Product Planning

Group Health informatics-2

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

Contents

Introduction

To study the behavior of kidney transplant patients, we need to study all kind of little details that were measured. Some details were measured by input in a ‘statsensor’ device. Other details were measured by hospital visits or feedback the patient was given by the ‘admire’ website.
All these details are stored in different kinds of files with different encodings. This becomes a problem when researchers want to do statistical analysis over this data. Also, not every detail is important for inserting into statistical analysis tools, but some details together can result in new event or new occurrences.
This program will be able to read all these different kinds of files and provide a query language for the 8 C’s (see glossary). The format of the output of this program will be valid input for SPSS (see glossary).
This way researchers can do quick analytics in an intuitive way to do their research in an easy manner.
At the end, the program will not be specifically for the ‘admire’ project but be as generic as possible, so it can analyze much more than only the data of the ‘admire’ project. However the main priority lies with the ‘admire’ project and the program’s must haves are based on the needs of the users of the ‘admire’ project.

Product

1. High-Level Product Backlog

These features describe very high-level what the program will be able to do, when every feature is implemented. All these features are likely to be split into smaller features, and these are in the MoSCoW model in the next section. There will be explained which feature is a must have or should have.

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

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

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

Won't haves: Features that will not be implemented in this project.

2.1 Must haves

- Program is able to format a .txt output file for analysis tool SPSS(sprint 3).
  This is a must have, because the main goal of the program is to format data in such a way it can be analysed by an analysis tool such as SPSS.
- Program is able to read multiple files which could be compared to the data of the ADMIRE project(sprint 3)
  This is a must have, because the main goal is to format the data of the input files in the way researchers want it.
- Program is able to filter data (constraints). (sprint 3)
  This is a must have, because researchers must be able to filter certain data.
- Program is able to compare values of input files. (sprint 4)
  This is a must have, because researchers must be able to compare input data.
- Program is able to chunk on data sets (sprint 4)
  This is a must have, because some parts of input files might not be useful for the researchers to analyze.
- Program is able to run an algorithm or input standard values to be compared with actual data. (sprint 5)
  This is a must have, because the data in itself does not always says something useful. Researchers want to compare en process the raw data, for it to become usefull information.

- Program is able to encode certain values (codes).(sprint 6)
  This is a must have, because you can only extract meaning from data, if you can identify certain data entry's or chunks.
- Program is able to count how many times values/events occur (computation) (sprint 6)
  This is a must have, because this is one of the essential operations that researchers have to be able to use on the data.
- Program is able to add comments to the data set output (comments). (sprint 7)
  This is a must have, because the researchers have to be able to add comments to give extra meaning to certain data entry's.

2.2 Should haves

- Program can output a user defined data set (sprint 6)
  This is a should have, because we can output already for SPSS in a .txt file. So it would be nice to output other file formats, but it's not necessary for the program to work.
- Program is able to connect different values as one event (connections). (sprint 7)
  This is a should have, because connections is an advanced sequential data analysis method, which the researchers should be able to use, but in absence it does not harm the program.
- Program is able to create new values from actual data set (conversions). (sprint 8)
  This is a should have, because conversions is an advanced sequential data analysis method, which the researchers should be able to use, but in absence it does not harm the program.
- Program is able to output if patients measure more if other factors like blood pressure increase (sprint 7)
  This is a should have, because this is one of the questions the researchers of the ADMIRE project will have.

2.3 Could haves

- Program is able to export visualizations of the imported data (sprint 7)
  This is a could have, because it would be nice for researcher to be able to export visualizations, but not necessary.
- Program can output whether external factors (holidays) affect how patients measure or not (sprint 8)
  This is a could have, because it's one of the question researchers of the ADMIRE project might have.

2.4 Won't haves

- Program can do static analysis over more than one person.
  This is something which could be implemented after this project, but it's unrealistic to implement it in the given time.

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

3.1. User stories of features

As a User,
I want to read my specified input data,
so I can put a filter or constraint on my specified data.

As a User,
I want to read my specified input data,
so I'm able to use all different Sequential data analysis methods on the input data.

As a User,
I want to select the data to output,
so I can format a .txt file for analysis tool SPSS.

As a User,
I want to read my specified data,
so I'm able to export visualizations of the data.

As a User,
I can select the input data,
so I can see if patients follow the websites advice according to the data of the stat sensor.

As a User,
I can select the input data,
so I can see 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,
I can select the input data,
so I can see in what conditions patients override their measured data.

As a User,
I can select the input data,
so I can see if external factors like holidays affect how patients measure.

As a User,
I can select the input data,
so I can see how many times patients measure before they enter the data on the website.

As a User,
I can select the input data,
so I can see if patients measure more if other factors like blood pressure increase.

As a User,
I can select the input data,
so I can 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 haves implementation
- Adding initial functionality to GUI

Sprint 4

- Adding must have functionality (initial)
- Updating architecture design for must-haves/should-haves

Sprint 5

- Implementing more must have functionality

Sprint 6

- Further GUI implementation
- Query language interpreter
- Adding initial should-have functionality

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 is the Static Analysis tool which the customer had in mind. Besides that the program passes 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
SPSS - A predictive analytics-software tool created by IBM.