### SELF-ASSESSMENT OF USU CS COURSE
#### BY INSTRUCTOR

<table>
<thead>
<tr>
<th>Course: CS 2450</th>
<th>Semester: Spring 2007</th>
<th>Instructor: SeungJin Lim</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>List Course OUTCOME</th>
<th>Assessment tool and passing criteria</th>
<th>% Passing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understand the software development process and each of its individual components.</td>
<td>Selected questions on exams and final project: greater than 75% on the questions and the project combined.</td>
<td>90%</td>
</tr>
<tr>
<td>Understand the object-oriented approach to building a software system.</td>
<td>Selected questions on quizzes, exams and final project: greater than 75% on the questions and the project combined.</td>
<td>90%</td>
</tr>
<tr>
<td>Experience the dynamics of a group project and build an object-oriented software system as a group.</td>
<td>Final project: greater than 75% of the total points.</td>
<td>95%</td>
</tr>
<tr>
<td>Understand the modern tools used to build software systems.</td>
<td>Final project: greater than 75% on the questions.</td>
<td>90%</td>
</tr>
</tbody>
</table>

Self assessments of strengths and weaknesses this semester (what worked, what did not work):

**What worked:**
- Class activities and assignments were organized to help students engage in the whole software process cycle.
- Each group was mandated to have a meeting with the instructor five times throughout the semester to discuss the major activities in the software process.
- The project was selected by each team.
- The programming language to use was selected by each team.
- The lecture notes, slides and largely multiple choice exam seemingly worked well for a better communication.

**What did not work:**
- Students wanted to see more real examples of artifacts such as good and bad design examples of software.
- There were too many concepts, principles and practices to present to the class.
- The user requirements analysis assignment was not very clear partly due to the fact that each team selected own project idea. Who is the user who is supposed to give feedback?
- Small number of students misunderstood assignment requirements. It is difficult for me to understand why.
- I couldn’t involve students during the lecture as much as I wanted.

Recommendations of changes to implement next offering:

- It may be worth to streamline the course materials to the essence of software engineering.
- I like to focus on selected two concepts in each class period. I like to present some poor examples/artifacts of the concepts to present and drive the students to find a better principles and practices.
- I like to give students a second chance for each artifact.
- I like to present advantages and disadvantages of choosing a particular programming language, and its impact throughout the semester.
- I like to let each team be the customer to some other team.
- I like to introduce some fundamental tools such as Eclipse at an earlier stage of the semester.