Department of Computer Science and Software Engineering

Software Construction and Evolution

CSSE 375 Spring Term 2014
Your Sections (2 & 3)

Class Times:

- Sec 2: MTR 5th Period, W 1-3 Period
- Sec 3: MTR 6th Period, W 7-9 Period

Meet: in Myers M 115

Instructor: Steve Chenoweth
Email: chenowet@rose-hulman.edu
Phone(s): 812-877-8974 (office
Office: Moench Hall, Room F220
Course Assistant: Adam Westerman

Introduction

Here's a course description, written in free verse to be amusing:

a. Constructing and delivering a software product entails
    a number of tasks to ensure that a software product is ready to go into production.
b. For the Software Engineering program, this course provides a final course
    in the Junior sequence,
    which ties a lot of things together from courses taken previously.
c. In this course, Junior Project Teams deliver their project to their clients.
d. **New ideas** in the course include the following:
    e. **Refactoring:** You will examine the methods for finding problems
        in the source code and design (bad smells).
    From the bad smells you'll use *lots of* refactoring techniques
        to reformulate the software into a more understandable and maintainable form.
    f. **Exception handling and fault tolerance:** You will learn good ways to
        deal with these concerns systematically in your project.
    g. **Stick with it, or redo it:** You'll discover what's different in keeping up an old system,
        and the "good" way to rework the whole thing when you have to.
    h. **Forks and load lines:** You'll experience more sophisticated control of code bases.
    i. **Code analysis:** We'll do some static code analysis, like
        what affects what else if you change it, and
        what speeds everything up.
    j. **The dreaded "D" word:** You will also learn about user documentation
            along with other documentation that normally gets delivered with a product.
k. Through the texts,
        a series of directed readings,
        active learning exercises, and
        code review exercises,
    you will be exposed to a blend of classic and emergent techniques
        to grasp a thorough understanding of key software construction and evolution principles.

➔ **An general goal of this course is to prepare you for senior project, which follows immediately!** Indeed, the organizing meeting for next year's CSSE senior projects will be while 375 is going on. In senior project, you will work more autonomously, proving that you can exercise the skills you learned in your first three years here. That will include what you get in 375. We will try to arrange it so that there is *at least one SE* on every team that also has CS-only majors. In this way, you will be a source of expertise to those students on the SE topics they did not learn, like what you got in CSSE 372, 375, and 477.

Topics to be covered in this course include the following:

1. Basic principles of software change
2. Program understanding and tools
3. Detecting bad smells
4. Applying software refactoring
5. Exception handling for fault tolerance
6. Software maintenance
7. Software modernization (reengineering)
8. Software configuration management
9. Software user and system documentation
10. Adv. Topics: Impact Analysis, code tuning

Course Description (from the course catalog): Issues, methods and techniques associated with constructing software. Topics include detailed design methods and notations, implementation tools, coding standards and styles, peer review techniques, and maintenance issues.

Learning Outcomes (longer versions of the *above* topics): Upon successfully completing this course, you should be able to.

1. Work with your ("big") junior project team to complete and deliver the junior project to the client. In doing so, demonstrate the ability to work within a team to deliver a multi-term-sized project to an external client successfully.
2. Apply appropriate refactoring techniques to resolve design problems in code.
3. Apply common construction and maintenance heuristics to enhance existing code, such as ways to eliminate global variables and ways to test difficult code.
4. Organize and develop software user documentation, which enhances long-term software viability.
5. Construct software so that it meets delivery and deployment objectives specified by the project.
6. Apply the corrective, perfective, adaptive and preventive types of software changes and maintenance types.
7. Apply impact analysis and other software source analysis to understanding existing software.
8. Use systematic exception handling and other techniques in promoting fault-tolerance.
9. Describe software modernization approaches such as reverse engineering, reengineering, salvaging, and restructuring.
10. Describe the ways configuration management is used in production systems.

