

## **CSE 403 – Winter 2011 Assignment 4 – Alpha Release**

**Due date: Mon Feb 7 by 11pm, via the Turn-In link on the class webpage**

Your customer wants to make sure you are off to a running start with their product development. This assignment proves to them that you have the development infrastructure in place for a successful delivery.

### **Deliverables**

The alpha release (also called zero feature release here in CSE) consists of a skeletal implementation of your product. This initial implementation must be able to be installed and run, but running it does not have to do anything except display an initial client (user interface) and show it is integrated with the backend server (database). Essentially, you'll want to show a successful round trip operation through your client-server architecture. For the assignment we also ask for you to show us a set of your good software development practices.

#### **Part 1 - Featureless Live Product Website**

Provide us with a URL to reach a front page for your product.

If your project is a web app, this URL would be the front page of your actual web app. The front page does not have to have any features working except something to demonstrate a round trip operation.

If your project is a mobile app, this URL would be the address of a web page where we can download the app to install or run it (on the simulator or on a real device). We may need to have one of the department phones returned for this exercise.

Part of completing your alpha release is creating your database. Therefore, your alpha release must in some way connect to this source of data, fetch at least one piece of data from it, and display the data in some way on the screen or web page.

#### **Part 2 – Usage and Development Material**

Provide us with documents describing the implementation practices of your team. You may consider the Joel Test and his advice as you put processes in place for your developers. Part of your grade for this assignment will be based on the simplicity of these processes, and how accurately your directions match what the user must actually do.

Please put the material in a location that the staff can access (ensure permissions are set correctly) to test the processes described in the document promptly after your submission. You may want to arrange a "dry run" with your customer group beforehand.

## 1. Source control and build process instructions

- a. Your project's build process is the set of tools and commands to compile and "build" your system. Turn in a set of directions to find your build system, check out its files, and build them. In grading, we will follow these directions.
- b. For full credit, your team should have a reasonable solution to the issue of version control. For example, you could handle this issue by using SVN repository in a reachable location, or by hosting the code on a public system such as Google-Code. Your instructions should explain how to access any such system or repository.
- c. The directions should be written in sufficient detail that a typical developer can follow them. If the system(s) require login information to access them, you should provide this information.
- d. Part of your grade for this item relates to the number and complexity of commands the developer must use. Ideally your system will have a single command that does a "one-step build," that checks out all source code from your repository, builds all necessary binaries, packages them, and places them in a known location.

## 2. Bug tracking system instructions

- a. Describe the set of tools used to document existing bugs and missing features in your system's code. Turn in a set of directions describing briefly how we can 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 so that an intelligent developer can follow them. Please also turn these directions to your customer group, so that they may also use them to track and contribute to your project.
- b. 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. If possible, the bugs should list priority, status and a timeline to fix them.

## 3. Database access instructions

Since your product must have a server-side data component, your alpha release should contain a set of instructions about where this data is stored and how to access it. Turn in a set of directions that tells the staff how to find your data, and how to briefly perform a trivial access of this data. For example, if your data is in a database, inform us how to connect to this data and perform one very simple query against it. In grading, we will follow these directions.

## **Submission and Grading**

Part of your grade on this assignment comes from your web URL actually working on our first attempt to connect to it. The URL should remain "live" while we are checking your work. You may want to move your development work to use a separate port (in the case of a web app) so that it does not interfere with our testing of this release.

Your data access instructions should be comprehensive and should list all steps necessary for us to find and access your data. For full credit, these directions must match the actual steps we need to perform to find and retrieve the resources.

Your build process instructions should be complete and correct. We should be able to successfully build and run your app on the first attempt. The build process should not be overly complicated; streamline the build process so that a reasonably intelligent developer can follow it.

Your bug-tracking system should exist when we go to examine it, and we should be able to log in and access it successfully on the first attempt. One bug must be listed per developer with suitable details such as to whom the bug is assigned and its priority.

A small part of your grade comes from the looks or aesthetics of your documents. They do not need to be beautiful or excessively formatted, but developers and your customers need to be able to read them and extract information from them. This means they should be clearly written, with proper spelling and grammar, clear wording, and formatted with enough organization to present your ideas clearly to the reader.

Your alpha release does not need to reflect "customer" interaction, but you can ask your customer to try testing out various aspects of your site before you officially submit your assignment ("Are you able to connect to the following URL?" etc.). If you make your request a reasonable amount of time before the due date, the customer will do his/her best to try to accommodate such requests in a timely manner and give you feedback about the results. You may also want to take the opportunity to tell them about any changes to your feature set that you are considering, and get their input.

One of your team members should turn in all the deliverable material together so that there is one coordinated input for the team. Put the team name in the filename of all components submitted.

### **High level summary of what to turn in:**

- URL to reach the front page of your product (Part 1)
- Source control and build process instructions (Part 2-1)
- Bug tracking system instructions (Part 2-2)
- Database access instructions (Part 2-3)

**Use the Turn-In link on the class webpage. This assignment is due by 11pm on Monday Feb 7<sup>th</sup>, 2011.**