechanism were implemented and the software was evaluated using metrical code analysis tools.

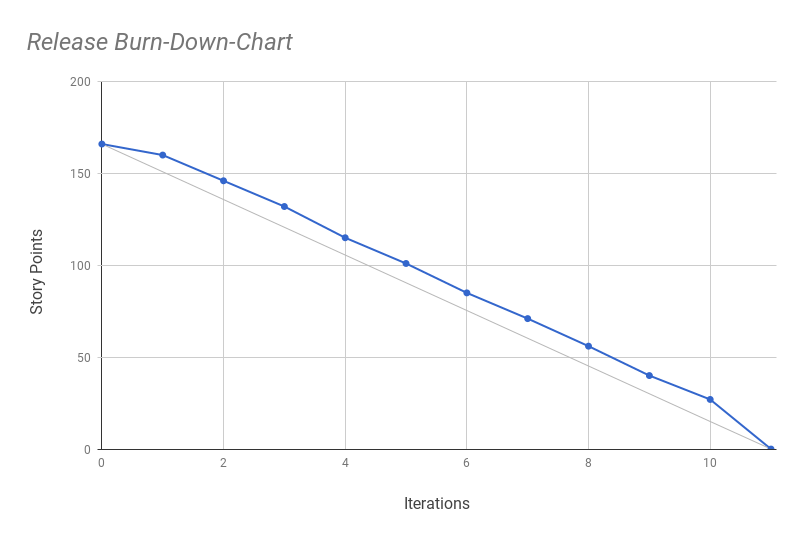
### 2.3.10 Iteration #10 (03/06/2017 – 17/06/2017) Latex Setup

This iteration dealt with setting up the latex file for the project’s documentation. Previously prepared drafts were modified and completed. Several chapters and parts of the annex were introduced into the latex document and the bibliography was formalized.

### 2.3.11 Iteration #11 (18/06/2017 – 02/07/2017) Documentation and Final Conversation Flow Adjustments

This iteration concentrated on completing the project's documentation, especially outlining the project's relevant aspects and come to final conclusions. The application's implementation was only adjusted in refining the conversation flow and small refactorings in order to improve the code quality.

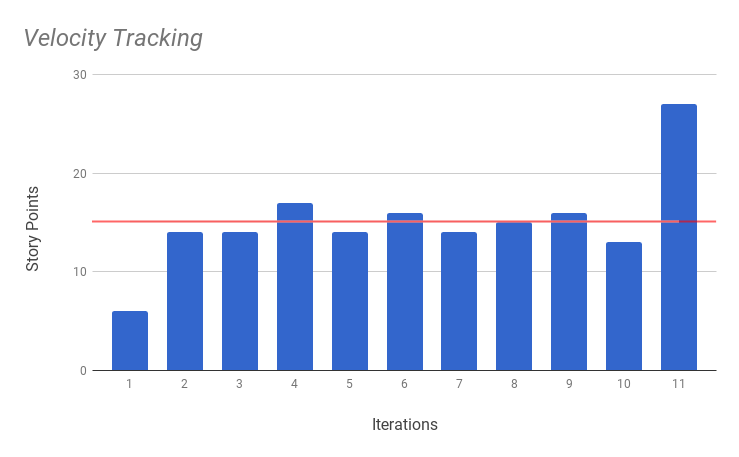
### 2.3.12 Burn-Down Chart and Velocity Tracking



The Burn-Down-Chart shows the progression of story points in the development process. The ideal progression is represented in the grey straight line. As we can see, the actual story point progression does not differ largely from the shown ideal line, as the work load was divided equally to the different iterations.

The workload divided to the different iterations is shown in detail in the velocity tracking diagram. The average work load per iteration is measured with 15 story points, which can be seen in the red trend line. Most iterations do not differ largely from this average measure, except for the first and the last iteration. This observation can be explained as in the first iteration, research was done and a first overview of all the different aspects of the project was made. After this iteration, the necessary tasks to be done became clearer and a first task overview was created.

In contrast, the last iteration is outstandingly bigger than the previous ones. This can be explained by the fact that the student had terminated all of the exams and semester’s course work at that time and could completely focus on the preparation of the project.



# Bibliography

1: The Scrum GuideTM­, Ken Schwaber and Jeff Sutherland