Blockchain Project: Product Planning

Bulls, Bears & Wolves

Jasper Hu, yjhu, 4356241
Naqib Zarin, nzarin, 4384474
Ashay Somai, asomai, 4366220
Ymte Broekhuizen, yjbroekhuizen, 4246586
Luat Nguyen, tlnguyen, 4467574

May 4, 2017

Table of contents

1. Introduction

An excellent product requires an excellent Product Planning. A product-level planning is needed to capture the essence of a potential product. Therefore, this document contains high level description of the rough planning over a horizon.

This document begins with Scrum Product Backlog which will be used in pair with the Product Vision to create the second item, the Product Roadmap.

At last a Definition of Done will be introduced, which is a checklist of the types of work that the team is expected to successfully complete before it can declare its work to be potentially shippable.

2. Product

The product that is going to be created will consist of a blockchain based database which will enable users to share data and identifications without the fear of being compromised. This will result in a network of trustworthy users, also called a web-of-trust.

2.1 High-level product backlog

ID	PBI Type	PBI Name	Description (in user story)
1	Feature	Android Application	As a user I want to be able to install and run the web-of-trust application on my android device(s). The application should have a simple and usable interface.
2	Feature	Adding a contact via Bluetooth	As a user I want to be able to connect my device with someone else's device via Bluetooth for their contact information.
3	Feature	Using blockchain model to acknowledge contact(s)	As a user I want to be able to let the blockchain acknowledge and store my contact(s).
4	Feature	Rating contact(s)	As a user I want to be able to rate the trustworthy of others “save” it into the blockchain.
5	Feature	Remove contact information	As a user I want to be able to remove the contact from my list, blockchain will thus add another block that will acknowledge that decision.
6	Feature	Overview of contact(s)	As a user I want to have an overview of users in which I or my friends (contacts that I have connected in person) have as contacts.
7	Feature	Application (automatically) update contact list	As a user I want my application to (automatically) update my web-of-trust contact list information automatically (by updating the blockchain).

2.2 Roadmap

Releases	Item description	Release Date
0.2	A simple runnable Android application should be there to provide product bare framework (files structure to program the blockchain into)	12/05/2017
	The barebone version of the blockchain database should be provided, this version should be at least be able to add data (contact public keys) into the blockchain. (not necessary via the application interface)	
0.4	The blockchain database should be able to "remove" data from the blockchain when a public key is revoked or a contact is removed. (not necessary via the application interface)	26/05/2017
	There should be an overview of the contacts available (either on the programmer terminal or on an external file)	
0.8	The application should be able to 1: automatically update the blockchain and displays the contacts on the user interface, 2: user is able to add and remove contact through the interface. At this stage a user friendly interface becomes important	09/05/2017
	Bluetooth should be implemented into the product: the users should be able to use bluetooth to share each other contact information (public key, name, etc…)	
0.9	The users of the application should be able to rate the trustworthiness of their contacts via the application and see his/ her own rating.	16/05/2017
1.0	Final version of the product, the product should contain: - Userfriendly interface - Passed the Definition of Done checklist	23/06/2017

