

Product vision

Draft version 0.1

Date: 28-4-2016

An introduction to 'Octopeer for Bitbucket'

The Octopeer plugin offers users an insight into their pull request review behaviour, but for whom is the Octopeer plugin useful? The Octopeer plugin tracks the behaviour of software engineers during their code review on pull requests and allows them to find patterns in the data. This is not only useful for the developers themselves, but allows software engineering researchers to gather data for their research. Besides the advantages for individual developers, the Octopeer extension allows software companies to have insight in their developers' talents.

When a developer wants to have his or her code added to an existing code base, he creates a pull request. This pull request might be reviewed by other developers on the code base. This review might be done in different degrees of thoroughness. If a reviewer is very thorough it is less likely that bugs or other unexpected behaviour will enter the code base, however if a reviewer reads through the request very quickly and doesn't test any of the code, the chances of something bad happening increase.

Using Octopeer for research

Software engineering researchers want to gain deeper insight into how developers build software. Understanding how developers work together and how they work, allows the researchers to develop new techniques, theories and teach future engineers the best practices. (Van Deursen, 2009)

However, to be able to research topics like pull request behaviour and other version control practices, researchers need data. The Octopeer chrome extension allows researchers to gather data on the behaviour of developers during their pull request reviews. The Octopeer extension tracks multiple factors like the amount of time that has passed between first looking at a pull request and approving or merging it.

[Explain why it is useful for developers]

[Explain why it is useful for software companies]

What Octopeer can do for you

The tool 'Octopeer for Bitbucket' aims to provide users with meaningful feedback regarding their review behaviour on pull requests in comparison to other developers. By providing this type of feedback, the tool should enable users to identify their shortcomings when peer reviewing code, and help them to act on these points of attention. Users will then in turn be able to see their progress, when they improve their code reviewing practices.

In order to provide Octopeer users with meaningful feedback the tool measures the browser activity of users when they are reviewing code. This includes, but is not limited to, the tracking of:

- *The time users spend on reviewing a PR in total.*
- *The time users spend on reviewing individual files in a PR.*
- *Which elements are in view, and how long.*
- *Which elements the user hovers with the mouse*
 - *Buttons*
 - *Code File*
 - *Comment box*
- *The amount of comments a user makes on a PR*
- *PR meta-data such as:*
 - *The amount of commits in a PR*
 - *The amounts of changed LOC per file in a PR*
 - *The amount of users that already approved a PR*

All of the data that Octopeer collects is stored in a database and can then be analyzed in order to provide users with feedback regarding their review behavior.

Octopeer can provide statistics that vary from straightforward data points, such as the time a user spent on reviewing a PR in comparison to their co-reviewers, to more sophisticated statistics such as the time users spend on reviewing PRs in relation to the amount of times that a PR has already been approved by other reviewers. If, for instance, it turns out that the effort reviewers spend on a code review is greatly reduced after a PR has already been approved for two or more times, it might be wise to drop a third code review as a requirement for a merge. Alternatively new methods can be adopted to ensure that a third review on the same PR is also carefully executed.

Octopeer can also do great work when analyzing additional web pages outside of Bitbucket. Users may leave the PR tab, but are they still considering the PR? Octopeer can partially answer this question by analyzing other tabs in the browser. When a reviewer is executing related search queries in Google, or when he is visiting Stack Overflow, it is likely the reviewer is still working on the PR. When tying these activities to certain files within the PR, we can identify difficult sections in the code, that required the reviewer to conduct additional research.

The focal user

While Octopeer can be used for quite a few different purposes, such as research or in large companies, the focal user of Octopeer will be an individual programmer. This means that a single programmer should be able to use the tool to improve his code review practices. To clarify how a programmer may want to use Octopeer we have created a persona:

“Bob is a 36 year old software engineer who works at a moderately sized software company. He and four of his colleagues work together in a team using an agile development model. Bob is pretty satisfied working in week long sprints as this is a balanced way of developing software. His company recently started using Bitbucket as code repository, and started using pull request instead of the code inspections they used to work with. Bob is glad they did that, because that means he doesn’t have to waste as much time sitting in an office with five others looking at deprecated code.

