[TI2806] Product Planning: Interactive Monitor of Tribler Network

BlockchainBoys

May 11, 2017

Student Name	**NetID**
Yinghao Dai	yinghaodai
Max van Deursen	mvandeursen
Geert Habben-Janssen	ghabbenjansen
Bram van den Heuvel	bramvandenheuv
Ruben Keulemans	rkeulemans
Tim Speelman	tspeelman1

Table 1: Because this document is publicly available, the student numbers are not specified.

Abstract

This piece documents the description of the planning which is used whilst creating the visualization of the Tribler network. This is done by describing the product backlog in two ways, as well as giving a release plan for the product. This piece covers the definition of done which will be used to determine the state of backlog items as well.

Contents

1 Introduction

"Tribler is an open source decentralized BitTorrent client which allows anonymous peer-to-peer by default"[1]. Tribler uses advanced techniques in order to regulate the traffic on its network, managed in a decentralized way. This all happens in the background, without offering the user a comprehensive and accessible way to view this process. The described product in this piece will offer a solution to this problem, by creating a visual representation of the local network of the user whilst displaying a variety of information metrics about the network.
This piece serves as a high-level description of the planning which is used whilst creating the product. First off the high-level product backlog is introduced, which illustrates the idea of the product in the form of epics. These epics are used to sketch our roadmap, where our release schedule and goals are explained. After this, the product backlog is elucidated. The second to last chapter explains the definition of done, which is used throughout the process.

2 Product

This chapter introduces both the epics of the product, as well as the release schedule of the product. This release schedule uses the described epics in order to correctly describe the minimum requirements for the product at the day of release.

2.1 High-level product backlog

The backlog items are described with user stories.

- As a user, I want to get insight into the trust mechanism so that I can gain greater understanding of the mechanism
- As a user, I want to see a visual representation of the network focused around me, so that I can get a visual image of the network
- As a user, I want to access the visual representation through a button in the Tribler window so that I can easily access the visual representation
- As a user, I want to be able to identify clusters in the network so that I can easily identify Sybil attacks in the network
- As a user, I want a clear set of statistics such as download, upload and pagerank available to me so that I can understand the factual data which is transferred between nodes efficiently
- As a user, I want to change the node that is focused so that I can easily browse through the network at hand
- As a user, I want to be able to filter the network on certain data (such as temporal) so that I can reduce the information given to me and instead focus on a certain part of the network

2.2 Roadmap

The first major release (the twelfth of May) plans on showing the first basic visualization as a demonstration of the essential functionality of the product. This includes a very basic display of the nodes and edges within the network for a limited level of neighbors. Moreover, the interactivity is limited to selecting another focus node through selecting a node from the focus box. This release is planned at the twelfth of May.
The intermediate release (the second of June) will feature a basic version of the final product; although the interaction will be through clicking in the visualized graph itself and some more detailed information is shown than in the first major release, the advanced features such as temporal filtering and advanced filtering are not yet included. This is to be released on June second.
The last major release (the twenty third of June) will feature the final product with as much of the above features as possible. Every Friday between the twelfth of May and the twenty third of June, a release is scheduled for the features we have worked on that week.

3 Product backlog

3.1 User stories of features

- As a Potential Tribler user, I want to understand why my reputation in the network affects my download speed so that I can better understand the trust mechanism of Tribler.
- As a Potential Tribler user, I want to know how I can improve my reputation in the network so that I can retrieve potential higher download speeds whilst using Tribler.
- As an existing Tribler user, I want to have detailed and extensive insight into my network of peers so that I can correctly understand the network that I am in.
- As an existing Tribler user, I want to compare myself against other users so that it becomes apparent how am I situated within the network.
- As a Tribler administrator, I want to have detailed and extensive insight into my network of peers so that I can analyze the network and see its behavior under certain policies.
- As a Tribler administrator, I want to be able to detect clusters of peers so that I can check whether these clusters are a sybil attack.
- As a Researcher, I want to be able to use the data in this network visualization so that I can conduct research on the subject matter.

3.2 User stories of defects

- As a user of the visualization, I want to continue with using Tribler if something went wrong whilst loading the visualization so that I do not have to restart Tribler in order to do so.

3.3 User stories of technical improvements

Since the written code is unique and there's no previous code, there are no user stories available for the technical improvements.

3.4 User stories of know-how acquisition

- As an existing or Potential Tribler user, I want to be taught the functionality of the visualization through some kind of tutorial so that I can efficiently use the visualization.

