Homework Assignment #4

Due Date: Thursday, July 21, before 10:00pm

Tasks:
This is an individual assignment – all questions should be answered by you, according to your thinking.
1.) If needed, familiarize yourself with the purpose of doing reflective essays (attached).
2.) Answer the following 4 questions.

Q1. *Is your zero-feature release a horizontal or a vertical prototype? Is it a throwaway or an evolutionary prototype? Explain briefly. (You may wish to consult the handout on prototyping for this question.)*

Q2. *Which of the suggested team structures in chapter 13 of "Rapid Development" did your team adopt? What were the alternatives discussed in your team and why were they considered inferior for this project?*

Q3. *The recent class discussion on usability design revolved around principles to follow for creating usable objects. Poor designs often neglect one or more of these principles.*

Describe an example of software that you personally find difficult to use. Analyze the difficulty of use – can you relate the difficulty to any of usability design principles? How would you make that software easier to use? (We suggest that you choose a very specific example, such as a dialog box that you have difficulty with, as opposed to a full program, such as Word. Showing a snapshot of a particular interaction you're describing may help to additionally clarify your point.)

In addition to the slides from class, you may find the optional reading handout "The Psychopathology of Everyday Things" helpful for answering this question.

Q4. *This question concerns the topic of unit testing. (You may find the several articles under "Useful Resources" on the course web useful as a way to approach the problems below.)*

Q4.1. *Write a simple Java class for representing a triangle. It needs to have public methods computing its perimeter and area. Now write unit tests for that class.*

Q4.2. *Imagine being hired as a consultant in a company to develop unit tests for their legacy code, some of which, unfortunately, comes without a specification. Consider the following Java class left behind by a programmer who is no longer with the company.*

```
import java.awt.geom.Point2D;

public class TOS {
    private Point2D.Double p1, p2, p3;

    public TOS(Point2D.Double p1, Point2D.Double p2, Point2D.Double p3) {
        double lp1 = Math.sqrt(p1.x * p1.x + p1.y * p1.y);
        double lp2 = Math.sqrt(p2.x * p2.x + p2.y * p2.y);
        double lp3 = Math.sqrt(p3.x * p3.x + p3.y * p3.y);
        this.p1 = new Point2D.Double(p1.x/lp1, p1.y/lp1);
        this.p2 = new Point2D.Double(p2.x/lp2, p2.y/lp2);
        this.p3 = new Point2D.Double(p3.x/lp3, p3.y/lp3);
    }

    public double tos_Obik() {
        return tos_DOC(p1,p2) + tos_DOC(p1,p3) + tos_DOC(p2,p3);
    }
    private double tos_DOC(Point2D.Double a, Point2D.Double b) {
        return Math.acos(a.x * b.x + a.y * b.y);
    }
}
```

Write unit tests for it. How do you know that what you're testing was the intended behavior?

Q4.3. *In addition to your code and the answer to the question in Q4.2, submit a screen snapshot showing the process of running your unit tests in a development environment of your choice.*

Format:

- Use at most 2 pages of text total for all of your answers, excluding the source code for Q4. Non-textual artifacts (e.g., images) are limited to 1 additional page embedded inside the same document. Put your source code in the document too, in addition to submitting it as 3 Java files.
- Format your document to be single-spaced, using font size 11 or larger.
- Save your work preferably in rich text format (RTF or DOC). We cannot edit (and add comments to) PDF without copying the contents over.

Things to remember:

- Name the file that contains your answers using the following convention:
    - *LastName*-cse403-hw4.rtf, where *LastName* is replaced with your last name.
- Put your name in the header field of your document, not only in the filename.

Submission: via UW Catalyst's eSubmit tool:

https://catalyst.washington.edu/webtools/secure/esubmit/turnin.cgi?owner=alanliu&id=3266

Follow-up: As part of our feedback, we will pose a follow-up question or two, based on what you wrote. You will be expected to answer that in writing too, though we anticipate that the answer will take you less time to produce than the original writing did.

Reflective Essays

The goal of reflective essays is to:

(a) get you to think more deeply about certain aspects of the course;
(b) practice the skill of reflecting.

Reflecting upon your experience deepens your understanding of the domain, shows it in a different light, and helps you to generate new ideas and possibilities in that and other domains. Practicing reflection is important, since this skill is critical for becoming an expert in *any* domain – only by reflecting upon what happened can you learn from successes and mistakes.

Here are some general techniques that will help you get the most from your reflections:

- Write in the first person about what happened to *you*, not to some abstract person.
- Write about things that are meaningful to *you* or happened to *you*. You will have more energy to dig into those topics.

Consider the following aspects as you reflect upon your experience:

- What stood out for you?
- What insights did it give you?
- What is *one* thing you will do differently as a result of what you learned?
- Use facts to ground your assertions.
- Do not confuse opinions with facts.

We are looking for stories about issues meaningful to you, communicated in a way that makes them meaningful to the reader (i.e., us).

Good essays have solid content, clear style, are grounded in factual assertions, and communicate well. Incidentally, good code has the same characteristics, so excellent software developers tend to also be excellent writers. In the end, writing code and writing a good essay are both about communicating effectively with your audience – an indispensable skill regardless of your domain of expertise.

In our experience, good essays often include one or more of the following aspects:

- Demonstrate how lessons from this course enabled you to do something that you were not able to do before. This may include noticing things about the world around you that you had not noticed before.
- Illustrate how this course is (or is not) changing your beliefs and ideas of what is, or is not, possible for you.
- Discuss how you have used lessons from this course in other parts of your life, or vice versa.
- Analyze situations using tools learned in this course, and discuss how well those tools worked for you.
- Discuss a situation from multiple perspectives.