CS 360 Software Development  
Spring 2008  
Tuesdays and Thursdays 3:30 p.m. – 4:45 p.m.

Instructor: Ingrid Russell  
Office: Dana 343  
email: irussell@hartford.edu  
http://uhaweb.hartford.edu/irussell

Course Description: This course focuses on the design and development of large-scale software that is reliable, understandable and maintainable. Students will work in teams on a semester long software development project. Project organization, professional standards and ethics will also be covered.

Course Prerequisite: CS 220

Required Text  

Optional Text  

Grading policy  
- Semester-long project: 60%  
- Exam 1: 15%  
- Exam 2: 15%  
- Participation: 10%
<table>
<thead>
<tr>
<th>Chapter 1:</th>
<th>The Scope of Object-Oriented Software Engineering: Historical aspects, economic aspects, maintenance aspects, requirements, analysis, design, team development aspects, ethical issues.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 3:</td>
<td>The Software Process: The Unified process, iteration and implementation, requirements workflow, analysis workflow, design workflow, implementation workflow, test workflow, post delivery maintenance, Phases of the Unified Process.</td>
</tr>
<tr>
<td>Chapter 4:</td>
<td>Teams: Team organization, teams for agile processes, open source programming teams.</td>
</tr>
<tr>
<td>Chapter 5:</td>
<td>The Tools of the Trade: Stepwise refinement, cost-benefit analysis, software metrics, software versions, configuration control, build tools.</td>
</tr>
<tr>
<td>Chapter 6:</td>
<td>Testing: Software quality assurance, non-execution-based testing, execution-based testing, what should be tested, testing versus correctness proof.</td>
</tr>
<tr>
<td>Chapter 7:</td>
<td>From Modules to Objects: The Object-Oriented paradigm, cohesion, coupling, data encapsulation, information hiding, inheritance, polymorphism.</td>
</tr>
<tr>
<td>Chapter 8:</td>
<td>Reusability &amp; Portability: Reuse concepts, reuse during design and implementation, design patterns, portability.</td>
</tr>
<tr>
<td>Chapter 9:</td>
<td>Planning &amp; Estimating: Planning and the software development process, estimation issues, components of a software project, software project management plan.</td>
</tr>
<tr>
<td>Chapter 10:</td>
<td>The Requirements Workflow: Client needs, understanding the domain, initial requirements, requirements revision, rapid prototyping, metrics for the requirements workflow, challenges of the requirements.</td>
</tr>
<tr>
<td>Chapter 11:</td>
<td>The Analysis Workflow: Extracting the entity classes, functional modeling, entity class modeling, actors and use-cases, challenges of the analysis workflow.</td>
</tr>
<tr>
<td>Chapter 12:</td>
<td>The Design Workflow: Object-Oriented design, the design workflow, formal techniques for detailed design, metrics for design, challenges of the design workflow.</td>
</tr>
<tr>
<td>Chapter 13:</td>
<td>The Implementation Workflow: Choice of programming language, good programming practice, Coding standards, code reuse, integration, the implementation workflow, test case selection, testing techniques, testing issues, unit testing, integration testing, product testing, acceptance testing, metrics for the implementation workflow, challenges of the implementation workflow.</td>
</tr>
<tr>
<td>Chapter 14:</td>
<td>Post delivery Maintenance: post delivery requirements, management of post delivery maintenance, maintenance issues, post delivery maintenance skills versus development skills, metrics for post delivery maintenance, challenges of post delivery maintenance.</td>
</tr>
</tbody>
</table>
Software Project
This is a team-oriented semester-long object-oriented software development project with deliverables throughout the semester. The goal of this project is to produce a useful, user-friendly, and powerful quality software package, a final product that everyone would want to actually use. The project will utilize object-oriented and software engineering techniques and should be written in Java.

The goal of the App-Trac project is to develop an application that will allow the Literacy Volunteers of Greater Hartford (LVGH) to monitor usage of literacy software applications in order to better evaluate students learning needs and streamline report generation for management.

LVGH does not have an independent system to monitor students’ usage of the organization’s literacy application suite, which consists of a mixture of applications, from third party vendors. Currently the technology manager at LVGH has a rudimentary clock-in system to allow users to log when they use the computer. The manager manually accesses individual literacy application backend databases to export usage information to Microsoft Excel spreadsheets, which are then combined by hand to generate specific use data statistics for management and instructors. This present approach does not allow LVGH to gather all the statistics required, due to the overhead associated with the manual data retrieval and collection process.

App-Trac is an attempt to introduce an automated system to help streamline the data gathering and processing, to help alleviate a significant amount of human overhead, freeing up valuable staff time and help reduce associated operational costs.

