

SE 591 Software Engineering Studio

Instructor: Dr. Jane Cleland-Huang

Term: Spring 2014

Course homepage: <https://d2l.depaul.edu/>

Office Hours: Wednesday 1.00pm-2.30pm

Office Location: CDM 836

Office Phone: 312-362-8863

Email: jhuang@cs.depaul.edu

Skype: Available upon request

Summary of Course

This course provides students with the opportunity to work within a small team on a software development project. Students are responsible for completing all life cycle phases of the project (inception, elaboration, construction, and transition) and executing the relevant development disciplines across all project iterations:

- Requirements
- Domain research and modeling
- Architecture and design
- Implementation
- Testing
- Deployment
- Configuration & change management
- Project management
- Management of the development environment

Course Goals

The broad objectives of this course are to:

- Provide students with experience in an end-to-end software development project;
- Have students work as part of a cooperating team;
- Have students produce substantive intermediate deliverables within realistic time and resource constraints; and
- Have students produce a functional final product.

A specific intent of the course is to allow students to draw upon the various skills, techniques, and experiences gained in other courses and apply them to a complex software development project. These include:

- Use Java and Java-related technologies;
- Employ advanced Java programming techniques;
- Learn/use other programming languages and technologies such as C#, C++.
- Use object-oriented analysis and design skills and create artifacts in the Unified Modeling Language (UML);
- Use a development methodology that combines aspects of the macro process structure of the Unified Process with the scrum development framework at the micro process level;
- Apply project management skills and techniques; and
- Learn new tools and technologies in order to satisfy specific requirements for the project.

Since the Studio sequence acts as the capstone course for the Software Engineering curriculum, the course also provides the chance for students to acquire non-technical knowledge and skills essential to the software engineering profession, including:

- Technical writing skills; and
- Presentation and oral communication skills.

Choosing a Project

As the SE491/591 sequence serves as the Software Engineering capstone, it provides the opportunity for students to apply all of the skills and technologies they have learned throughout the MS degree in a holistic way. For this reason many of the project-level decisions will be made at the team level i.e. choosing a project and choosing a SDLC process. However, each project is expected to be challenging, to have scope for a two-quarter sequence, and to provide sufficient work for 4-5 team members. While team members are expected to work together and support each other, it is also expected that individual team members will have specific tasks that they will be responsible to deliver **on-time** to the team. Regular team meetings with the instructor will provide visibility into each team-member's contribution to the project.

During the first week of the quarter, the instructor will present several project ideas; however it is not necessary for any of the teams to take up these projects. All students are expected to think about their own ideas for a project and to bring these to class for week 2. Project selection will be a collaborative activity between students and the instructor.

Course Logistics

This instance of SE Studio includes both an in-class and online section. Students will continue working on the project teams formed during SE491. Any new students will be assigned to an existing team.

The instructor will meet with each team for 30-40 minutes each week during the class time. Classroom students will be assigned meeting spaces in research labs in the CDM building. Online students will be expected to make themselves available during Wednesday night class time for a 30-40 minute online meeting with the instructor and for additional time to coordinate with their team mates. **Online students are strongly encouraged to make arrangements to be available during this time each week.** Students unable to do so should contact the instructor during the first week of class.

Every 3-4 weeks each team is expected to make a formal presentation to the entire class. It is expected that during each quarter, each team member will make some contribution to this presentation. Online teams are expected to make their presentations in realtime using online collaborative tools that will be provided/described by the instructor. The tentative schedule will therefore be:

Weeks 1,2,3 (April 2nd, 9th, 16th)

No lecture.

Meet in team groups. The instructor will join each group meeting for 20-30 minutes according to the previously defined schedule.

Week 4 (April 23rd)

Team presentations to the entire class. Presentations will focus on: (1) **Field tests** and your findings, (2) Action items that you derived from your field tests. This presentation is therefore all about the **usability** of your app. Does it really support your users in the way you anticipated? Why or why or not? What improvements do you need to make?

Meeting time: 6.30pm-8.15pm. Each team will have 15 minutes presentation + 5 minutes switch-over time. Please make sure your presentation fits into 15 minutes.

Deliverable: A short report that describes your field test, results, and planned improvements.

The report and presentation are worth **20% of the final grade.**

Week 5 (April 30th)

Kanban week.

6.00-7.00pm Hands-on Kanban activity. This will be **recorded** for online students. We'll be transforming our projects from traditional sprints to Kanban.

7.00pm-9.00pm Group time.

Transformation to Kanban for the final iteration is worth 5% of the final grade.

(Note – this transformation is not difficult – just a small change in perspective)

Weeks 6-8 (May 7th, 14th, 21st)

No lecture.

Meet in team groups. The instructor will join each group meeting for 20-30 minutes according to previously defined schedule.

Week 9 (May 28th)

External Panel presentation in the boardroom (hopefully). We will have 3 external panelists representing Software Development, Business, and User Interaction Design. Each team will give a 15 minute presentation followed by 10-15 minute Q&A from the external panelists.

Presentation is worth 20% of the final grade.

Week 10 (June 4th)

Teams will work independently to finish up their projects.

Finals Week

Group deliverables will include: **30% of final grade**

1. A SHORT you-tube (style) video of your App.
2. Completed project report (builds on partial deliverables from SE491)
3. Complete source code for the app. (Please note: This source code belongs to each team. I request the source code purely for evaluation purposes; however each team retains full ownership of their own code. The code will be treated confidentially for grading purposes only).

