Visualisations

List of visualisations

The following visualisations are designed:

- Amount of time spent on a pull-request
- Amount of time spent on a pull-request compared with the amount of changes
- Number of comments on a pull-request
- Number of comments on a pull-request compared with the amount of changes
- Number of comments on a pull-request compared with the lifetime of a pull-request
- The average length of comments on pull requests
- Average comment size

Amount of time spent on a pull-request

When peer reviewers can see how much time they spent on peer reviews. The data is displayed in a graph. On the x-axis are the (id numbers of) pull-requests placed. On the y-axis is the durations displayed.
Peer reviewers can make some predictions about how much time they are going to spend on future peer reviews, making it easier to plan peer reviews. This is a benefit for both peer reviewers and managers. While making a schedule, it can be ensured that there will be enough time for peer reviewing.The peer reviewer knows he has enough time to do his job and the manager knows that the quality of the code will be maintained.
Another benefit is that the peer reviewer can compare his times with other peer reviewers. When there is a big difference, these two peer reviewers can investigate on each other’s methods and choose the best method to improve quality.

Amount of time spent on a pull-request compared with the amount of changes

Pull-requests differs on a couple of points, like amount of changes and the level of (new) code complexity. The data is displayed in a graph with an x-axis, left y-axis and right y-axis. On the x-axis are placed the (id numbers of) pull-requests. On the left y-axis the duration is displayed and on the right y-axis the amount of changes in a pull request. For each pull-request a vertical line displays the time that a peer reviewer spent on peer reviewing. A curve shows the amount of changes in each pull-request.
A high amount of changes requires more time than a low amount of changes to review. This gives better insights in how much time is needed to review a certain amount of changes.

Number of comments on a pull-request

The data is displayed in a graph with on the x-axis the (id numbers of) pull-requests and on the y-axis the number of comments. A vertical line for each pull-request displays the number of comments for that particular pull-request.
When a reviewer places a lot of comments on a pull-request, then the peer reviewer has probably analysed the changes in depth. It says something about the quality of the peer review. Generally, a higher amount of comments describes a better quality of peer reviewing.

Number of comments on a pull-request compared to the amount of changes

The data is displayed in a graph with on the x-axis the (id numbers of) pull-requests and on the left y-axis the number of comments. On the right y-axis is the amount of changes displayed. A vertical line for each pull-request displays the number of comments for that particular pull-request. A curve shows the amount of changes of each pull-request.
Using the number of comments alone says something, but more accurate data can be derived. For example: when a pull-request contains about 700 changed rows and a peer reviewer makes five comments and another pull-request contains about 50 changed rows with 5 comments then it is clear that the latter one is reviewed with a higher precision. So, it says something about the accuracy of the review. It could be the case that a piece of code is so good that the peer reviewer has nothing to comment, but with a higher amount of changes the chances on faults are bigger.

Number of comments on a pull-request compared to the lifetime of a pull-request

The data is displayed in a graph with on the x-axis the (id numbers of) pull-requests and on the left y-axis the lifetime. On the right y-axis is the number of comments displayed. A vertical line for each pull-request displays the number of comments for that particular pull-request. A curve shows the lifetime of all pull-requests.
When a lot of comments are made on a pull-request then there is some time needed to solve issues. A pull-request with a huge amount of comments and a short lifetime could indicate that the comments are ignored. So it says something about the way comments are handled.

The average length of comments on pull requests

The data is displayed in a graph with on the x-axis the (id numbers of) pull-requests and on the y-axis the average length of comments. A vertical line for each pull-request displays the average length of comments for that particular pull-request.

By counting the number of characters in a comment, the value of the comment can be estimated. A comment consisting of a couple of words has probably less value than a comment with multiple sentences.

Average comment size

The data is displayed in a graph with on the x-axis the (id numbers of) pull-requests and on the y-axis the average comment size. One curve displays the average comment size of the peer reviewer, another curve displays the average comment size of all peer reviewers.
This graph compares the average comment size of the peer reviewer with the average comment size of all peer reviewers. This gives some insight in the relative quality of the peer reviewer compared to all peer reviewers.