CSE 403, Winter 2007
Homework #4: Zero Feature Release ("ZFR"); 20 points

Due: Friday, February 16, 2007
(in class and/or by 11:59pm electronically)

Assignment Description:

The third milestone you will submit for this project is a "zero-feature release" or "ZFR" for short. Your three SteppCo customers, Marty Stepp (stepp), Saleema Amershi (samershi), and Brian Harris (brharris) would like to see the major subsystems of your project work. The purpose of this milestone is to work out bugs in the various processes such as version control, bug tracking, and deployment. Unless otherwise specified, no actual system functionality needs to be present or working in this milestone, though you may include functionality if you like (it will not be graded).

Items Required:

Your group should submit a ZFR document describing several items below. You should also put the necessary electronic resources in place so that the graders can examine and test the processes described in the ZFR document. The document should address the following four (4) items:

A. Installation/Deployment Process

Your project's installation/deployment process is the set of steps that must be taken to download / install (if necessary) and run the system. Turn in a set of instructions the user should follow from start to finish, to get your system to appear on his/her screen. In grading, we will perform these steps to test your deployment. After the user performs these steps, the system should appear with some sort of initial screen, but beyond that, no actual functionality needs to be present in the system.

The directions should be written in sufficient detail so that a normal user (who understands the operating system and internet, and knows how to download and install software, but is *not* a developer) can follow them. Assume that the user does NOT understand command-line interfaces such as SSH or a DOS/Unix terminal. If there are any common problems the user may face, please describe them briefly and how to deal with them if possible.

If your system is web-based, perhaps the process is simply for the user to type the site's URL. If your program is a client application, perhaps the process requires the user to download an executable file and run or install it. If using your system requires any prerequisites of the user's computer, such as certain libraries to be installed or for the user to connect with a certain browser or operating system, please note this in your document.

Part of your grade will be based on how simple this process is for the user to perform, and how accurately your directions match what the user must actually do.

B. Documentation

Your project's documentation is the set of files and documents that the user can read to understand the system. Turn in a set of instructions the user can follow to find your documentation. In grading, we will follow these steps to retrieve and view the documents. After performing these steps, a "front page" document should be visible, but beyond that, the documentation does not need to be complete or contain any particular information.

The directions should be written in sufficient detail so that a normal user (as described above) can follow them.

C. Build Process

Your project's build process is the set of tools and commands necessary to compile and otherwise "build" your system's code. Turn in a set of directions to your customers, describing how they would find your build system, check out the files from it, and compile/build them. In grading, we will check out and build your code.

The directions should be written in sufficient detail that an intelligent developer can follow them.

For full credit, your team should have some sort of reasonable resolution to the issue of version control. For example, you could handle this issue by using a CVS or Subversion repository in a commonly reachable location, or by hosting the code on a public system such as Sourceforge. Your instructions to the customer should provide instructions for how to access any such system or repository.

If the system(s) involved in your build process have security controls or require login information to access them, you should provide this to the customers as part of your turnin. For example, if you are storing your code on a server, you should create user accounts for the customers if necessary, or if it is hosted on a shared university server where the customers already have accounts, you should set your system such that the customers' accounts can access it.

Part of your grade for this item relates to the number and complexity of commands the developer must use to achieve his/her goals. Ideally your system will have a single command that does a "one-step build," that checks out all source code from your repository, compiles and builds all necessary binaries, packages them, and places them in a known location ready for deployment. (Technical issues may prevent such a one-step process for some projects.)

D. Bug-Tracking System

Your project's bug-tracking system is the set of tools used to document existing bugs and missing features in your system's code. Turn in a set of directions to your customers, describing briefly how they find your bug-tracking system, examine the list of current bugs, and file a bug. In grading, we will follow these directions and examine whether the bug tracking system exists and is usable.

The directions should be written in sufficient detail that an intelligent developer / tester can follow them.

Many bug-tracking systems allow bugs to be assigned to particular users or teams. Your bug tracking system does not need to contain a comprehensive list of bugs or missing features. But for full credit, it should have at least one bug filed for each active developer or relevant team member. These bugs may be requests for features that are incomplete, such as "TODO: implement login behavior." If possible, the bugs should list relevant priority and approximate phase that they will be fixed (beta or final version).

Submission and Grading:

If you choose to turn in your ZFR documents electronically, please submit them in Word (.doc) or PDF format; otherwise turn them in as printed pages in lecture. Each document's file name should begin with your project's name and reflect what it contains. For example, if your project is called "SuperAwesome", acceptable file names might be `SuperAwesome_ZFR.pdf` or `SuperAwesome_bug_tracking.doc`. You may receive a deduction if you turn in clumsily named or organized files. Place your project's name and all group members' names atop each document. Only one copy of the documents should be submitted for each group.

Any electronic resources connected or referenced by these documents should be available at the moment the documents are turned in, so that grading can occur promptly after your submission. You may receive a deduction if the grader is unable to reach resources named in your documents. If necessary or desired, you may wish to speak to the customer(s) to arrange a "dry run" to make sure they can reach your resources successfully.