Prerequisites: CSSE CSSE 374. (Software development and/or maintenance experience, and an ability write and communicate effectively will make this course more meaningful.)

Textbooks (2 required):

- *Working Effectively with Legacy Code*, by Michael C. Feathers. Publisher: Pearson Education, Prentice-Hall ISBN-10: 0-13-117705-2

- *Refactoring: Improving the Design of Existing Code*, by Martin Fowler Publisher: Addison-Wesley Professional; 1 edition (July 8, 1999) ISBN-10: 0201485672

Course Evaluation and Feedback: Please feel free to provide feedback about the course at any time. I believe that you will learn a lot by talking with me face-to-face about issues or problems – practice for the "real world" you'll be working in. Toward that end, discussions in my office are strongly preferred when the topics are serious or personal. There will be at least one anonymous plus-delta opportunity, as well.

Grading

Examinations (Probably 2)	25%
Daily Quizzes and Class Participation	10%
Homework / "Swap" Assignments (we'll discuss these)	25%
Junior Project Participation, Project Work & Presentations	40%

Expectations

Overall, this class is a junior-level practice session for working in the software industry.

You will be expected to attend and participate in class and at team meetings outside class. You will use the CSSE375 course website (http://www.rose-hulman.edu/class/csse/csse375/) 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.

Assignments: Homework or project assignments will be assigned regularly. Unless otherwise requested, please post your work on Moodle in the associated drop boxes. Homework and projects are necessary instruments for tracking progress of students. Expect to work approximately 8 hours outside of lectures each week on this course (depending on your background and the project work for a sprint). This is a demanding course covering a great deal of material -- please avoid falling behind on the assignments. 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 homework and project deliverables will be due at the specified time on the specified day, in the specified format. Late quizzes, homework assignments, and project deliverables will also not be accepted late (receiving a reduced grade, down to zero at my discretion).

Academic Integrity: CSSE Honesty Policy governs class and performance (see http://www.rose-hulman.edu/Class/cs/HTML/about/honesty.html). 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. On daily quizzes, the policy is for help to be shared equally. You can ask other people for help on a question, but then you should be a resource for them, on other things. In no case should you just copy what someone else said – the quizzes are to promote your own thinking.

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

Question – Do I have to attend all of the lab time and work on the project then (on Wednesdays or otherwise)? Answer – You can do as you please if the work for your next milestone / sprint is all complete. If you have not shown me that this is the case, expect me to ask about that. An absolute no-no is for you to be absent or working on other things while your teammates do the project work![1]

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.

Laptop Policy: You may need to use your laptops during some portion of the class period. Please be sure to bring your laptop, a power brick, and a network cable to class. During class discussion, please do not use your laptops. Laptop use during discussions can be distracting to your classmates, the instructor, and may also keep you from focusing on the material. If you typically use your laptop for note taking, please get permission from the instructor in advance so arrangements can be made.

[1] This is a direct application of Kant's Categorical Imperative. Things would turn out badly if everyone did it.

Writing: Written communication is important in CSSE 372, 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 my office or consult with your project manager, if you have any questions regarding your document. We would be happy to look at it and suggest some changes. You should also be aware of the service provided by the Learning Center.

Tangible Distractions (Sights/Smells/Sounds): As would be expected in the workplace, please be respectful of those around you. If your visual appearance (e.g., offensive computer desktops), smell (e.g., hygiene), or sounds created (picture cell phones and snoring) are disruptive to class, you will be asked to leave until the issue can be corrected. As you know, I am liberal about dress code – I believe that modeling difference gives you a chance to see life from alternative perspectives, which builds a crucial skill for the software business. In all these areas, the bottom line is your awareness of others and your motivations for what you do.

Caveat: We instructors reserve the right to modify the course content, schedule, policies, etc. outlined in this syllabus, as needed while the course progresses.

Schedule of Classes and Assignments

Please check online Class Schedule from the course website, for the current schedule and latest assignment updates.