SlugTrade Testing Process

We decided at the start of the project to integrate features by creating pull requests on GitHub. All of us had previous experience with integrating features directly to a develop branch (or master!) and wanted to avoid the risks inherent to doing integration that way. As the person who was most motivated to enforce the pull request system, I offered to handle pull requests as they came in. I figured that as long as there were no more than an average of one pull request per day, integration of new features would stay fast. In actuality, we had an average of one and a half pull requests per day, which is too much for one person; Austin often ended up picking up the slack. Also, some pull requests had so many changes and came in so late in the sprint that it was difficult to test them at a sustainable pace.

Over the course of the project, we came to consensus about how pull requests should be handled, which we codified in our Team Working Agreements. When a feature is ready to be reviewed, the developer creates a pull request on GitHub. In the description of the pull request, they describe the feature in terms of what a user would see. They also include a test plan (and, ideally, the commit would contain a automated unit tests for the new feature) so that the reviewer can verify that the changes are good. The developer then lets the group know that a pull request is up. The reviewer checks out the feature branch, locally merges the develop branch into it, and tests according to the pull request description. The reviewer also visually inspects the code by doing a diff of the develop branch and the feature branch to make sure that the commit contains only the changes described, and that the new code is reasonably clean. The reviewer describes any issues by commenting on the pull request, and the issues described in the comments must be addressed (and possibly fixed) by the pull requester before the feature is merged. When all issues have been addressed, the reviewer merges the pull request via the Big Green Button on GitHub.

We came up with the process by trial and error over the course of the term, and writing it down in our working agreements allowed us to make some substantial improvements. The most useful of these was including a severity estimation of the issue described; this made it more clear to the pull requester how close the feature was to being merged and what the quality standards for features were. The system is definitely still a work in progress, and with two or three more sprints I think we would have a very good integration pipeline in place. With another few sprints (and the time to do it--handling pull requests takes up almost all of my time in this project), we would probably be able to start automating the pipeline.

Example Pull Request

Navbar button convert (developer: Austin)

This pull request was a refactor of the buttons on the navigation bar. Before the refactor, only the text of the buttons were clickable, which frequently resulted in user error when navigating the site. After the refactor, the entire button was clickable, which improved the UX of the navigation bar.

Due to an implementation quirk of the navigation bar, there are two HTML templates for the navigation bar--one for the landing page and one for all of the following pages: products.html, not_authenticated.html, edit_profile.html, profile.html, add_closet_item.html, transaction.html, my_offers.html, item_details.html. The equivalence classes are:

- “Products” (landing page)
- “Login” (landing page)
- “Sign Up” (landing page)
- “Profile” (landing page)
- “Offers” (landing page)
- “Logout” (landing page)
- “How It Works” (landing page)
- “Products” (other pages)
- “Login” (other pages)
- “Sign Up” (other pages)
- “Profile” (other pages)
- “Offers” (other pages)
- “Logout” (other pages)

To test these equivalence classes, I clicked within each button outside of the text and verified that the button still linked to its expected page. I also clicked within the text to verify that clicking the text still behaved as expected. I did this on the landing page for the landing page navigation bar, and on the products page for the other navigation bar.