Product Planning

TI2806 Contextproject
Health Informatics

Group A

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Contents

1

INTRODUCTION

Researchers at the LUMC have developed a system that makes the detection of the signs of rejection of a donor kidney after a kidney transplantation easier. [6]

The goal of their system is that the patients have more control about their own recovery process and will not need as many hospital visits.

The system is mainly based on the Creatinine level in the patient's blood. This level can be obtained by testing the patient's blood with a portable machine. Then this information is entered into a website (mijnnierinzicht.nl), which gives the patient feedback about his/her current health status.

This data is what we want to be able to analyze, in order to learn how the patients execute the self management system in practice and how they follow the advice offered by the computer system. Several possible user patterns we want to look at are the kind of mistakes that are made with entering the obtained information in the website, how people abide by their testing schedule, how they abide by the advice offered by the system, or whether they submit dummy values for example. [2]

Our goal is to create a stand alone program that can help analysts analyze this data.

2

PRODUCT

According to Kenneth S. Rubin, the goals of product-level planning are to capture the essence of a potential product and to create a rough plan for the creation of that product. Envisioning begins with the creation of a vision, followed by the creation of a high-level product backlog and frequently a product roadmap.[4]

2.1. HIGH-LEVEL PRODUCT BACKLOG (SET OF EPICS ALIGNED WITH THE PRODUCT VISION)

Epics are typically stories that include too many unknowns to tell how big it is, or when the requirements are known but its effort is too huge to complete in a single sprint. [1] So for our set of epics, we have defined a list of backlog items that are too big to complete in a single sprint, but which we will eventually split into smaller tasks. Our product backlog items include the following epic-level user stories:

- As a user I want to be able to input files of an excel and text format.
- As a user I want to get an output file that I can use as input for other statistical programs.
- As a user I want to to able to apply sequential data analysis operations on the data-set.
- As a user I want to be able to create several visualizations of the data-set.

2.2. ROADMAP (MAJOR RELEASE SCHEDULE, RELEASE GOALS)

Sprint 1

- High level system design
- Organize our overall vision for the product

Sprint 2

- Implementation of the data-structure
- Finalize our product vision
- Create configuration XML files
- Work out a scripting language for the instructions

Sprint 3

- Create a working parser for the input files
- Create objects based on the processed input
- Finalize our product planning

Sprint 4

- Start on designing a more concrete version of our User Interface
- Connect the UI to the selection of a configuration XML

Sprint 5

- Plan implementation of the 8 c's

Sprint 6

- Create modifications of the data sets for exportable output

Sprint 7

- Create concrete list of the visualizations that the user wants to to able to obtain from our system
- Create these visualizations

Sprint 8

- Finalization of a very user friendly and simple UI
- Make the distinction between what we want to include in the script and what we want to translate into radio buttons, etc. In order to create a good balance between user friendliness and simplicity of the system.

3

PRODUCT BACKLOG

According to Mario Moreira [3], "Product Backlog is a repository for user stories and other Product Backlog Items (PBIs) such as tasks, epics, and themes. The Product Backlog is the singular place to store all PBIs related to the product. Most Product Backlogs are either a form of document or in an agile planning product that offers automation to manage PBIs."

3.1. USER STORIES OF FEATURES

For the requirements defined in our Product Vision, we define user stories. To prioritize the backlog items, we categorize them according to the MoSCoW method.

3.1.1. MUST

- As a user,
  When the application has started,
  And I am in the main screen,
  I want to be able to input a configuration XML file

- As a user,
  I want to be able to input XLS and TXT data-set files

- As a user,
  When my input files have been entered into the system,
  I want to be able to apply chunking, coding, connecting and constraining operations on the data-set

- As a user,
  when I have manipulated the data-set,
  I want to be able to visualize the data-set in a Frequency diagram

- As a user,
  when I have manipulated the data-set,
  I want to be able to visualize the data-set in a Stem-and-Leaf plot

- As a user,
  when I have manipulated the data-set,
  I want to be able to visualize the data-set in a Box-and-Whiskers plot

- As a user,
  when I have manipulated the data-set,
  I want to be able to visualize the data-set in a State transition-matrix

3.1.2. SHOULD

- As a user,
  when manipulating the data-set,
  I want to apply conversion operations
- As a user,
  when manipulating the data-set,
  I want to apply comparison operations
- As a user,
  when manipulating the data-set,
  I want to apply constraints operations

3.1.3. COULD

- As a user,
  when I have manipulated the data-set,
  I want to visualize the data-set in a histogram
- As a user,
  when I have manipulated the data-set,
  I want to visualize the data-set in a Markov chain graph
- As a user,
  when I have manipulated the data-set,
  I want to visualize the data-set as time series (2D)

3.2. USER STORIES OF KNOW-HOW ACQUISITION

- Create several concepts for the way in which the manipulations on the data-set are defined. Discuss with the user for usability feedback.
- Create several concepts for linking the files and the XML's in the User Interface. Discuss with the user about his/her preference.

3.3. INITIAL RELEASE PLAN

The initial release plan is based on the sprint planning described in the previous chapter.

Sprint	Milestone
Sprint 1	High-level system design.
Sprint 2	Design of data structure and input module.
Sprint 3	System is able to parse input and generate the data structure.
Sprint 4	User can select a configuration XML in the GUI.
Sprint 5	Sequential data analysis can be performed using several operations.
Sprint 6	Modified data-set can be exported to different formats.
Sprint 7	Data can be visualized in several different graphs and plots.
Sprint 8	Implementation of input method for the data analysis.

4

DEFINITION OF DONE

According to Ken Schwaber and Jeff SutherlandSchwaber and Sutherland [5], "When a Product Backlog item or an Increment is described as 'Done', everyone must understand what 'Done' means. Although this varies significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of 'Done' for the Scrum Team and is used to assess when work is complete on the product Increment". In this chapter we define for our project the Definition of "Done" on three levels: backlog items, sprints and release.

4.1. BACKLOG ITEMS

We call a backlog item done when it meets the following requirements:

- Acceptance criteria of the related user story are met
- Code has been completed, re-factored and commented
- Code has been inspected by the fellow group members
- Code has been tested using unit tests
- The implemented feature has passed all related acceptance tests
- End-user documentation has been updated to reflect changes to the product

4.2. SPRINTS

We call a sprint done when it meets the following requirements:

- All backlog items included in the sprint are done
- Documentation and diagrams have been updated to reflect changes to the product
- Build is done without errors

4.3. RELEASE

We call the release done when it meets the following requirements:

- All sprints included in the release are done
- All Must-Have requirements have been met
- Product passed UAT

5

GLOSSARY

- UAT - User acceptance testing

BIBLIOGRAPHY

[1] Admin (2015). Scrum epics. http://scrummethodology.com/scrum-epics/. [Online; accessed 7-May-2015].

[2] Brinkman, W. P. (2015). Analyse tool voor zelfmanagement gedrag van niertransplantatie patiënten.

[3] Moreira, M. (2013). *Being Agile: Your Roadmap to Successful Adoption of Agile.* Apress.

[4] Rubin, K. S. (2013). *Essential Scrum: A Practical Guide to the Most Popular Agile Process.* Pearson Education, Inc.

[5] Schwaber, K. and Sutherland, J. (2013). The scrum guide. http://www.scrumguides.org/scrum-guide.html#artifact-transparency-done//. [Online; accessed 6-May-2015].

[6] van Lint, C., van der Boog, P., Romijn, F., van Dijk, S., Schenk, P., and Cobbaert, C. (2014). Zelfmonitoring van nierfunctie na niertransplantatie: de patiënt als regisseur.