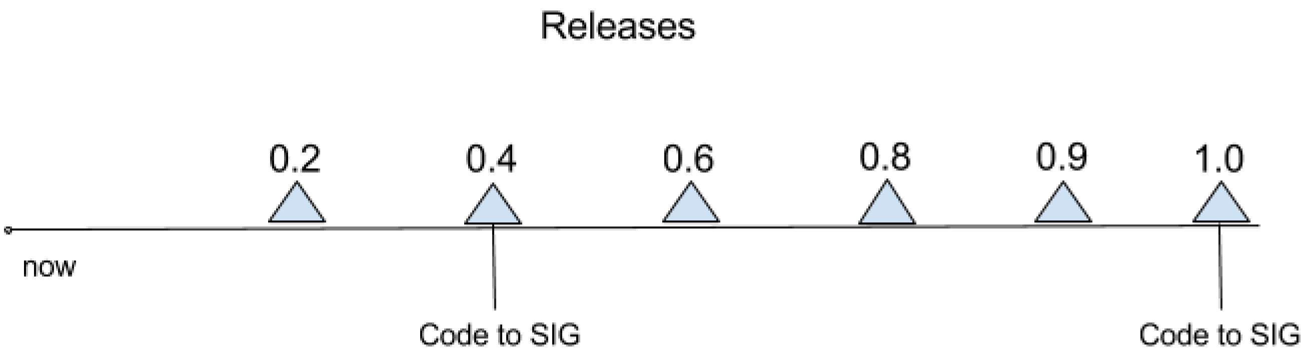

Figure 1.0: Roadmap visualization

3. Product Backlog

3.1 User stories of features

ID	PBI	Why	Given	When	Then	Est in hours	Priority
1	Android Application	Because I want to provide a barebone (files) framework so the team has the place to start with	I have not ran the application	I start the application	There should be a simple user interface presented to me with functional buttons. (Not necessarily user friendly yet)	15	A
2	Adding a contact via Bluetooth	Because I want to receive contact information (name, publickey) when I bluetooth pair with another person	My bluetooth is on and I am presented with a list of pairable partner near me	I choose select the person I want to pair in the bluetooth discoverable list	My device will send the selected device my public key and vice versa. Contact is added into my contact list.	5	E
3	Using blockchain model to acknowledge contact(s)	I want the program to add my contact into my database (not necessary via the user interface yet)	I have the public key of the contact I want to add	I have the contact's public key	The history of this transaction is mined into its blockchain	25	B
6	Overview of contact(s)	I want to have an overview of the contacts available (not necessary via the user interface yet)	I am in my programming terminal	When run show contacts	A list of my contacts and their contacts should be displayed on the screen	15	C
5	Remove contact information	I want my application to update the blockchain when I have removed a contact (not necessary via the user interface yet)	I am in my programming terminal	When I run the contact delete command	The blockchain should acknowledge that I am no longer contact with the person.	20	C
7	Application to (automatically) update contact list	I want my application to update and display the changes in my contacts on the application interface	I am using my application	When I make changes to a contact (add, remove, rate) the application should update the interface accordingly	My app contact list will take a look at the blockchain and update my contact list accordingly	10	D
4	Rating contact	I want to rate my contact trustworthiness	I have rated/edited my trustworthy rating for a contact	When I click save rating	My app will calculate the average rating of that contact and update it on the blockchain.	20	D

3.2 User stories of know-how acquisition

PBI	Scenario	Est.	Priority
Product planning	As a developer I want to make a clear Product Planning because a product-level planning is needed to capture the essence of a potential product. The product planning is a high-level description of the product backlog, roadmap and release schedules.	6	A
Product vision	As a developer I want to make a clear product vision because I need a clear description of the areas in which the stakeholders, such as users and customers, get value. In other words, this is an elaborated description and vision of the final product itself.	7	A
Architecture design	As a developer I want to design a (prototype) architecture design with the main idea: the client keys are stored securely in the blockchain.	13	A

3.3 Initial release plan

Releases	Item ID	Item description	Release Date
0.2	1	A simple runnable Android application should be there to provide product bare framework (files structure to program the blockchain into)	12/05/2017
	3	The barebone version of the blockchain database should be provided, this version should be at least be able to add data (contact public keys) into the blockchain. (not necessary via the application interface)	
0.4	5	The blockchain database should be able to “remove” data from the blockchain when a public key is revoked or a contact is removed. (not necessary via the application interface)	26/05/2017
	6	There should be an overview of the contacts available (either on the programmer terminal or on an external file)	
0.8	7	The application should be able to 1: automatically update the blockchain and displays the contacts on the user interface, 2: user is able to add and remove contact through the interface. At this stage a user friendly interface becomes important	09/06/2017
	2	Bluetooth should be implemented into the product: the users should be able to use bluetooth to share each other contact information (public key, name, etc…)	
0.9	4	The users of the application should be able to rate the trustworthiness of their contacts via the application and see his/ her own rating.	16/06/2017
1.0	Final	Final version of the product, the product should contain: - Userfriendly interface - Passed the Definition of Done checklist	23/06/2017

