

Product Planning

Group Health informatics-2

Rick Proost – 4173619
Pascal Remeijssen – 4286243
Wim Spaargaren – 4178068
Arnout vd Knaap – 4223969
Jelmer de Boer – 4223152

Contents

Introduction.....	2
Product.....	2
1. High-Level Product Backlog	2
2. MoSCoW	2
2.1 Must have	2
2.2 Should have	3
2.3 Could have	3
2.4 Won't have.....	3
3. Product backlog (first version with estimates and prioritized user stories)	3
3.1. User stories of features	3
4. Roadmap	4
5. Definition of Done (backlog items, sprints, releases).....	5
6. Glossary	5

Introduction

Researchers want to analyze raw data, but this is not always structured in a way it can easily be put in Static Analysis tools. You get a diversity of data types like text and excel files from real world applications that needs to be formatted.

This product will format and do quick analytics in an intuitive way so researchers will be able to do their research in an easy manner.

Product

1. High-Level Product Backlog

1. User can convert text data to valid input for SPSS tools.
2. User can convert excel data to valid input for SPSS tools.
3. User can filter their data before converting to valid input for SPSS tools.
4. User can process the 8 C's on their data.
5. User can visualize basic analysis on their input data.
6. User can export visualizations of their analysis.

2. MoSCoW

In this section we display the features of the program, described according to the MoSCoW method. This means that every feature/aspect of the program are divided into four groups:

Must have: Features whose functioning is required to get the program working. These features have a high priority.

Should have: Features that are wished for, but in absence they do not harm the program. They have a medium-high priority.

Could have: Features that are only implemented if there is time, but otherwise are planned for a follow-up project. They have a semi-low priority.

Won't have: Features that are probably for a follow-up project, they have low priority, but could improve the program

2.1 Must have

- User can format .txt file output for analysis tool SPSS (sprint 3)
- User wants to know the difference between what the user enters and what they actually measure (sprint 4)
- User wants to know in what conditions patients override their initial data(green orange red) (sprint 5)
- User want to be able to input data which could be compared to the data of the ADMIRE project (sprint 6)

2.2 Should have

- User can determine output data set (sprint 6)
- User can filter on certain data (sprint 5)
- User is able to use all different Sequential data analysis methods (sprint 7)
- User wants to know if patients follow the websites advice according to data of stat sensors (sprint 7)
- User wants to know the time difference between measuring and entering creatinine values (sprint 6)
- User wants to know if patients measure more if other factors like blood pressure increase (sprint 7)
- User wants to be able to find cases where the website advised to contact the hospital (sprint 5)
- User wants to know how many times patients measure before they enter (sprint 7)

2.3 Could have

- User is able to export visualizations of the imported data (sprint 7)
- User want to know if external factors (holidays) affect how patients measure (sprint 8)

2.4 Won't have

- Static analysis over more than one person

3. Product backlog (first version with estimates and prioritized user stories)

3.1. User stories of features

As a User, I can determine the output data set.

As a User,
When the input data is read,
Then I can filter on specified data.

As a User,
When the input data is read,
Then I'm able to use all different Sequential data analysis methods.

As a User,
When I selected the data to output,
Then I can format my .txt file for analysis tool SPSS.

As a User,
When the input data is read,
Then I'm able to export visualizations of the imported data.

As a User,
When the input data is read,
Then I can select the data to check if patients follow the websites advice according to the data of the stat sensor.

As a User,
When the input data is read,
Then I can select the data to check the time difference between when the patient measured the creatinine values and when the patient has entered the creatinine values on the website.

As a User,
When the input data is read,
Then I can select the data to check in what conditions patients override their measured data.

As a User,
When the input data is read,
Then I want to be able to select data to check if external factors like holidays affect how patients measure.

As a User,
When the input data is read,
Then I'm able to select the data to check how many times patients measure before they enter the data on the website.

As a User,
When the input data is read,
Then I can select the data to check if patients measure more if other factors like blood pressure increase.

As a User,
When the input data is read,
Then I can select the data to find cases where the website advised to contact the hospital.

As a User,
When I start the program,
Then I'm able to input data which could be compared to the data of the ADMIRE project.

4. Roadmap

Sprint 1

- Planning and draft of product vision.
- Programming environment setup.
- Creating initial architecture.

Sprint 2

- Finalizing product vision
- Product planning draft
- Interface creation
- Data management

Sprint 3

- Initial must have implementation
- Adding initial functionality to GUI

Sprint 4

- Adding must have functionality
- Adding should have functionality

Sprint 5

- Adding filter and comparison functionality

Sprint 6

- SPSS output file

Sprint 7

- Add last should haves

Sprint 8

- Finalizing all must-haves, should-haves and GUI

5. Definition of Done (backlog items, sprints, releases)

Here we will discuss when features, sprints and the end product are really done. We focus on the most important parts before we can say any of these are done.

A feature of our system is finished, when a backlog item is completely processed. This means that it's implemented, fully documented and tested. To check if the feature is well documented we use Checkstyle to check for any errors in the documentation and code formatting. With Maven we check if the 75% test coverage is achieved. It's important to have well tested code to avoid software bugs. When that happened we push the feature to Github in a separate branch. When the new feature is on Git, we let two other group members review the code and if they also think it's correct and well implemented the feature will be merged with the main branch and we can say the feature is done.

A sprint for our product is finished when all specified tasks of the sprint plan of the given week are done. This means all features are done in the manner described above and all deliverables are delivered. Also all features should be merged with the main branch of the program and we have a working program which passes all written tests. After that we can say our sprint is done.

The product is finished if it's a working program which passes all tests written. Exactly the same as a finished sprint. Also it should be the Static Analysis tool which the customer had in mind. Besides that the program should pass all tests and works, we want to test the end product with researchers. So we can see how they feel about using the program. To call the product really done, we need at least all must-have features implemented in the program. If any of the must-have features are missing, we don't have a working tool for researchers to use. The should have features are also a requirement of the product, but they are not necessary in order to have a working tool. Though our goal is to implement at least 50% of all should have features into the product.

6. Glossay

The 8 C's - The 8 C's is short for the 8 sequential data analysis methods: Chunking, Comments, Codes, Connections, Comparisons, Constraints, Conversion and Computation.

GUI - Graphical User Interface