



Department of Computer Science and Software Engineering

Senior Project 2

CSSE 498 Winter Term 2012-13

Instructor(s): Shawn Bohner

Email: bohner@rose-hulman.edu

Phone(s): Shawn - 812-877-8685 (office)

Office: Moench Hall, Shawn- Room F212

Office Hours: Check with Senior Project Advisor at first meeting. Or by Appointment.

Introduction: Software projects are challenging efforts and as CSSE students prepare to enter the workforce, experience is needed. In the Junior Sequence of courses (CSSE 371/372/374/375) students gained experience with key aspects of software engineering as they completed a year-long junior project. Topics covered included:

- | | |
|---|--|
| 1. Software requirements analysis & specification | 4. Software construction & evolution |
| 2. Software project planning & management | 5. Forming and managing project teams |
| 3. Software architecture & design | 6. Software risk planning and management |
| | 7. Software configuration management |
| | 8. Software validation & verification |

Senior projects follow a similar approach, but the students play a much more directive role in their projects by working with the clients to define the projects (typically 1400-1600 hours of effort for 4 team members). Rather than meeting in class, there are weekly meetings with the faculty advisors and the clients. The focus for the students is largely on the full development of a software system for a real-world client where the students define, plan, develop and deliver a usable product. CSSE 497 is the first of the three terms for this year-long project. Students are graded each term for their performance and participation on the teams, as well as using input from the clients.

Course Description: Group software engineering project requiring completion of a software system for an approved client. Tasks include project planning, risk analysis, use of standards, prototyping, configuration management, quality assurance, project reviews and reports, team management and organization, copyright, liability, and handling project failure.

Learning Outcomes: Upon successfully completing this series of senior project courses, a student should be able to.

1. Demonstrate effective Communication Skills
 - **Reading:** read technical documents and offer constructive criticism of their content and style
 - **Writing:** write several different types of technical documents
 - **Oral presentation:** prepare and deliver technical material at the appropriate level of detail
2. Demonstrate effective Management Skills
 - **Leadership:** lead a small software team (if team leader) or ability to support the leadership of the team (if not)
 - **Time management:** estimate and monitor personal time across multiple tasks
 - **Meeting facilitation:** lead and participate in small groups in constructive meetings

- **Estimating:** estimate effort required to complete technical tasks
 - **Risk:** assess project risks and plan mitigation strategies
 - **Planning:** prepare a feasible plan for the accomplishment of several technical tasks
 - **Monitoring:** track the progress of several tasks according to a plan
3. Apply Technical Skills on a real-world problem.
- **Analysis:** analyze technical requirements and proposals for feasibility and to model the consequences of proposed solutions
 - **Design:** construct appropriate abstractions of problems and solutions
 - **Coding:** produce and inspect implementations of software according to project standards
 - **Testing:** prepare test plans and to participate in both unit-level and system-level testing activities
4. Demonstrate key areas of Professionalism
- **Ethics:** identify and prevent unethical professional behavior
 - **Intellectual property issues:** make appropriate professional judgments regarding choice of methods for protecting intellectual property
 - **Social issues:** evaluate/avoid possible negative social aspects of a software product
 - **Relationships with clients:** interact with clients in a professional manner
 - **Teamwork:** working collaboratively with the team to deliver the project and accomplish other project goals.

Prerequisites: CSSE 374 and CSSE 497.

Textbooks: Readings may also be assigned from relevant papers (e.g., case studies).

Course Evaluation and Feedback: Please feel free to provide feedback about the course at any time. If you feel uncomfortable talking with me directly, there is an anonymous feedback box under the ANGEL account for this course where you can provide feedback throughout the term; I check it regularly and will try respond to feedback in a reasonable time. There will also be two anonymous plus-delta evaluations of the course where you can offer suggestions on how to improve the course and its delivery.

Grading:

Meetings	20%	Documents (reviews)	15%
Engineering Journals	10%	Presentation(s)	10%
Project deliverables	30%	Client Feedback	15%

Expectations: Students will be expected to attend and participate in senior project related meetings. Students will be required to use the CSSE497 course website on Angel to obtain relevant information, and interact with instructor and other students. Announcements and assignments will be conveyed via Rose-Hulman email addresses and/or posted on the website. Students will be expected to work on most assignments with other team members.

