Pull Request Standardisation

This is intended to act as a guide/reference for how we should conduct pull request reviews at SmartStream.

Pull request are a powerful tool within software development, it's widely accepted that an author is incapable of efficiently reviewing their own work and thus requires a 3rd party to help. This gives the author and observers a chance to learn from previous mistakes and generally improve their work. However, with lack of standardisation throughout the company, each person has a different definition of what a "good" pull request looks like, and therefore different requirements to get it into a merge-able state. This can lead to large amounts of time wasted during this stage of development. I am hoping to drive a discussion into how pull requests are handled within SmartStream.

Issues

The main areas I've identified as problematic:

- Waiting for reviews.
- Comments which are not necessary for the code to be merged holding the pull request - not enough use of the "Needs work" feature
- Comment "tennis" e.g. waiting for a response to a query on someone’s comment before implementing
- Different opinions on code style/implementation which reaches the same result holding the pull request.

Comments are valuable to both the reviewer and author, but they should not be where the discussion starts and end. The comment section on a pull request should be a reference point where conversations can start from, it's useful for identifying an area for change but past this, the value decreases rapidly.

Suggestions

These issues can be easily solved mostly through communication and collaboration, getting a pull request merged should be everyone primary goal and focus. Efficiency in acheiving this goal can be improved if everyone is working from the same page, i.e. a shared idea of what effective review looks like. Here are some of my suggestions:

- Comments on a pull request marked with "Needs work" are requirements that need to be fulfilled before approval.
- Comments on a pull request without a "Needs work" mark are suggestions on the code that don't need to be completed before approval.
- Discussions away from stash (e.g. Skype calls) will have more value than comment discussions.
- Code written is always more valuable than code not yet written and therefore should take priority.
- Two reviews minimum before merging.
- An incorrect comment has more value than no comment.
- Code must be formatted using the Google Style Guide - IntelliJ/Eclipse variations in Google Style should not be the reason for holding the pull request.
- Multiple smaller pull requests are more manageable and produce better results than one large one

Priority

Code reviews must become a priority, code that is currently being written is always of lower priority. They are an opportunity to learn, help others learn and (what we're here to do) merge code!

Communication

Nobody enjoys revisiting their code to find large numbers of critical comments - it can feel quite impersonal and frustrating. However, a quick call between reviewer(s) and author can alleviate most of these feelings. An informal discussion on general and more specific issues can lead to wider discussion, and not only correct that instance of the problem, but help prevent future occurrences of the same issue. Helping the author understand the benefits of the recommended changes to the code decreases the likelihood of the error being repeated, therefore reducing future workload for both parties. This communication is much more effective than a string of back and forth comments.

Size of pull requests

Large pull requests with many comments can become impractical to properly manage and keep track of. It may seem more work to get team members to look at 3 different pull requests, but this method particularly when done incrementally throughout development can help reduce the risk of repeating an error throughout the whole task and makes the pull request more readable for reviews.

Confidence

If you have a suggestion, but you aren’t 100% confident that it is correct, there is still value in leaving a comment. Even if it turns out to be wrong, there’s value in being informed that your suggestion is incorrect and can lead to a wider conversation about why so, supporting team and individual learning even if there was no change to the pull request. Your question/comment may be something that someone else was also thinking but didn't want to ask. It's okay to be wrong.

WIP Pull Request

This is a method of pull requesting which is done as soon as work starts and is updated as development continues. It allows early feedback and cooperation on something that may not be viewed until days later, reducing the amount of comments and allowing a steady stream of feedback into your work. This is one method to improve the process, to read more view the link below.

https://ben.straub.cc/2015/04/02/wip-pull-request/

Please feel free to add any comments or suggestions on this document, it's meant to be a collective agreement.