CS 471/571 Team Project

The objective of this team project is to help you get a firm understanding about the key software development activities through the engineering of a realistic product.

1. Grouping

In general, each team should have 3..5 members. Each team should have a coordinator, e.g., for taking lead in setting up group meetings, making contact with the instructor, and integrating the project reports. You are not supposed to work alone even if you can. Teams must be formed before the third week of the semester. Each team should choose a name that can appear in public. **Each *team coordinator* should email the instructor the name of their team and the names of all the team members including the coordinator**. Contact the instructor if you cannot find a team.

2. Problem Description

Your customer is asking you to develop software of the board game Mill, also known as Nine Men's Morris (http://en.wikipedia.org/wiki/Nine_Men's_Morris). The software should allow a human player to play against either a human or a computer opponent. The computer opponent should make a reasonable attempt at beating the human player (optimal or perfect play is not required). You can play against existing web-based programs to get ideas for your game (e.g., http://www.smartlittlegames.com/ninemensmorris). You may determine the functional and quality requirements according to the rules of the game (e.g., placing pieces, moving pieces, and flying), the problem description, and your imagination. You may implement the game as a standalone program, a web-based applet, or a mobile app using a **general-purpose object-oriented programming** language (e.g., Java, C#, C++). You should use a public project hosting site such as code.google.com or github.com, and a version control system such as subversion or git, unless you have made prior arrangements with your instructor.

The project consists of two phases that apply different software engineering principles and practices.

Phase I creates a working prototype of your team's product using agile development techniques, including prioritized user stories, pair programming, test driven development, refactoring and acceptance criteria. The entire team must work together to define an agile product backlog, including user stories, their acceptance criteria, and their estimates. Each member must be paired with a teammate to practice pair programming with test-driven development. The minimum time of paired development by any one member is 8 hours during which swapping of roles is required.

Phase II produces a ready-to-release product with detailed documents of requirements and design using UML tools (e.g., use case, class, and sequence diagrams), and specific test cases for validating the requirements and design. Development tasks should be divided among team members. Interaction and collaboration among team members are required. The documents must be consistent with your submitted source code.

You should keep track of how much time you spend doing each of the software development activities and what errors or misunderstandings you encounter. Monitoring your own performance will help you to improve.

3. Reporting and Grading Policy

Reporting requirements include midterm/final presentations and midterm/final project reports. Your instructor will provide additional details.

(1) Midterm Presentation/Demonstration 5%.

(2) Midterm Report 35%.

(3) Final Presentation/Demonstration: 15%.

(4) Final Report: 45%.

Extra credit: Up to 5% may be added to your ***final grade*** for quality or other enhancements, for exceptionally well-written reports, and for overall impression of the project, which the instructor deems to be deserving of special recognition. Example: Collaborate with another project team to test each other’s product, track each defect found, and report the status (found/fixed) of each.