1. **Project plan for remaining semester**

There are still 2 weeks remaining in this semester. For the first one, we will do regression test for 2015 from 2011 test cases. For another one, we will do new feature test from 2014. Also, we will deal with problems that may be found in the first week.

For detail, there is also Team Project Plan in attachment.

1. **Tasks accomplished this past week**

Last week, we read the code in baseline 2015 and made [a header relationship graph](https://docs.google.com/drawings/d/1xGxFXYaHzRrPw_GAN_Wlg3QwbAWU_GtDu25LrNYM5t8/edit?usp=sharing) for it(easy for taking clear about the whole project). We merged the Objective C counter .h and .cpp to the 2015 baseline and do several basic test. We found there were some error in Cyclomatic Complexity count in the test and revised the code to pass the test. Also, we delete the functions in Objective C which were copied from 2011 baseline and overrided the functions from its superclass (functions in superclass have new features).

1. **Tasks scheduled for this next week**

Our schedule for the next week is to finish regression test from 2011 and fix the problems that may generate from the test. (test case number from 5 to 25).

1. **What your team plans to complete by next week**

We plan to complete the regression test part by next week and prepare to do some of new feature test according to time we have.

1. **Any questions/concerns you have**

In this week, we found two bugs from both 2014 and 2015 baseline.

The first one: the baseline can only count “#define” and “# define”. If the user use more space between “#” and “define”, the counter cannot count it.

The second one: the counter counts “int main(int a)” and “int main( int a)” with different results. The counter cannot count keywords close to other characters.

However, both bugs do not influence on the SLOC and cyclomatic complexity.

We found that 2010 Objective-C code is exactly the same as C\_CPP counter and 2011 Objective-C code is just added functions from the superclass of C\_CPP, Java and C sharp counters. In other words, the Objective-C counter we have is totally the same as C\_CPP counter now. Although, Objective-C is a super set of C and all C files is acceptable in Objective-C files. I think the Objective-C counter should also count keywords from itself like “BOOL”, “NSString” and etc.