Product Planning

Health Informatics 3

May 5, 2015

1 Introduction

TODO

2 Product

TODO an introduction about his section

2.1 Product Backlog

TODO ... For the product backlog we use the MoSCoW method.

2.1.1 Must Haves

- Language for description the analysis
- Reading in the analysis description file
- Reading in the data description file
- Indicating the data connections between the different datafiles
- Indicating the meaning of the various data inputs
- Reading in data from different sources using the data description file
- The 8 C's for data analysis
  - Chunk analysis
  - Comments
  - Codes
  - Connections
  - Comparisons
  - Constraints
  - Conversions
  - Computations
- Specifying the output and output format

- Visualizations from the data analyzed
  - Frequency bars
  - Line graph
- Manual for the analysis description language

2.1.2 Should Haves

- Visualizations
  - Box plot
  - Stem leave
  - State transistion matrix
  - Lag analysis
- Exporting the visualizations to images
- Preimplemented codes in our analysis description language

2.1.3 Could Haves

- Visualizations
  - Histogram
  - Markov chain graph transition diagram dingen
- Editor for inputting the analysis description
- Mass input for batch processing
- Preview of the output from the analysis

2.1.4 Would Haves

- Handy GUI for specifying the analysis

2.2 Roadmap

This section will describe the planning for the product. The roadmap is based on sprints of one week. The numbers of the week correspond to the week of quarter 4. A new iterations starts on every Friday. For each week we will list which features the product should have and which additional task must be done.

Ik weet niet of die nu de roudmap of de realese plan is

2.2.1 Week 4.1

- Setup the software that is used during the project.
- Obtain the requirements.

2.2.2 Week 4.2

- A basic architecture for the product.
- A design for the user interface.
- A draft version of the product vision.

2.2.3 Week 4.3

- A minimal user interface according the design of week 4.2.
- The final version of the product vision.
- A draft version of the product planning.
- The user must be able to specify in a data description file how a file should be read by the program.
- The user must be able to specify which data must be written to file.
- The user must be able to perform constraint analyses.

2.2.4 Week 4.4

- The final version of the product planning.
- The user must be able the perform chunking analyses.
- The user must be able the perform connections analyses.
- The user must be able the perform computation analyses.

2.2.5 Week 4.5

- The user must be able the perform codes analyses.
- The user must be able the perform comparisons analyses.
- It must be possible to show the data as frequency bars.
- It must be possible to show the data as a line graph.

2.2.6 Week 4.6

- The user must be able the perform comments analyses.
- The user must be able the perform conversions analyses.
- It must be possible to show the data as a box plot.
- It must be possible to show the data as a Stem-and-Leaf plot.
- Input for SIG

2.2.7 Week 4.7

- multiple file input
- It must be possible to show the data as a state transition matrix.
- It must be possible to show the data as a Stem-and-Leaf plot.
- The user must be able to export the visualizations as an image.
- It must be possible to show the data as a Histogram.

2.2.8 Week 4.8

- Show the data with Lag analysis
- It must be possible to show the data as a Markov chain.
- Simple user interface for specifying the analyses
- Implement certain analyses functions in our language

2.2.9 Week 4.9

This is the last week where it is possible to work on the code. This week will add no new features. In this way we will be able to handle some delay during the process. Furthermore this week is used to repair the last bugs. Therefore there is a feature freeze on Wednesday June 17.

- Final input for SIG
- Draft version of the final report

2.2.10 Week 4.9

- Final report
- Product presentation

3 Definition of Done

In this section we will discuss when a task is considered as done. In general a task is done when there is nothing left to do for that task. We will discuss the definition of done for backlog items, sprints and releases.

3.1 Backlog Item

A backlog item is done if it is implemented as described and it follows the description of the user stories. Furthermore the code should have been tested with unit tests. All the other features should still work and all the test should pass. The code must be reviewed by at least two persons who has not worked on that specific item. The code should be clear and when needed, it should contain comments. Furthermore the code should follow the languages conventions and it should have clear names for the variables. When the item meets all these requirements, than it is considered done.

3.2 Sprint

Each sprint should have a sprint plan and a sprint reflection. Any deliverable that has a due in or at the end of the sprint should have been made and hand in. Furthermore if needed, relevant documents, such as the architecture design, should have been updated. Critical bugs and errors that are discovered during the sprint should be fixed. If it is not possible to fix them during the sprint, than they have to be solved in the next sprint. Finally all the task of the sprint should be completed as described in the previous section. If it is not possible to complete a certain task in a sprint, than the sprint reflection should explain why it is was not possible to finish the task.

3.3 Release

Each sprint ends with a new version of the product. Sections 2.2 and 3.5 give an overview of the planned features for each release. Based on that, each sprint will ad some new features to the product. A release is only allowed to contain features that are considered done, see section 3.1 Therefore all the features in a release are tested and the code should be proper. Additionally we must test whether the features work correctly together. Furthermore a release may not have any critical bug or error. Finally for each release a demo has to be prepared and demonstrated.