3.5 Initial release plan

The first release will be at the twelfth of May, where the essential features will be first presented to the product owner. This release will include the features described as must-haves in the Product Vision document of the product, which include:

- A 2d visualization of the locally saved network. This means that the users in the network as well as their relations are visualized on a 2d plane

- A focus node in the visualization representing the user. This means that the centered node in the visualization is the selected node which is focused
- A visualization of the relationship between nodes representative of the data flow between those nodes
- A display of the trust ranking of a user in the overall network
- Regular (every 10 minutes) updates of the visualization
- User interaction by selecting different focus node for the visualization. This will be done through a basic way which does not require handling clicking events but instead relying on a method such as a dropdown menu or keyboard presses

The intermediate release at the second of June will include the features presented at the initial release, along with the following features:

- Visual cues which allow users to identify clusters in their local network
- User interaction by clicking on nodes, which will give the user additional options
- Regular updates (every 30 seconds) of the visualization

The deadline for the final release is on Friday the 23rd of June. At this date, the following features are expected to be implemented in addition to the features scheduled for the initial release:

- An advanced visualization (e.g. a 3d visualization or some other way of efficiently displaying a complicated network) of the local network
- The ability for the user to focus on different parts of the network more easily
- The ability to filter nodes / edges on different metrics. Think of filtering on a certain time period or a certain sub network
- Visual cues to help users identify free-riders and isolated clusters in the network

To complete the these tasks, the team is expected to spend 1680 hours. This is however including time spend on mandatory side activities like writing documentation and completing individual and group assignments concerning project and information skills and interaction design. The estimated time the group is able to spend on implementing features is 800 hours, being divided as such:

Feature	**Est. time**
Visualization of nodes and edges	120 hours
Focus node on the user	40 hours
Exploration of the network	80 hours
Visualization of local network in terms of nodes and edges	220 hours
Cues for clusters / free-riders	100 hours
Display of trust in overall network	40 hours
Regular updating of the visualization	120 hours
Filtering nodes / edges	80 hours

Table 2: Estimated time per feature

Since this is the first time this team works together, these estimates are solely based on the individual developer's experience, not on the recorded productivity of the team.

4 Definition of Done

In this chapter the definition of done will be explained. This will be done for backlog items as well as sprints and releases.

A backlog item will be considered done if it has completed the full process of our way of developing. This can be described as follows:

1. The developer develops the feature and writes tests whilst doing so, in a test driven development way.
2. The developer make sure that he has not introduced faulty software (i.e. all written tests still pass)
3. The developer submits a pull request to request a merge for his feature.
4. The pull request will be reviewed by at least one of the other developers, whilst leaving constructive criticism on the newly written code.
5. The developer rewrites and fixes any parts of code that were indicated by the reviews or discusses these parts with the reviewer at hand.
6. Repeat step 4 and 5 until there are no more code snippets indicated by the other developers.
7. After no more code snippets are indicated to be faulty, the feature is ready to be merged.

After the above steps are executed, a backlog item can be considered Done.
Moreover, if there is a backlog item which does not include writing code (e.g. documentation), the developer on the item will complete the item, whereafter the other developers will review the item and request changes if necessary. Then these changes will be fixed until no other changes are requested, after which the backlog item can be considered Done.

A sprint will be done at the end of each week, at Friday. A sprint is done when the the team has looked back at the past week in the form of a Sprint Retrospective, as well as making sure that all the backlog items planned are either completed or successfully migrated to the next sprint.

Lastly, a release is done whenever the backlog items which are planned for a certain release are completely Done, as described above. This is preferred to be at before the planned release date.

5 Glossary

epic "really large user stories that are consistent with the product level of planning" [2].

network "a network can be defined as a graph in which nodes and/or edges have attributes (e.g. names)"[3]. In this case, the nodes are users and the edges are transfer totals between two nodes (note that the edges are directed).

potential "possible when the necessary conditions exist"[4].

sybil attack a form of attack on a reputation system which forges multiple pseudo identities in order to exploit the vulnerability to a peer-to-per network.

test driven development "refers to a style of programming in which three activities are tightly interwoven: coding, testing (in the form of writing unit tests) and design (in the form of refactoring)"[5].

References

[1] Tribler. (n.d.). *Tribler Wikipedia Page.*. Retrieved May 1, 2017, from `https://en.wikipedia.org/w/index.php?title=Tribler&oldid=777445841`

[2] Ruben, K. S. (2012). *Essential scrum: A practical guide to the most popular agle process.* Boston, Massachusetts: Addison-Wesley.

[3] Newman, M. E. J. *The structure and function of complex networks.* Ann Arbor, Michigan: University of Michigan

[4] Cambridge Dictionary. (n.d.) *"potential" in British English.* Retrieved May 10, 2017, from `https://dictionary.cambridge.org/dictionary/english/potential`

[5] Agile Alliance. (n.d.) *Glossary: TDD.* Retrieved May 10, 2017, from `https://www.agilealliance.org/glossary/tdd/`