Assignments: Project assignments are somewhat different than our usual classes. Please provide access to your software project material/artifacts in your configuration managed repository to the instructor. Some assignments may be requested separately (e.g., a response to a case study). Unless otherwise instructed, please post these on Angel in the associated drop boxes. While this course is demanding, it is also rewarding for those that want strong understanding of software engineering as a discipline.

Late Submissions: Please note that project assignments and deliverables will be due at the specified time on the specified day, in the format agreed upon with advisor. Late assignments and project deliverables will also not be accepted (receiving a zero for grade).

Academic Integrity: CSSE Honesty Policy (see <http://www.cs.rose-hulman.edu/index.php/courses-mainmenu-28/82-honesty-policy.html>) governs class and performance. Joint study is allowed (even encouraged) on some items as expressed by the instructor; however, each student must produce his or her solutions individually. Students must not collaborate on tests or homework that is passed in unless directed by the instructor.

Attendance Policy: Attendance of meetings is mandatory (unless with a legitimate excused absence such as illness). If you cannot make it to meeting, you are still responsible for all information covered in the meeting. Students who have more than 2 unexcused absences will receive a final course grade reduced by up to one full grade level; a student whose total absences (excused and unexcused) exceed 8 will fail this course.

Valid Excuses: A valid excuse consists of a memorandum on Institutional letterhead from the Dean of Students. Job and graduate/professional school interviews, attending scientific conferences and Institute-sponsored activities are also valid excuses provided that every attempt has been made to avoid missing major assignments and examinations, and the student notifies the course instructor in writing at least one week in advance of the event. Illness and exceptional circumstances are, of course, valid excuses if a confirming memorandum from the Dean of Students is provided within one week of the illness/circumstance.

Writing: Written communication is important in CSSE 497, as it is in the software profession in general. Remember that a software document has several unique and important characteristics:

1. Technical documents are often the result of group authorship, thus it requires planning and final tweaking.
2. Specificity and organization are more important than flow; hence technical documents are often ordered around lists and tables rather than paragraphs.
3. Documents are often the reader's only source of information on the particular subject or product; hence they must be thorough and complete.
4. Documents are often used to answer specific questions; hence, they should facilitate finding specific pieces of information (navigation).
5. Documentation must bridge from general specifications to particulars of implementation and operation, hence it must make abstract concepts concrete and make concrete facts fit generalized concepts.
6. Documentation can be presented in many forms: online via HTML, MS help files, just plain text, and on paper as reference manuals, tutorial, quick reference guides, etc. It is important to choose the correct medium and even more important to write to fit the medium.

You can always drop by your project advisor's office if you have any questions regarding your documents. I would be happy to look at it and suggest improvements. You should also be aware of the service provided by the Learning Center.

Professionalism: As would be expected in the workplace, you are asked to behave in a professional manner. This includes your appearance, such as your apparel, your hygiene, as well as potentially offensive computer desktops. We furthermore ask that you turn off your cell phones during meetings, let people finish their turn talking and do not be disruptive in other ways. Violations of this policy will have a negative impact on your "Meetings" score.

Caveat: The instructor reserves the right to modify the course content, schedule, policies, etc. outlined in this syllabus.

Tentative Schedule of Course Deliverables: Please check online for latest updates daily.

Week Date	Project Assignments & Deliverables Due	Topic Content & <i>Notes</i>
1-Wed 11/28	Establish Time(s) for Weekly Meetings (Client, Advisor, & Team meetings)	By 9:00am Friday, 1 st week
2-Mon 12/3	First Weekly meeting	- Discussion of term contents - Reflection on Fall Deliverables - Setting expectations for the term
3-Mon 12/10	Improved/Updated Architecture and Design Document	- Have Requirements Material also to inform the design
4-Mon 12/17	Discuss Wednesday's Presentation	
4-Wed 12/19	Present Design & Demonstrate Prototype (showing design feasibility)	- Presentation during 7th-9th periods
12/22 - 1/6	Christmas/Winter Holiday Break	Enjoy!
5-Mon 1/7	Midterm Peer Evaluations	- On Angel and due by midnight.
6-Mon 1/14	Architecture and Design Review	- Architecture and Design Document(s)
7-Mon 1/21	Architecture and Design Document Signoff (by client)	
8-Mon 1/28	Updated Project Document(s) Audit	- Project Document(s) due by midnight Monday
9-Mon 2/4	1. Dry run of Sr. Project Expo Poster 2. Working prototype shown to client	<i>Note: Informal presentation at Weekly Advisor Meeting.</i>
10-Wed 2/11	1. Senior Project Exposition 2. Documents Due (paper version) 3. Final Team Peer Evaluations	- Wed., Feb 15, (10:00 a.m. - 2:00 pm), in the lobby of the Union building. Teams will have tables, etc. Clients and the press will be invited. - Document(s) and Team Peer Evaluations due by midnight Wednesday
Finals Week	TBD	

Senior Project

Winter deliverables (rev 11/27/2012)

The following are brief descriptions of the Winter Term deliverables along with some relevant instructions and recommendations. These deliverables are a starting bid. If your team feels that any part does not apply to your project, please make your case.