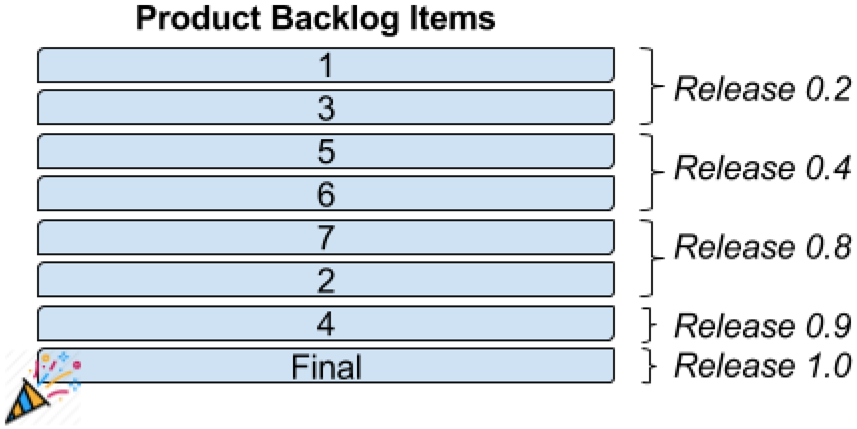

Figure 2.0: Product roadmap releases visualization

4. Definition of done

4.1 Weekly sprint checklist

This is a checklist that is used to check the code after each weekly sprint is finished.
Before each sprint's work is considered shippable and is pulled into the master branch, it has to go through this checklist.

Weekly Sprint Checklist	
☐	Code completed (all todo items completed)
☐	Code is commented
☐	Code inspected
☐	Code is peer reviewed
☐	Code tested
☐	Unit tests pass
☐	Integration tests pass
☐	Regression tests pass
☐	Zero checkstyle error
☐	Documentation finished
☐	Sprint retrospective
☐	Sprint backlog for next iteration
☐	Architecture design updated
☐	Running prototype for the Demo

4.2 Final product checklist

This is a checklist consisting of the types of work that the team is expected to successfully complete before it can declare the final product as shippable.

Definition of Done (Final Product)	
□	Architecture reviewed
□	Code completed
□	Code refactored
□	Coded in standard format
□	Code is commented
□	Code inspected
□	All to do items are completed
□	Code tested
□	Unit tests pass
□	Integration tests pass
□	Regression tests pass
□	Zero known bugs
□	Builds without error
□	Acceptance tested (meets all requirements)
□	Documentation added

5. Glossary

A

Acceptance test. A test that defines the business value each product backlog item must deliver. It may verify functional requirements or nonfunctional requirements such as performance or reliability. It is used to help guide development.

Android App. An Android app is a software application running on the Android platform. Because the Android platform is built for mobile devices, a typical Android app is designed for a smartphone or a tablet PC running on the Android OS.

B

Blockchain. A blockchain is a public ledger of all Bitcoin transactions that have ever been executed. It is constantly growing as 'completed' blocks are added to it with a new set of recordings. The blocks are added to the blockchain in a linear, chronological order.

D

Definition of done. A checklist of the types of work that the team is expected to successfully complete by the end of a sprint, before it can declare its work to be potentially shippable.

I

Integration test. Integration testing, also known as integration and testing (I&T), is a software development process which program units are combined and tested as groups in multiple ways.

P

Product Backlog. In the simplest definition the Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories.

PDI. Product Backlog Item. See *Product Backlog.*

R

Regression test. Regression testing is the process of testing changes to computer programs to make sure that the older programming still works with the new changes.

U

Unit test. Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated**.**

S

Shippable. A shippable product is one that has been designed, developed and tested and is therefore ready for distribution to anyone in the company for review or even to any external stakeholder. Adhering to a list of “done” criteria ensures that the sprint product is truly shippable.

W

Web-of-trust. An informal mechanism for determining the validity of public keys, especially for PGP users

6. Bibliography

Conoscenti, Marco, Vetro, Antonio and Carlos de Martin, Juan. 2016. *Blokchain for the Internet of Things: A Systematic Literature Review.* Retrieved from https://nexa.polito.it/nexacenterfiles/Blockchain%20for%20the%20Internet%20of%20Things%3A%20a%20Systematic%20Literature%20Review.pdf

Crispin, Lisa and Gregory, Janet. 2011. *Agile Testing: A Practical Guide for Testers and Agile Teams.* Pearson Education, Inc.

DaVeiga, Nada. 2008. Change Code Without Fear: Utilize a Regression Safety Net. Retrieved from: http://www.drdobbs.com/tools/change-code-without-fear/206105233

Fowler, Martin. 2014. *Unit Test*. Retrieved from: https://martinfowler.com/bliki/UnitTest.html.

Gupta, Mayank. 2008. *Definition of Done: A Reference.* Retrieved from: https://www.scrumalliance.org/community/articles/2008/september/definition-of-done-a-reference

Penning, Henk P. 2017.*Analysis of the strong set in the PGP web of trust.* Retrieved from https://pgp.cs.uu.nl/plot/ .

Rubin, K. S. 2012. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley.

https://www.atlassian.com/agile/backlogs

http://searchsoftwarequality.techtarget.com/definition/integration-testing

https://techterms.com/definition/android