CS-AD 209 - Software Engineering Professor Michael Paik
New York University Abu Dhabi 3 April, 2014

Assignment 3

Due Date: This assignment is due at 23:59:59 on 10 April, 2014.

Submission: A report detailing the work done for this assignment should be emailed to `mpaik@cs.nyu.edu`.

Indications: You will work in your project groups.

PLEASE read the entire document before beginning.

All teams now have approved projects, and are hopefully gathering more requirements and starting to code (if not, things are going to get ugly later...). Between the previous assignment and this one we have covered some best practices. For this assignment you will have the following tasks:

1. [**10 points**] Set up source code management for your project. Your first instinct may be Github, but please read the rest of the tasks before you make a determination. Setting up source code management means creating a project, a folder/module layout, users, etc. **not** just the bare creation of a repository.

2. [**20 points**] Set up and specify a way in which an observer (i.e. me) can easily see commit frequency and volume per user, e.g. contributor graphs on Github or any of a number of `git` or `svn` scripts available online that tabulate statistics. If you're using a script to pull statistics, create a `cron` job to email me the per-user stats each week.

3. [**20 points**] Set up and specify a code review process. You get code review for free on Git, but it is somewhat limited. If you are using your own git repository, Gerrit is the tool of choice; for Subversion (and also Git) Review Board and Rietveld are good choices (Rietveld is the trunk off which Gerrit was forked, but supports Subversion). Trac offers one or two code review plugins which may or may not work with Bloodhound.

4. [**20 points**] Set up bug tracking, ideally with integration into your repository. Bloodhound, Trac, and Bugzilla are good open-source choices; you get some hobbled bug tracking with Github, but it is not recommended.

5. [**30 points**] Set up continuous integration. Depending on your chosen language, this may include periodic automatic compilation, but at a minimum should include execution of the tests you specified in the previous Assignment, and any tests that are added as time goes by, along with a generated report of which tests succeed and which fail.

6. **Extra Credit [10 points each; 50 total]** You can get many of these services online, sometimes for free for small projects. However, every developer should learn to set up such an environment on his or her own at least once. As such, for each of the previous 5 tasks you perform on a server you control rather than a pre-built online alternative, a 10 point bonus is available.

   You may, if you desire, opt for the extra credit and subsequently change to online alternatives if you want the experience of setting up an environment but don't want to maintain the system under potential issues that arise, but you will have to sufficiently demonstrate that your environment is working first. Keep in mind that using shell scripts to pull/push to/from e.g. Github or Bitbucket will allow you to maintain your source code repositories in parallel.