1) Meetings

Each week, the project team will meet with both your client and project advisor. By the end of the first week, you will provide the day and times for these meetings to your advisor.

What	Notes
Weekly adviser meetings	<p>For each weekly meeting with the team's advisor, a message must be sent in advance of the meeting (i.e., by 9:00am) indicating the following:</p> <ol style="list-style-type: none">1. A list of tasks completed this week along with how much time they took2. A list of tasks planned for next week along with an estimate of how much time they will take,3. A list of Red and Amber risks4. A list of issues/requests for resources/observations relevant to the project.
End-of-term document audit	<p>At the regular advisor meeting, Week 8, we will go through an "audit" of your documentation. More details to follow. Basically, look for things like, "Ok, let's see where that feature was done in your project plan; let's see the test cases for it,..."</p>

2) Document focus for Winter Term

Below is a list of documents, their delivery times, and general document expectations. There are no templates/examples given this quarter on purpose. Your format may vary and content, depending on your project and processes. Please also note that good documentation practices are expected on each of these documents. In some cases, we may exclude or change the meaning of some of these documents to fit your project. If so, you need to get an explicit agreement from your faculty advisor that this is the case, and not just have a casual conversation about it!

Document	Wk 5	Wk 10	Notes
Engineering journals (for each team member)	✓	✓	Continuation of journals we asked for in the fall.
Project plan	✓	✓	<ul style="list-style-type: none"> • Up to date schedule • Risks <ul style="list-style-type: none"> ◦ Updated and added to (includes all risks) ◦ Mitigation strategy w/ revisit date (for all risks) ◦ Resources responsible • Updated configuration management <ul style="list-style-type: none"> ◦ Includes build/release schedule • Overall completeness <ul style="list-style-type: none"> ◦ Includes structure for next quarter ◦ Good documentation standards • Metrics <ul style="list-style-type: none"> ◦ Appropriate effort for 4 cr. hr. course ◦ Note that most of the grading for your metrics will be based on these being up to date. Typically these include how much time each team member spent each week, for example. • Event tracking <ul style="list-style-type: none"> ◦ Show history of what actually happened, versus plan. This includes what was done on time, and what wasn't.
Architecture/ Design	✓	✓	<ul style="list-style-type: none"> • Transition from the problem statement and requirements • Module design & component/connector design • Data model/design • Describe design details of algorithms and functional interfaces used • Provide a guide for implementation <ul style="list-style-type: none"> ◦ Anyone on the team could code from this independently • Evidence of changes from the action items in the Week 5 design review (after this has occurred)
Test plan	✓	✓	<ul style="list-style-type: none"> • How testing is being done (test environment, test setup, etc.) • Test cases (text) <ul style="list-style-type: none"> ◦ Acceptance, System, Integration and Unit • Test procedures <ul style="list-style-type: none"> ◦ These should be as automated as possible • Test results • Easily trace to requirements/design • Evidence of a team peer review over these
Deployment/ installation plan		✓	Details how the product is deployed and installed in the customers environment
Working system	✓	✓	Graded at the presentations/expo Includes getting customer feedback
Initial User documentation		✓	Should contain information for both: <ul style="list-style-type: none"> • Actual users • System administrators
Initial Maintenance plan		✓	<ul style="list-style-type: none"> • Prioritized list of features for spring term • Plan for getting and using customer feedback • Plan for supporting the customer

3) Engineering Journal

- Each team member has an electronic log, documenting what you did, how long it took, decisions your team made, and how those were made. The idea is to keep adding to this daily and weekly. The instructor will be periodically looking at these (typically weekly). Continue from Fall term.

4) Configuration Management Plan – Continue from Fall Term

5) Metrics – Continue from Fall Term

6) Requirements Document – Continue from Fall Term