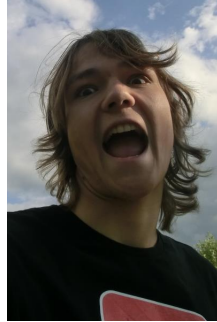


CONTEXTPROJECT PROGRAMMING LIFE EMERGENT ARCHITECTURE DESIGN

TU DELFT



Ruben Bes
rbes
4227492



Mathijs Hoogland
mhhoogland
4237676



Jasper Denkers
jdenkers
4212584



Robbert van Staveren
rhvanstaveren
1527118



Willem Jan Glerum
wglerum
4141040

May 15, 2014

Abstract

Product planning gives a great overview of the features that the application will have. Mainly following the MoSCoW method with Must, Should, Could, Would have. These will all be backed by their relevant user stories. Also we will give the definition of when a feature is really done.

Contents

1	Introduction	3
2	Product	3
2.1	High-level product backlog	3
2.1.1	Must haves	3
2.1.2	Should haves	3
2.1.3	Could haves	3
2.1.4	Would haves	3
2.2	Roadmap	3
3	Product backlog	4
3.1	User stories of features	4
3.2	User stories of defects	4
3.3	User stories of technical improvements	4
3.4	User stories of know-how acquisition	4
3.5	Initial release plan	4
4	Definition of Done	4
5	Glossary	5

1 Introduction

The product planning contains all the important information of the product, the product backlog, roadmap and user stories

2 Product

2.1 High-level product backlog

This section describes the desired features of the application according to the MoSCoW method [1]. Categorized into four groups:

- **Must have**s describe requirements that must be satisfied in the final solution
- **Should have**s describe high-priority requirement that should be included if possible
- **Could have**s describe requirements that are considered desirable but not necessary.
- **Would have**s describe requirement that stakeholders have agreed will not be implemented in the solution, but could be added in the future.

2.1.1 Must have

- Visualize trioddata of father, mother and child
- Looking for known disease mutations
- Retrieving data from existing genetic databases
- Reading VCF (Variant Call Format) files
- Easy to use GUI for doctors.

2.1.2 Should have

- Uploading VCF files to the server in the background

2.1.3 Could have

- Exporting visualization data

2.1.4 Would have

- Spread computational power over multiple threads, cores, or systems.
- Support for mobile web browsers.

2.2 Roadmap

Major release schedule, release goals.

3 Product backlog

First version with estimates and prioritized user stories.

3.1 User stories of features

Een kind heeft een ziekte, terwijl de ouders deze ziekte niet hebben. Een arts wil met behulp van ons product uitzoeken of een mutatie van een bepaald gen de oorzaak kan zijn van deze ziekte. Hiervoor heeft hij de volgende werkwijze:

De arts kiest een vcf bestand van de ouders en het kind om te analyseren. Hij krijgt een visualisatie waarin hij kan zien welke mutaties er in het dna van het kind zitten en of de locaties waar deze mutaties hebben plaatsgevonden worden geassocieerd met de ziekte van het kind. Uit de visualisatie leid hij af welke mutaties mogelijk de oorzaak zijn van de ziekte.

3.2 User stories of defects

(if applicable)

3.3 User stories of technical improvements

(if applicable)

3.4 User stories of know-how acquisition

3.5 Initial release plan

milestones, MRFs per release

4 Definition of Done

This section describes when a feature is really done and ready to be integrated in the system. We will define this for a feature, sprint and end product.

A feature is finished when it is fully tested and the code is accepted by other developers. These test should done with JUnit and the code coverage should be high enough. Furthermore the code must be fully documented with JavaDoc and should match the rules from CheckStyle.

A sprint is finished when the whole application is tested and approved, just like a feature. However the continuous integration system should also accept the build. Next the developers and users will test the system by hand to check for bugs.

The end product is finished if all the Should have's and Could have's are implemented and tested as described above, because only the should have's give a very basic application. The product should have the looks and feels approved by the stakeholders, they should be happy with the product.

Furthermore in addition to all this the code should be well documented, tested, style checked and integrated. This will all be evaluated by the SIG (Software Improvement Group) and should be improved after the first check.

5 Glossary

glossary

References

- [1] Kevin Brennan et al. *A Guide to the Business Analysis Body of Knowledge*. Iiba, 2009.