Syllabus - COMP 474
Software Engineering
Dr. William L. Honig

Course Description
The course is an advanced experience in Software Engineering, the application of managerial, organizational, and technical methods to large-scale software development. Software Engineering will be taught in such a way as to expose the student to a “real world” or “industrial strength” development process and programming. Students will work as a member of a large programming team for the entire course. In addition to implementing a software application, the team will be responsible for creating plans, reviewing progress, measuring quality, and making written and oral reports on the team’s status. The team will learn and use the Team Software Process (TSP, http://www.sei.cmu.edu/tsp/), the leading industry methodology for high quality software development.

Prerequisites and Key Information
COMP 271 Object Oriented Programming; COMP 313 Intermediate Object Oriented Development recommended: Students should have an advanced knowledge of software development, ideally using Java, proven skills in writing, compiling, and debugging software, and experience with database management software. Students must also have the style and communications skills to work effectively with a group of other students. An undergraduate course in software engineering, although not required, will be very helpful to the student in Comp 474. Such courses typically cover topics including project planning, software scheduling tools, object oriented system design, and software testing.

Special Course Requirements
1. Students will be expected to spend considerable time beyond formal class sessions on this course. Each student will be part of a large programming team that will be required to meet at regular times once or twice a week with each session lasting 1 to 2 hours. Students are expected to attend all these team meetings in person throughout the course; if you cannot make and keep this commitment, you will not succeed. In addition to the team meeting, each student will need to devote time to their portion of the team’s software and generating of plans and metrics on their work; this individual programming time commitment will be similar to other programming intensive courses. Expect a total time requirement of at least 10 to 15 hours a week in addition to class room time to be devoted to team meetings, your own work on documents and software for the team, and personal reading and study of course materials.

Comp 474 Software Engineering Syllabus, version 4
2. Students may also have difficulty with key course learning areas, especially “process” concepts. Software engineering with real world process methods is very different from an individual programming course. It will seem slow and awkward to do the necessary planning, documentation, and reports. Expect to be frustrated at some point during the course. This feeling is a necessary and valuable experience for the real world and inherent in the team learning process.

3. The course may include some online components using Loyola University Chicago tools such as Adobe Connect. You will need to have good internet access and proper software on the computer you use for online work. See http://www.luc.edu/itrs/teachingwithtechnology/adobe-connect.shtml for more information.

4. The course will use Sakai to organize some materials. You will generally submit your assignments using Sakai. Ask for help if you are not familiar with Sakai. (and see http://www.luc.edu/itrs/teachingwithtechnology/sakai/sakai-student-tutorials.shtml)

Textbooks


The primary text for learning the Team Software Process, Introductory (TSPi); this text will guide the team software project through the required phases of development. The book includes materials on team roles, planning and design, team metric generation, and quality improvement. The TSPi will be followed closely in class and by the teams; all students will need to become very familiar with this text.


A secondary text for learning software engineering processes and methods other than TSPi. This text will be used to augment the information in the primary text, in the areas of software configuration management, project planning, requirements engineering, software validation and testing, metrics, and cost estimation. The book contains a wealth of additional information that we will not have time to cover in the course; it will be useful future reference for those who go into industry in software roles. You should consult this text as necessary, including its online companion materials, to help you understand topics throughout the course.

Academic Honesty

Students are expected to have read the statement on academic integrity available http://www.luc.edu/academics/catalog/undergrad/repacademicintegrity.shtml. This policy applies to the course. The minimum penalty for academic dishonesty is a grade of F for that assignment. Multiple instances or a single severe instance on a major exam or assignment may result in a grade of F for the course. All cases of academic dishonesty will be reported to the department office and the relevant college office where they will be placed in your school record.

Academic dishonesty includes, but is not limited to, working together on assignments that
are not group assignments, copying or sharing assignments or exam information with other students except in group assignments, submitting as your own information from current or former students of this course, copying information from anywhere on the web and submitting it as your own work, and submitting anything as your own work which you have
not personally created for this course. If you do wish to use materials that are not your own, please check with me ahead of time and cite your source clearly.

In addition, I wish to emphasize issues relevant to this course.

**Discussion and interaction encouraged on team products**

Software engineering is inherently an interactive, person-to-person activity. One of the key things you will learn in the course is how to work together in a group to produce software. For this reason discussion and sharing of ideas, software, problems, and solutions is encouraged in the work of your team. Sharing of information and materials between different teams is also allowed and encouraged.

**Individual work entirely your own**

Work other than your team products is to be entirely your own. Individual work includes quizzes, examinations, individual home work assignments, and all other course assignments not specifically described as a team product. You are not to discuss solutions to these individual assignments with others, nor are you to use solutions you find from other sources. Individual work submitted in the course is to be entirely your own and original creation.

**Use of web resources**

Materials copied from the web are the same as copied from another student and are not allowed on any course work (team or individual products). This prohibition applies to all web-based materials and specifically to prior course notes and web sites, textbook web sites, industry web sites, software parts or components, and other university web sites. Your goal is learning, not just turning something in to meet the assignment due date. *Never represent something as your own work when it is not!*  

**When in doubt, ask!**

Please ask any question at any time on the issue of academic integrity and honesty. If you think something should be allowed, ask before you use it. I am always willing to make an individual, case-by-case decision if you ask before using something. There may be cases when using something from other sources is the best way to enhance the learning experience in the course.

**Course Objectives and Goals**

Upon the successful conclusion of the course, the student will:

1. Become experienced in process based software development in general and in particular the Team Software Process Introductory (TSPi). Be able to define, organize, and implement a large software development using TSPi.
2. Experience the dynamics of team software development in an environment much like a real word or industrial software organization. Be able to understand and select appropriate mechanisms for team communications to ensure team goals are achieved.

