Product Planning
Octopeer for BitBucket

Out of Context

Date:
04-05-2016

Group members:
Arthur Guijt
4377338
a.guijt@student.tudelft.nl

Lars Stegman
4365801
l.s.stegman@student.tudelft.nl

Laurens Kroesen
4350286
l.kroesen@student.tudelft.nl

Cas Buijs
4345185
s.j.m.buijs@student.tudelft.nl

Thomas Overklift
4080890
t.a.r.overkliftvaupelklein@student.tudelft.nl

Responsible Teacher:
Alberto Bacchelli
a.bacchelli@tudelft.nl

Teaching Assistants:
Bastiaan Reijm
A.B.Reijm@student.tudelft.nl
Aaron Ang
A.W.Z.Ang@student.tudelft.nl

Introduction

Code is becoming more and more complex these days. In order to avoid bugs and low software quality, code reviews are needed. The currently most used tool - pull requests - have changed a lot over time, but there are still a lot of improvements possible. This 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 should track mouse movements on Bitbucket web pages.
- The extension should track mouse positions on Bitbucket web pages.
- The extension should track mouse clicks on Bitbucket web pages.
- The extension should track the number of commits per pull request.
- The extension should track the amount of lines of code added per commit per pull request.
- The extension should track the amount of lines of code removed per commit per pull request.
- The extension should store tracked data in a database.
- The extension should be able to display 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 should track the timestamp of when a pull request is created.
- The extension should track the timestamp of when a pull request is approved by a user.
- The extension should track the timestamp of when a pull request is declined.
- The extension should track the timestamp of when a pull request is merged.
- The extension should track the HTML elements the mouse hovers.
- The extension should have a settings page.
- The settings page should have an on / off button for mouse tracking.
- The settings page should have an on / off button for keystroke tracking.
- The settings page should have an on / off button for commit message tracking.
- The settings page should have an on / off button for meta data tracking.
- The settings page should support editing the whitelist.
- The settings page should have a way to point to a different octopeer server installation.

1.3. Could haves

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

1.4. Won’t haves

- The extension should support web browsers other than Chrome.

Roadmap

- Sprint 1 (Week 2)
    - Start of project / Setting up
- Sprint 2 (Week 3)
    - Product Vision and Product Planning
- Sprint 3 (Week 4)
    - Start testing
- Sprint 4 (Week 5)
    - Logged data to database
- Sprint 5 (Week 6)
    - Alpha
    - Start data processing and displaying
- Sprint 6 (Week 7)
    - Beta
    - Submit to SIG
- Sprint 7 (Week 8)
    - Process SIG feedback
- 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 whether my username is hashed or not.
As a user I want to choose whether my data is encrypted by the extension or not.
As a user I want to get a report of the data that is collected.
As a user I want to get a notification whenever I am not logged in on BitBucket and visit a pull request page for the first time in a browser session.

User stories of defects

As a user I want to get a notification whenever a crash of the tool occurs.
As a user I want to get a notification whenever there is no connection with the database possible.

User stories of know-how acquisition

As a user I want the extension to be intuitive.
As a user I want non trivial options within the extension to be explained.

Initial release plan

New features per release (milestone):

- Sprint 1 (Week 2)
  - Initial chrome extension without functionality
- Sprint 2 (Week 3)
  - Mouse movement logging to console
  - Established database connection
  - Initial popup menu
- Sprint 3 (Week 4)
  - Mouse movement logging to database
  - Improved popup menu
- Sprint 4 (Week 5)
  - Logging of screen elements to console
  - Local storage of user preferences
  - Logging of additional mouse events to database
- Sprint 5 (Week 6)
  - Logging of screen elements to database
- Sprint 6 (Week 7)
  - Basic display of collected data
- Sprint 7 (Week 8)
  - Basic display of analysed data
- Sprint 8 (Week 9)
  - Final product version

Definition of Done

Done Feature:

Level 1:

-All code is commented sufficiently
-All code meets the general coding standard
-All source code related to this feature has been committed to bitbucket
-The code is built 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, if applicable

Level 2:

-All requirements for Done Feature Level 1 have been met
-All code is fully commented
-The code is covered by a minimum of 70% and all tests succeed, if applicable
-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:

-All requirements for Done Feature Level 2 have been met
-Regression tests have been written and passed
-The code is covered by a minimum of 80% and all tests succeed, if applicable
-The feature is peer reviewed by at least 3 team members other than the author
-QA testing done
-Feature is tested against acceptance criteria and passed them
-Feature works on Linux/Windows/Mac OS
-Code reviewed passed manual security tests

Level 4:

-All requirements for Done Feature Level 3 have been met
-The code is covered by a minimum of 85% and all tests succeed, if applicable
-The feature is peer reviewed by all team members other than the author.

Done Sprint:

Level 1:

-All sprint backlog items have been completed with a minimum of Done Feature Level 1
-Diagrams for all sprint backlog items have been created
-The code is covered by a minimum of 80% and all tests passed, if applicable
-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/OS X

Level 2:

-All requirements for Done Sprint Level 1 have been met
-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
-All tests has passed
-Product is able to run on Windows/Linux/OS X
-The product owner has accepted the release
-The performance test has passed
-Help documentation has been written for the product and is complete
-Documentation has been accepted by the product owner
-Code is deployed to test environment and passed the manual test

Done Document:

-At least 2 team members have performed a grammar check
-The product owner has accepted the release
-The document meets the requirements set at the beginning
-The document meets the general reference style
-The document has been written in one clear writing style
-No multiple interpretations are possible in any part of the text

Glossary

- Bitbucket
    - Hosting provider for Git based code repositories.
- Git
    - A version control system for source code that models version data structure as directed acyclic graphs.
- GitHub
    - Hosting provider for Git based code repositories.
- Pull Request (PR)
    - A request from a developer to incorporate his code changes and /or additions into the existing code base. The new code is typically reviewed by other developers before being merged.