Testing

Quality Assurance: Jamie Alexander (jalexa17@u.rochester.edu)

Our development technique for this project was based on test driven development, through the implementation of a three stage processes that cycled through feedback and design modification. The goals of each stage were:

- First Stage: Individual Unit Tests
  - To supply module writers with the ability to develop using correct test driven development techniques (standardized unit tests and reviewed testing spaces), and to give QA the ability to check on progress and module synchronization based on test performance
- Second Stage: Integration Tests: Communication
  - To make sure that each module was communicating correctly with other modules through dependency based integration tests
- Third Stage: Integration Testing: Live Functionality
  - To test for each functionality within project through manual hands on testing. Originally, tests were being written to try and automate this process, but as the complexity of the project become clearer, it was decided that the most efficient way to finish the testing was for the team to meet and test functions together initially, and then have QA, the Project Lead, Tech Lead, and module writers who were availed continue to test functions/ make changes to module code as final required changes were discovered.

First Stage: Individual Unit Tests (Oct.15th- Nov 7th)

- Test functions were written to pair every function in each module according to original design specs.
- Functions relying on direct dependency of other modules were to be tested with hard-coded data to simulate a live call
- Feedback was given for initial module unit test functions (Oct. 18th)
  - Assertions were reviewed for each test function to ensure appropriate bounds were met
  - Inter-module dependencies were checked to be synchronized based on expected parameters and expected return types of approved tests
- Unit tests for each module were edited based on feedback (Oct. 22nd)
  - Persisting problems were edited by QA
- Meeting between Project Manager, Tech Lead, and QA to discuss possible issues with current design based on Unit Test expectations (inter-module dependencies through calls and parameter passing, Oct. 23rd)
- Full team meeting was done to clarify design specifications (Oct. 28th)
  - Unit Tests were modified to fit new design specifications (Nov. 1st)
- More complex function issues were found through test driven development
  - Issues found by module writers were reported to Project Lead, Tech Lead, and QA
  - Issues found by QA were reported to module writers if error occurred from discrepancies in module code and design document, and to Project Lead

    - and Tech Lead if the issue persisted with even with code aligning with design document
    - Meetings (in person or over messaging systems) were held between Project Lead, Tech Lead, and QA to make necessary revisions according to development test feedback. This was repeated through the week process of stage 2
- Number of tests passing for each module for Nov. 7th benchmark

Module	Test Functions Passed	Assertions Passed
User Interface	12/12	36/36
Push-Pull	6/9	29/40
Workspace	3/5	10/17
Revlog	3/8	6/19
Repos	3/6	8/16

Second Stage: Integration Tests: Communication (Nov 8th- Nov 19th)

- Each dependency for each module was tested by calling every method that could be called from outside of any given module
- Test parameters were sent from dependent modules in cases were the function call required parameters
- Return values for each function call that requested data were fed into dependent modules to check for type-data error free flow between modules
- As any final updates were made to the design (due to running into issues while modules fixed the last of their errors/ unit tests failures), the intergradation tests were changed if any dependency was changed or added
- Communication integration testing was finished by Nov 19th

Third Stage: Integration Testing: Live Functionality (Nov 20th- Nov 25th)

- Full team meeting to actually begin thoroughly testing live functionality for each module, and the project as a whole (including every command expected to be available for users of the system, Nov 20th)
- The entire process of creating a new repository, creating new branches, checking out branches, committing code, pushing code, and pulling code was tested by attempts to use the system, with debugging as a group/ issue ticketing when things did not work.
- Team rewrote and debugged code (especially the Project Lead and Tech Lead. This is where the majority of their code writing happened)
- Second full team meeting to continue live testing was done on Nov 23rd (in class and then again that evening)
- Merge testing was never completed in live development due to lack of time remaining after finishing required rewriting and debugging of certain key features within project (Nov 25th)