

## CSE 403 – Spring 2006 Assignment 4

**Goal:** Develop the beta release of your product using good software engineering practices.

**Due date:** Tuesday May 9<sup>th</sup>, before 10pm, turnin –c cse403 –p beta

**Presentation dates:** Wed, Thurs, May 10-11<sup>th</sup>, in class and section

**Overview:** A large portion of this assignment is product development. Take the time to implement and test your product thoroughly. The documentation is a description of how your team is using the software engineering processes we have discussed in class.

### Deliverables

1. A beta release of your software (40 pts). A “beta” release should show very basic functionality in place, integrated, and working, for all pieces of the system. It should be possible to perform an operation that starts at the client, reaches through to the server, and back to the client. This is referred to as a “tracer bullet” in the Pragmatic Programmer (chapter 2).

A "release" includes several elements packaged in two distributions.

- a. Binary distribution. The elements needed to run your application only. A separate zip file for each host (client, server) is often a clean way to organize this. The release should include clearly identified release notes in the zip file that describe how to install and run the software and a list of any known issues. It should include initial user documentation.

If your project needs an initialized data base (or some other initialized server data set) to show its operation, provide a demo area with the data, and instructions on how to target the binary distribution to use this.

- b. Source distribution. The elements needed by someone who is going to pick up the project at this stage and do further development. Again, a separate zip file for each host target is often a clean way to organize this. The distribution should include clearly identified release notes in the zip file that describe how to build the product from the original sources.
2. A description of the implementation practices in your team, based on the first four elements of the Joel Test outline (10 pts). The list below shows the Joel Test question, a more general question for you to address in your description, and suggested artifacts that would clearly indicate how you have addressed the issue. Equivalent artifacts are acceptable, as long as they address the question. Note, do not just provide the artifact; briefly answer the question with the artifact supporting your answer.
    - a. *Do you use source control?*  
Question: What are you doing for source control management?  
Artifact: Source control dump showing source file revision levels and reasons.

b. *Can you make a build in one step?*

Question: What have you done in order to manage your build process?

Artifact: Build script, output from a run of the build script

c. *Do you make daily builds?*

Question: Do you use your build process to maintain the health of the project?

Artifact: List of build versions.

d. *Do you have a bug database?*

Question: How are you managing the tracking and resolution of defects in your project?

Artifact: Log of bugs identified, open, and resolved

3. Updated planning documents (short - list/bullet/table form is sufficient) (10 pts):
- Updated schedule, mapping out your team's plans from beta to final release.
  - Updated release feature list (spec) that you're targeting for final release.
  - Updated list of your team's top five technical risks, and your mitigation plans.
  - Updated test plan: What is in place, what is left.
  - Short (10 min) presentation/demo to update the customer (class) with your progress.

### **Turnin**

**One of the team members should turn in all the deliverable material together** so that there is one coordinated input for the team. Use the "turnin" script, "attu% turnin -c cse403 -p beta <beta files>". Put the team name in the filename of all components submitted, i.e., cool\_server.tar, cool\_client.tar, cool\_planning.doc.

### **Grading**

The beta phase review is worth 15% of your final grade, broken down as follows: 11.25% is a group score based on the deliverables specified herein; 3.75% is an individual score based on your self, peer, and staff reviews.

For the individual component of the review, you will be asked to identify for yourself and your team members: strengths to sustain, areas to improve, rating of technical contribution, rating of team contribution. More details will follow separately. This individual component will be administered separately from this turnin.

### **References**

The Pragmatic Programmer, Hunt and Thomas

The Joel Test: 12 Steps to Better Code, Joel Spolsky