Research Hint Generation

This document contains the results of the Hint Generation research.
During the project we came up with the idea that hint generation, based on the data that is logged, might be an interesting and a useful feature for Octopeer, as it is a way to actually give users feedback on their peer review behaviour.
The research has been divided in the following sections: Suggestions, Feasibility and a recommendation.

Suggestions

- Add an extra page, which can be reached via the pop-up, on which hints are listed.
  The way in which this page visualizes these hints can differ:
  - All possible hints are shown and the applicable once are distinguished (by either separating them or adding marks to them).
  - Only all applicable hints are shown.

  + It is clear for the user where they have to be for feedback.
  + The feedback part is separated from the rest of the extension.
  − Users aren't directly notified about hints, but should go to the extra page.

- Create pop-ups (that will appear in the browser) for every hint when they become applicable, based on the data.
  + The user will get directly notified by new interesting hints
  ± At some point the hints may get annoying, as the user just wants to review a pull request, but they can disable the hints via the options.
  − The user is not provided by an overview of applicable hints. Only the new ones will occur. So if the user forgot about a hint, it might pop-up later on, but they can't look it up.

- Process the most important (a top-3 for example) hints into the pop-up of the extension. New hints may be communicated to the user via a change in the icon of the extension (for instance add a small one in the corner of the image, indicating that there is one new hint).
  + Users will have an overview of the most applicable hints.
  + Users will also be notified about new hints.
  − Some kind of weight system has to be designed for determining which hints are important enough to add to the pop-up of the extension.

Feasibility

Because at the moment of the research there is only one week left for implementing the feature, we decided to add a section to this research on the feasibility of this feature as well. In relation to feasibility, the following things have to be taken into account:

- Rules have to be designed from which hints can be deduced. These rules have to be created 'from scratch' basically, as our project didn't involve any visualisation (yet) of the data that is gathered. In order to create a sufficient set of hints, quite some time has to be invested on creating rules. This is important as if the set of rules is not sufficient in size or quality, then

the hints that are deduced won't be very useful. If that's the case, then the feature is basically a waste of time.

- In relation to the previous point, hint generation is actually more applicable to the data visualisation group, as they have spent time on creating (visual) overviews of the data and thus it will be easier to design rules based on those overviews.
- Based on the suggestions from the *Suggestions* section, a new interaction design research has to be performed, as there are a lot of ways to interact with hints. This research has to be documented and processed, as it will be included to the final report.
- As our architecture is based on sending data to a database and not retrieving data, changes in the architecture are required for implementation of the features. These changes have to be thoroughly thought through before implementing them.
- As already noted, after the research week there is only one week left for implementing this feature.
- The implementation of the hint generation has to be stable and thoroughly tested.

Recommendation: The feasibility section simply points out too many points that make implementing the feature too risky during the last sprint. Besides, another feature will be implemented for visualisation, which will be extendable and has relatively a lower level of risk. Therefore we will not implement this feature in the last sprint.