Product Planning

Draft version 0.1
Date: 28-4-2016

Introduction

Code everyday is becoming more and more complex. In order to avoid bugs and low software quality, code reviews are needed. However, whether the currently most used tool - pull requests - is the best tool for the job, is currently unknown and requires more research. Here is where *octopeer* comes in, it collects data on the usage of pull requests, allowing researchers to find ways to develop better tools, and developers to find their weaknesses and improve themselves.

Product

High-level product backlog

See also the requirements specification:
For the code review analysis tool 'Octopeer for Bitbucket' the functional requirements are grouped into four categories and can be identified using the MoSCoW[1] model for prioritizing requirements.

1.1. Must haves

- The extension shall track mouse movements on Bitbucket web pages.
- The extension shall track mouse positions on Bitbucket web pages.
- The extension shall track mouse clicks on Bitbucket web pages.
- The extension shall track the number of commits per pull request.
- The extension shall track the amount of lines of code added per commit per pull request.
- The extension shall track the amount of lines of code removed per commit per pull request.
- The extension shall store tracked data in a database.
- The extension shall support a web page with user statistics.

[1] DSDM Consortium, http://www.dsdm.org/content/10-moscow-prioritisation, July 2008

1.2. Should haves

- The extension shall track the timestamp of when a pull request is created.
- The extension shall track the timestamp of when a pull request is approved by a user.
- The extension shall track the timestamp of when a pull request is declined.
- The extension shall track the timestamp of when a pull request is merged.
- The extension shall track the HTML elements the mouse hovers.
- The extension shall have a settings page.
- The settings page shall have an on / off button for mouse tracking.
- The settings page shall have an on / off button for keystroke tracking.
- The settings page shall have an on / off button for commit message tracking.
- The settings page shall have an on / off button for meta data tracking.
- The settings page shall support editing the whitelist.

1.3. Could haves

- The extension shall track keystrokes on Bitbucket web pages.
- The extension shall track HTML element selections on Bitbucket web pages.
- The extension shall track keystrokes on Bitbucket web pages.
- The extension shall track the amount of approvals for a pull request.
- The extension shall track data on web pages apart from Bitbucket.
- The extension shall store a whitelist of trackable web pages apart from Bitbucket
- The extension shall track commit messages.

1.4. Won’t haves

- The extension shall not support web browsers other than Chrome.

Roadmap

- Sprint 1 (Week 2)
  - Loading extension
  - Cursor logging
  - Window size logging
  - Code wrapper for connection to database prepared.
- Sprint 2 (Week 3)
  - “Semantic” hover logging
  - Mouse clicks
  - Interface for sending events to the database finalized
  - Start UI design
  - Plan for code structure
  - Few components are tested

- Sprint 3 (Week 4)
    - Start testing logging component
    - “Semantic” on screen logging
    - Working ON/OFF global tracking toggle.
    - Enforced consistent code style
    - Most components tested
- Sprint 4 (Week 5)
    - Working ON/OFF toggles for privacy and tracking
    - All components (apart from UI) are tested

Data Analysis (No plans yet)

- Sprint 5 (Week 6)
- Sprint 6 (Week 7)
- Sprint 7 (Week 8)
- Sprint 8 (Week 9)
    - Final product

Product backlog

User stories of features

As a user I want to disable tracking by the extension.
As a user I want to choose what is tracked by the extension.
As a user I want to choose if my username is hashed.
As a user I want to choose if my data is encrypted by the extension.
As a user I want to get a report of the data that is collected.

User stories of defects [if applicable]

n/a

User stories of know-how acquisition

As a user I want the extension to be easy to learn how to use.

Initial release plan

[milestones, MRFs per release]

Definition of Done

Done Feature:

Level 1:

-All code is commented sufficiently
-All code meets the general coding standard
-All source coderelated to this feature has been committed to bitbucket
-The code is build without any errors
-Unit tests are written and passed
-The code performs the task it should perform and nothing else
-The code is covered by a minimum of 40% and all tests succeed, this does not include the UI.

Level 2:

-All code is fully commented
-The code is covered by a minimum of 70% and all tests succeed, this does not include the UI.
-Integration tests are written and passed
-No known bug is left in the feature
-The feature is peer reviewed by at least 2 team member other than the author.
-Diagram for the feature has been created and included in the full diagram architecture
-Quick guide for users has been updated for this feature
-Performance test has been passed

Level 3:

-Regression tests has been written and passed
-The code is covered by a minimum of 80% and all tests succeed, this does not include the UI.
-The feature is peer reviewed by at least 3 team members other than the author.
-QA testing done
-Feature is tested against acceptance criteria
-Feature works on Linux/Windows/Mac OS
-Code reviewed passed manual security tests

Level 4:

-The code is covered by a minimum of 90% and all tests succeed, this does not include the UI.
-The feature is peer reviewed by at least 4 team members other than the author.

Done Sprint:

Level 1:

-All sprint backlog items have been completed

-Diagrams for all sprint backlog items have been created

-All sprint backlog items have passed level 2 of done for features

-The code is covered by a minimum of 80% and all tests passed, excluding the UI

-All features have been peer reviewed by at least 2 team members other than the author.

-Verification demo has been created and has been accepted by the product owner

-Version number has been updated

-Documentation for updates has been written for this sprint

-Performance test has passed on Windows/Linux/Mac OS

Level 2:

-Code is deployed to test environment and passed the manual test

-All sprint backlog items have passed level 3 of done for features

Done Release:

-All product backlog items have been completed

-Product has been tested on Windows/Linux/Mac OS and works.

-The product owner has accepted the release

-The performance test on all platforms have been passed.

-Help documentation has been written for the product and is complete

-Documentation has been accepted by the product owner

-Security tests have passed

-Code is deployed to test environment and passed the manual test

Glossary

All