The project will be divided into smaller assignments of varying length that address the various phases in the software development life cycle. These are used as milestones in the development of the project and are due at various times through the semester. Proper documentation and a user’s guide should also accompany your final software package.

These smaller assignments are used as milestones in the development of the project and will consist of various deliverables including:

1. Requirement specification document (RSD) (15%)  
   Due Date: February 12, 2008
2. RSD– Revisited + presentation (10%)  
   Due Date: February 19, 2008
3. Design Model + presentation (20%)  
   Due Date: March 6, 2008
4. Test plan document (15%)  
   Due Date: March 13, 2008
5. Software Release (25%)  
   Due Date: April 22, 2008
6. Class presentation (15%)  
   Due Date: April 29, 2008
**Project Rules:**

Each deliverable will include additional details about the various rules listed below.

- Come up with a team name and a project name. Be creative!

- Each team member is expected to contribute equally to each portion of the project. Specifically, each team member is expected to contribute to the specifications document preparation, and to the overall design. In addition, each team member should be responsible for a specific portion of the detailed design, implementation and coding, and the final product documentation.

- Each team must select a team leader. The team leader is responsible for coordinating and supervising the work of the deliverable and upon consultation with team members, assigning or reassigning people to various tasks. This should be a rotating role among the members and rotated among deliverables. Some useful tips to the team leader:
  
  - Create a clear agenda addressing the essential tasks that must be accomplished in order to complete the necessary deliverables.
  - Stick to the agenda items during a meeting.
  - At the end of each meeting make sure to identify action items and do not adjourn until every member is clear on what is he/she expected to do.
  - Ensure that tasks are equitably distributed.
  - Select a recorder to record the minutes of meeting for the duration of the deliverable portion of the project and make sure minutes are posted soon after the meeting is over.
  - Select the next meeting time and location.

- Each team must select a recorder to record minutes of all meetings. This should be a rotating role among the members and rotated among deliverables. The minutes should include:

  - Date and time of meeting
  - Team leader name and recorder name
  - Members present
  - Summary of discussions with relevant details and a list of action items.

- The recorder shall also be responsible for maintaining a project online notebook. The project notebook should include current versions of all documentation, specifications, design, schedule, etc. Documentation is an essential part of this project and the notebook can be used in preparing the user documentation. Note that this does not necessarily mean that the project recorder will do all the work but just coordinate it. It is the responsibility of the recorder to distribute these tasks among team members.
Each team member should keep a log of the time spent on this project with specifics as to dates, tasks, and amount of time spent on each task. The logs must be up to date at all times and available online for me to review as I grade each deliverable. I will review all logs and provide comments for revisions. These logs should also provide guidance to you in future software design projects. Each team member should maintain a webpage to post the logs. Your project website should include links to team members log web pages.

Each team member and team is expected to adhere to all deadlines. Failure to do so may impact the progress of the project which in turn will impact your grade on the project and may result in a late penalty.

In addition to being evaluated by me, your work will be evaluated by your team leader, team members, and by your client.

A limited amount of class time will be used for team meetings in order to give you an opportunity to receive feedback from me. However, most team meetings should be scheduled outside of class time. Team members are expected to attend all team meetings.

Each group should submit one copy of the document due as each stage with names of all team members on it. In addition, each team member must submit 1-2 pages summarizing his/her contribution to the deliverable submitted. Your team will receive a grade on each deliverable. Each team member will also receive an individual grade. Your grade on each deliverable is the average of your individual grade and your team grade.

Each piece of code and documentation should clearly identify the author and date created.

If a student withdraws from the class, the team will work toward the same set of goals with the current members. If a team is left with 2 members, arrangements can be made to adjust the requirements.

Finally, working in a team can be fun and at the same time challenging. As we work on this project we will discuss issues related to teamwork. Sometimes, it is tempting for a team member to work alone or do most of the work. You should avoid the temptation to do so. You will find out that team members have different work habits, personalities, and opinions. You should respect your team members’ points of view, even if you disagree. You will need to work around your differences. If there is a conflict, you should try to resolve it among yourselves. Only if you are not successful should you request my involvement to resolve the conflict.

Note: This project handout will be revised throughout the semester as we add the various deliverables.
**Deliverables**: Each assignment includes a list of deliverables and deadlines. You must have a webpage to post all project related assignments. You must also submit hardcopies for all assignments by the deadline specified in each. These deliverables must all be compiled and again submitted with the final project along with proper system documentation by April 22, 2008. Each group is expected to present their project in class. These presentations will be held April 29.