Tackling the QA work

1. Create a test plan – I always like to start work with a QA test plan. The first thing to know is the extent of the API in question, and which parts of it should be tested all the time, and which parts we only want to test from time to time. This will make it easier to decide which cases should be automated, and which should be done manually – or which ones will require some different user input.
2. Custom users, and Increase knowledge of GitHub API – It would be easier to test if were to increase my knowledge of the GitHub API, then make several custom users with different settings so that it would be easier for me to find and test the different data with different users. It is difficult and time consuming to shift through individual users looking for the correct data that I want to test.

Detailed Overview of manual/automated by section

* 1. Activity – These are mostly user feeds, events, notifications and the like. These are all pretty quick and straight forward, I would think the majority of these could be automated, but I would have to go through them in detail to figure out which ones shouldn’t/couldn’t be automated.
  2. Gists – Most of these tests can be automated, but if the Gists get too large, or there are too many of them, that might cause some problems with automation. Another case of needing to dig into it more.
  3. Git Data – One of the more complicated areas of the API. Like the others I expect there are many tests here that you could automate, but I’d need more time to figure out which ones. Most of the updating tags, blobs and the like can be automated though.
  4. Integrations – This would only need to be checked when the API is changed. I don’t think this would need to be automated.
  5. Issues – This is a large section, but most of it is pretty straight forward, it would take a while to set up, but I think the vast majority of these tests can be automated, even the custom media tests.
  6. Migration – This is another area that doesn’t seem like it would need to be checked very often, these could be mostly manual tests.
  7. Miscellaneous – Most of these could be automated, I would need to check through the license and markdown sections, but as long as the files are not too large, then these could all be automated.
  8. Organizations – Besides the OAuth scope testing (which I still need to learn) all of this can be automated.
  9. Projects – Despite the name this is mostly Gets to find project resources, most if not all of this can be automated.
  10. Pull Request – I’d want these tests to be manual until I learned more about Pull Requests – how long they take, the size etc
  11. Reactions – These can almost all be automated.
  12. Repositories – The biggest section of the API. There are a lot of things going on in this section, and a lot of them require moving files around. Some of this could be automated, but the big heavy lifting, and important sections would probably have to be manually tested to ensure they are getting the proper look and time.
  13. Search – A fairly small and straight forward section, these can pretty much all be automated.
  14. Users – I didn’t go into this section as much as the others, but it looks like emails, followers, and keys. Most of this could probably be automated.
  15. Enterprise – This looks fairly complicated and requires extra authorization, these probably have to be done manuall.

1. Manual – Automation – For now I think anything that doesn’t stress the system – comments, issues, feeds, reactions and the like can almost all be automated. For the more involved actions – moving files, downloads, merging then those should be done manually, until a way to automate them that doesn’t stress the system can be found.