### User requirements, programming languages and project method proposal for Project56

## Group 1: Cooperatio

Table of Contents

[Part 1 – User Requirements 3](#_Toc430787905)

[Part 2 – Programming Languages 7](#_Toc430787906)

[Part 3 - Project methodology 8](#_Toc430787909)

## Part 1 – User Requirements

Our user requirements have been made in the form of a scrum-style product backlog (You will be able to find more information about our scrum-style project method in Part 3 – Project Method) with user-stories for each business requirement, prioritized using the well-known MoSCoW-system.

Our original intention was to retrieve the user requirements directly from CityGis itself. During the second week of the project we tried to contact CityGis by phone and by email with this request but at the moment of writing (23-9-15) we still have had no answer.

Therefore, we took to the advice of our Tutor ms. Van Ijperen and created our own User requirements, according to our interpretation of the CityGis business requirements.

**B1: The system will be able to read and save CityGis-data at high speeds.**

1 As a User, I want to be able to import CityGis-data in CSV format into the system, so that data can be analyzed. (**M**)

2 As a User, I want the system to be as fast as possible, so I don’t have to wait too long. **(M)**

3 As a User, I want to have a simple graphical interface for importing Data, so I can easily store my data. (**M)**

4 As a user, I want to be able to see a progress bar, so I can see how the importing process is coming along. **(S)**

5 As a User, I want the system to inform me of any possible errors in the importing process, so I know what went wrong. (**S**)

6 As an Administrator, I want the system to contain methods for data-importation that aren’t manual, so storing data doesn’t need an employee physically importing the data. (**S**)

**B2: The systems’ architecture will enable the creation and the saving of parallely aggregated reports**

1 As a User, I want to be able to download reports about the collected data and about more specific parts of that data, so I don’t have to go through all the data manually. (**M**)

2 As a User, I want the system to automatically aggregate useful information about the collected data, so I don’t have to select what I want in the report by hand. (**M**)

4 As a User, I want to be able to download these generated reports in pdf, so I can easily distribute these reports through to my colleagues. (**S**)

5 As a User, I want to be able to generate and download multiple reports at the same time, so I do not have to wait for each one of them to finish before the other. (**M**)

6 As a User, I want to be able to see progress bar of the generating/downloading process, so I know when I can safely quit the program. (**S**)

**B3: The system will provide a web –interface that will display CityGis software- and hardware performance statistics, for use in product development.**

1 As a User, I want the web-interface to be easily understood and navigated, so I do not need to be an expert-data analyst in order to use it. (**M**)

2 As a User, I want the web interface to display simple measurements and averages, so I don’t have to make these calculations myself. (**M**)

3 As a User, I want the web interface to display the statistics in a visual way, so I can easily see which components require further attention. (**S**)

4 As a User, I want to be able to generate and download reports containing the performance statistics, so I don’t have to go through all the data manually. (**S***) (See also Business requirement B2)*

**B4: The system will provide a web-interface in which vehicle maintenance-data can be made into reports and downloaded.**

1 As a User, I want the web-interface to be easily understood and navigated, so I do not need to be an expert-data analyst in order to use it. (**M**) *(See also Business Requirement B3)*

2 As a User, I want the system to automatically aggregate useful information about the collected data, so I don’t have to select what I want in the report by hand. (**M**) *(See also Business Requirement B2)*

3 As a User, I want to be able to download reports about the collected data, so I don’t have to go through all the data manually. (**M**) *(See also Business Requirement B2)*

4 As a User, I want the system to also represent the vehicle-maintenance data in an easy-to-understand visual way, so I can easily see the trend of the data. (**S**)

**B5: The system will provide a web-dashboard on which some important vehicle-maintenance data points will be shown to the vehicle-administrator.**

1 As a User, I want any alerts coming from the web-dashboard to be easily visible (from across all the web-interfaces), so I can address these issues quickly. (**M**)

2 As a User, I want alerts coming in the web-dashboard to contain basic information about the problem at hand, so I can quickly see what the issue is. (**M**)

4 As a User, I want the system to allow me to quickly share the information in the alerts, so I can alert my superiors. (**S**)

**B6: The system will provide a web-portal on which CityGis-meta data will be up for sale to 3rd parties.**

1 As a User, I want to buy data from CityGis, so I can use it to my personal purposes. **(M)**

2 As a User, I want to be able to pay via PayPal, so I can easily transfer the money. **(M)**

3 As a User, I want to be able to select which data I want to buy, so I only have to buy the data that I need. **(S)**

## Part 2 – Programming Languages

## Input & Processing: C++

For the input-part of our system, we will use the C++ programming language. The C++ language is relatively low level compared to languages like Java and C#, allowing us to exert greater control over the systems’ memory consumption. This greater control will lead to an increase in the systems’ speed, compared to higher-level languages. C++ also bears some (if not complete) resemblance to Java and C#, allowing us to get started quickly rather than spending hours and hours watching tutorials.

## Output and visualization: Laravel or .NET

For the output-part and visualization of our system, we will use a Model-View-Controller web-framework. Currently we are still considering whether to use the PHP-framework Laravel or Microsofts’ .NET package, as there is still some internal debate over which platform would suit our needs best. We have decided to give each framework a 2 week “probation period” in which we will experiment with it and look at how much it suits our needs.

# Part 3 - Project methodology

Since the beginning of year 1, we used Scrum as our software development methodology. Because Scrum worked nice for us, we want to use it again. But we decided to change the rules of Scrum a bit, because we don’t agree with all of them. So we removed some rules and edited some others. Technically, our methodology is not Scrum: It is Scrum based. Below we describe the differences of our methodology with Scrum.

**Removals:**

* Planning poker (Because none of us keep up the total time he spent on a work item, it isn’t very useful to do so.)
* Retrospective (Because we talk with each other every day and provide each other with necessary feedback we don’t think it is necessary to do a time-consuming retrospective every 2 weeks.)
* Burndown-chart (Trello automatically generates a burndown-chart; but since we all are able to understand when/if a sprint is not going according to plan during the sprint itself, we do not look at the Burndown-chart very often.)
* Scrummaster (If there are any issues between the team and the product owner these are usually resolved by the entire team already, therefore we consider it a waste of time to appoint someone special to be a scrummaster and represent the team.)
* Shippable product. (Sometimes you spent a lot of time on back-end work that is of great importance, but because you spent less time on the front-end, it will look like little work has been done; Even though everyone has spent a lot of time on it. Therefore we would rather present our actual code/progress during Sprint Reviews and look at that with the PO, instead of constantly focusing on creating a shippable product.)

**Changes:**

* The development team is allowed to modify the **sprint backlog** during the iteration without permission from the product owner. (Because sometimes as a team you need to spent more time on a specific work item than originally planned, so you don’t have time for other work items. You don’t want to be forced to rush a work item so you can finish all the other work items as you promised and end up getting a good grade, but delivering a bad product.)

Our iterations will last 2 weeks. On Thursday during the project lesson we will have the sprint review. Here we will ask for some feedback and show our product/code.