However, Bob is wondering if he is being effective at using the pull requests, and reviewing the code before merging. They used to look at code with everyone in one room before, and now he feels that he has more responsibility. That is why Bob wants to start using Octopeer. He wants to measure what he is doing when reviewing a pull request, and whether he is being effective or not. He hopes the tool can offer insights and tips as to which practices to change. If the tool can help Bob maybe he can help his team in turn!”

Difference with competing tools

Not many tools exist with the singular goal of collecting data about pull requests. Existing tools collect data about an entire code repository or collect very little data about the pull requests at all. What is unique about the tool `Octopeer`, is that it focusses mainly on pull requests and related data.

Upsource is an example of an existing polyglot code review tool which assists people with performing their code review. Besides assisting with the code review process it also offers statistics analysis.

Upsource is concerned with the following data:

- Amount of comments
- Amount of reviews

Octopeer goes further. It collects much more data related to pull requests, which includes:

- Mouse movements, clicks and hovers during the pull request
- Keystrokes during the pull request
- Comments regarding the pull request
- Who reviewed the pull request
- Time taken for reviewing the pull request
- Amount of commits in the pull request

- Additional sources which were used during the pull request

Octopeer targets two types of users, researchers and companies.

As researchers want to store as much data as possible and companies prefer keeping their information hidden, Octopeer allows users to change their privacy settings for themselves. The user can indicate what kind of information should be tracked and whether it should be anonymized or not and the settings can be changed during run-time.

As most existing tools have been developed for use with GitHub, Octopeer has a different focus and is developed for BitBucket which makes it a unique tool for collecting data about the code review process.

Target timeframe and budget

'Octopeer for Bitbucket' is planned to be designed and implemented by a team of five software engineers in a timespan of ten weeks. The development is part of the 'Context project' where each developer works on for approximately 28 hours per week. There are also other activities involved in this project so each developer has roughly 20 hours a week that can be invested in the development of the tool. This adds up to a total of $20 * 10 * 5 = 1000$ hours that can be invested in the development over a period of ten weeks.

The costs for this tool are nonexistent because all of the work is being done by students. In a business setting however each developer would receive an hourly rate somewhere in between €30,- to €45,-. This adds up to a total estimated market value from €30.000,- to €45.000,- for the tool.

Reference List

Van Deursen, A. (2009). *The Software Engineering Group*. Retrieved from <http://swerl.tudelft.nl/bin/view/Main/SoftwareEngineeringResearchGroup> . Accessed: April 27, 2016.

Possible sources

<https://www.scopus.com/record/display.uri?eid=2-s2.0-84938791961&origin=resultslist&sort=plf-f&src=s&st1=%22pull+request%22+AND+%28behaviour+or+uses%29&st2=&sid=76BD285F66F4FE04C9BF5925ABD3B637.CnvicAmOODVwpVrjSeqQ%3a10&sot=b&sdt=b&sl=53&s=TITLE-ABS-KEY%28%22pull+request%22+AND+%28behaviour+or+uses%29%29&relpos=9&citeCnt=3&searchTerm=>

<https://www.scopus.com/record/display.uri?eid=2-s2.0-84944104897&origin=resultslist&sort=plf-f&src=s&st1=%22pull+request%22+AND+%28behaviour+or+uses%29&st2=&sid=76BD285F66F4FE04C9BF5925ABD3B637.CnvicAmOODVwpVrjSeqQ%3a10&sot=b&sdt=b&sl=53&s=TITLE-ABS-KEY%28%22pull+request%22+AND+%28behaviour+or+uses%29%29&relpos=3&citeCnt=0&searchTerm=#activation>