Product Planning

TI2806 Contextproject
Health Informatics

Group A

Technische Universiteit Delft

TU Delft
Delft University of Technology

Challenge the future

Contents

1

INTRODUCTION

1.1. SECTION

2

Product

According to Kenneth S. Rubin, "The goals of product-level planning (which I also refer to as envisioning) 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 chunking, coding, connecting and constraining 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)

The concrete planning has yet to be defined, but below you will find the general overview.

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 out 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 MoreiraMoreira [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."
In this chapter we define user stories according to the following template provided by Mike Cohn in User stories applied[2]:
I as a (role) want (function) so that (business value)

3.1. User stories of features

-

3.2. User stories of defects (if applicable)

3.3. User stories of know-how acquisition

-

3.4. 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] Cohn, M. (2004). *User stories applied.* Addison-Wesley.

[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.*

[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].