Comp 474 Software Engineering Syllabus, version 4
Course Grading
Your grade will consist of three components with relative weights as follows (I reserve the right to adjust the percentages in your favor if circumstances warrant):

Team Software Project (40%)
All members of each team will receive the same grade based upon the quality and completeness of the software implementation, the team’s use of and adherence to software engineering process, the effectiveness of the team’s reports and demonstrations, and the success in becoming an effective, well functioning, and supportive team. This grade component will be based on your team project notebook, project documents, team presentations, and the completeness and quality of your team software product and other evidence of your team’s effectiveness.

Personal Participation (20%)
All students are expected to attend all class sessions and all team meetings in person. The participation grade will be based upon attendance and contributions to discussions in class, attendance and full, enthusiastic involvement in the assigned team, and scores in classroom quizzes and other individual course assignments.

Examinations (40%)
The midterm and final examination scores. The final exam will be a larger part of the grade than the midterm.

The course will be graded on a curve based on performance of all students. If undergraduate students are participating in the class the curves for graduate and undergraduate student grades may be different.

Participation and contributions in class and team meetings are an important part of learning software engineering. All students are expected to attend all classes and all team meetings in person. Attendance and valuable participation in class discussions are a factor in your course grade. Attendance and full participation in your team are a factor in your course grade. Attendance will be taken periodically in class. Attendance for each team meeting will be recorded. Providing false attendance information in either class or team meetings is a serious offense. See the section on Academic Honesty.

If for any reason you do miss a class session, it is your responsibility to determine what you missed, locate any handouts, determine any changes in assignments, course plans, or schedules, etc. It is not my obligation to help you make up for missing class. A key part of the class will be your participation and understanding of class room work. You will be at a great disadvantage if you miss a class. No amount of studying the textbook and other materials can provide all the knowledge you’ll need for the examinations and assignments.
A short quiz may be given in class at any time without advanced notice; this approach serves to encourage you to keep up-to-date on reading assignments. There will be no “make-up” quizzes unless you have made a prior arrangement to be excused from class that day.

Please do not ask for “extra credit” to improve your grade as this is neither practical in the course nor fair to your fellow students. I will be happy to discuss your performance in the course with you at any time, including discussing possible grade based on performance to date and ways to improve your performance during the remainder of the course.

Grades on each assignment are posted in Blackboard. Blackboard also allows you to see your current total and an estimate of your current total percentage. Points on assignments and grading comments are to be reviewed promptly by the student. To encourage your use of the grading comments to improve learning, requests for further explanation or adjustments to your score must be made no later than two weeks after the points are posted; after that time the grading is final and cannot be changed.

**Timely Completion**

Timely completion will be stressed as a key part of learning software engineering process and discipline. All personal and team assignments, reading, software, reports, metrics, etc., are expected to be completed on time. Your team will be depending on you to do what you say you will do and to do it on time with high quality. Repeated failure to meet your commitments to your team will result in dismissal from the team and you will get a grade of 0 for your team project.

Late work will not be accepted. Exceptions only by prior approval and will rarely be given. *Time is of the essence of this course*

You are welcome to ask questions on all assignments and course work, seek additional information on the assignments, and offer observations on the assignments to me either in or outside of class. To discourage procrastination, no questions on the assignment will be answered immediately before the assignment is due – please plan your work ahead and do not wait for the last minute to begin work!

**Classroom Behavior**

As a sign of respect for your fellow students and professor, mobile phones, pagers, etc. are to be turned off during class (exceptions for situations when you need to be reached for truly urgent and critical matters of an emergency nature). Exchange of text messages and mobile short messages (SMS) is also not allowed during class time; during examinations it will be automatically considered an instance of academic dishonesty.

Side conversations and interactions with other students during class lectures are rude and you will miss something important.

We all have days when we are tired and cannot keep our eyes open. It this happens to you during class feel free to stand up at the rear of the class room instead of looking bad sleeping at your seat.

Comp 474 Software Engineering Syllabus, version 4
Office Hours, Discussion Board, and Help
See my web page at http://cs.luc.edu/whonig/. There will be both traditional office hours and online interactions with me and other students in the discussion boards. If these times do not work for you, please let me know so adjustments can be made. Additional times are available by appointment.

There will be organized discussion sessions to encourage exchange of ideas and for interactions between students. These discussions will use Blackboard. Participation in the discussion boards for the class will count as part of your class participation grade. The purpose of these boards is for students to exchange questions and ideas. Help other students and share your insights. Asking things like “What’s the answer to homework number 3?” is not allowed (obviously). However, you are encouraged to ask for help on details and things that stump you. For example, “Anyone know how to find the average of a variable length list of numbers? I need to do that as part of my programming project 4”. Students are expected to both post questions and answer those of others in the discussion board. As a guide, you should be posting valuable new items and answering others helpfully at least three to four times for each class session.

Continuous Improvement
I believe in a personal quality process of continuous improvement. Anything can be improved by applying the quality process of “Plan, Do, Check, Act” (PDCA). To improve the course and the learning of these computer science concepts, I welcome your feedback, comments, suggestions, and complaints at any time.

In support of this PDCA process, I may ask you to participate in surveys during the course. These surveys will measure student impressions of the course; when time permits I will share the results with the class. Your inputs on these surveys are anonymous and in no way affect your grade.

Version 4 (Adding Sakai, online options, Jan 2014)
Version 3 (Format changes, text information updated, grading updated, special requirements updated: Aug 2012)
Version 2 (text changes; grading and late assignment updates: 26 Aug 2008)
Version 1 (original: 15 Jan 2007)