

int team = 4;

11/14/2013

EXPERIENCE ON CREATING 25 TEST CASE:

Most of the work between the last deliverable and this one has been polishing. The specifications required 5 test cases for deliverable #3 and 25 test cases for deliverable #4. We already had more than 25 test cases by the last deliverable. We have updated our previous test case to have some specific requirements. Evan has made some modifications to the script and has been working on a search method that will make the framework more robust by removing the need for line numbers in the test cases. He also added the ability to have a custom output in case the return is a struct that needs to be checked in a manner that "cout" would not work well with.

We encountered a strange problem while trying to test another function. In "snprintf.c" they used the old K&R style function declarations where the parameter types come after the parameter list. None of us had seen that style before so we were a bit confused. Our script wouldn't compile the object files for these test cases as well. After some playing around we figured out that the g++ compiler doesn't support this type of declaration. We changed the output and compiled it with gcc and it worked fine. The problem that this introduces is that we assumed everything that would compile with gcc would compile with g++ and we built our framework accordingly. In the end we decided to not support K&R style declaration as we would have to add a flag in the test case file and add more complexity to the script. Being that we are not supporting it we had to throw that function out.

At this point we have noticed that our framework is kind of limited. We found a very simple function at first to test and designed our framework around it. The way we have designed our framework may be too simple currently. We chose the easy story and assumed that all other stories would be of the same complexity which has made us make the compromises that we have had to. Creating test cases that are testing complex functions proves to be difficult. We may need to redesign a bit to allow those functions to be tested. Operating in an agile style has gotten things done, but using the waterfall development method could allow for a little more of the long term thinking.

We are currently testing three functions. We plan to test five. Robert has found a possible fourth at the time of this being written and will be determining test cases for this function.