Individual deliverables will include: **10% of final grade**

1. A journal documenting individual contributions. Note: As in SE491 you may substitute this polished journal with informal weekly reports describing your weekly contribution.

Major Assessment 5% of final grade

As part of a university-wide effort, we are required to assess all Software Engineering students (and in fact all students taking SE591). To that end, there will be a take-home assessment to be completed in weeks 8-10 of the quarter. The assessment is worth 5% of your total grade. It will include a series of short answer questions that evaluate your Software Engineering knowledge. (More details later)

Individual Ethics Presentations 10% of final grade

During the quarter each student is required to create one 5-10 minute recording that addresses an ethics issue. During the first couple of weeks of the quarter we will select 3 different issues of interest to us. I will initiate this through an online recording and then we'll select through online voting. Once we've selected the three issues – each person can select any of the issues they wish to argue either FOR or AGAINST. (Note: You can argue either side – regardless of what you really believe). One idea I have is to find issues that are particularly relevant to Requirements Engineering and then to wind this discussion into my bi-monthly IEEE Software Column. I thought that might be fun, and I'll ack everyone who contributes.

(I'll post an example of my column under documents for you to see). Alternately we will look at broader Software Engineering ethical questions. (tbd)

This activity will be due by May 14th i.e. Week 7.

Additional Logistics

- All teams are expected to use the Perforce version control system. A presentation on its use will be provided in Week 3 and accounts will be set up for each team.
- Virtual machines will be made available to students for development purposes if requested. Students are responsible for installing any **legally available** software needed in their work. It is against the university policy to install illegal copies of software, and doing so will result in a significant grade reduction for the perpetrator.
- Teams may choose their own collaborative software in conjunction with the instructor (i.e. especially in the case of online teams – this software will be used for weekly team meetings). Tools will be discussed during the first two weeks of class.
- The course uses Desire2Learn (<https://courses.depaul.edu/>) as its instructional management system. All course facilities (such as communication forums) and content (such as presentations and reference papers) are available on Desire2Learn (D2L).

Grades

As specified above.

Please note that 25% of the grade is individual in nature, and there are no guarantees that all team members will achieve the same grade for the project. At the same time, the instructor recognizes that different students will bring different strengths and experiences to the project, and each student should make contributions commensurate with their own skill set. The team grade assumes that the delivered solution executes correctly. 10% will automatically be deducted from the team grade if the software fails to execute. Final execution may be demonstrated in one of three ways (1) during the final presentation in week 10, (2) by team members during finals week, (3) by sending the instructor instructions to execute (but this is the riskiest and least approach).

Grades will be assigned as follows:

If the final numeric grade is less than:	and greater than or equal to:	the final letter grade is:
-	94	A
94	90	A-
90	87	B+
87	83	B
83	80	B-
80	77	C+

If the final numeric grade is less than:	and greater than or equal to:	the final letter grade is:
77	73	C
73	70	C-
70	67	D+
67	63	D
63	60	D-
60	-	F

Time Budgeting

Students should allow additional time for this course as compared to conventional lecture/homework-style courses. A general guideline for the latter type of course is approximately 3-4 hours of work outside of class for each scheduled hour of class; this works out to 10-12 hours for most students. Studio requires additional time investment beyond actual coursework in the form of team meetings, activity coordination and reporting, ramping up knowledge on new domain concepts and/or technologies, etc. In addition, the workload for an individual may vary widely from one week to the next due to deliverable schedules. Please plan your schedule accordingly.

Academic Integrity

Students should note that this is a zero-tolerance course with regard to academic integrity violations. All students are expected to read and fully comply with DePaul's Academic Integrity Policy, the text of which is available at: <http://academicintegrity.depaul.edu/>

A discussion of academic integrity is presented during the first week of class. If you have any questions about what constitutes an academic integrity violation or what its consequences might be, please be sure to have these questions answered before your first coursework submission.

Any reuse of code and/or components **must be clearly specified in the presentations and/or final documents**. Reuse is allowed in this course as long as the reused code is publicly available (i.e. open sourced or otherwise used with permission). However, it should be noted that if code is reused from other sources, then it is expected that the scope of the project will be far more extensive than if code is programmed from scratch. Any questions about scope of the project or code re-use can be discussed with the instructor during the weekly meetings.

Textbooks and printed resources

There are no required textbooks for this course. Reference and reading materials are provided by the instructor as required.

Prerequisites

SE 450 is the minimum prerequisite for this course. In addition, completion of SE 430 and at least two other software engineering

Other Course Policies

Student Support. Support for students is provided through online office hours dedicated to the course and through online question-and-answer discussion forums on D2L or other collaboration tool such as Skype or Wimba. Online discussion forums are available for posting general coursework questions and comments. The instructor monitors these forums and makes every effort to respond to postings within 24-36 hours. However, due to schedule issues, it may take longer to receive an instructor response. Email should be used only for personal issues, sensitive topics, or for student-specific coursework questions. Make all questions clear and specific.

Communication. All correspondence and communication, such as email and phone messages, must include your full name and course number. The subject line must include SE491.

Exceptional Circumstances. Every effort is made to accommodate students who encounter exceptional personal circumstances during the quarter. Students who experience unanticipated personal, work, health, or family emergencies should notify the instructor by email or phone as soon as possible with a brief explanation of the

circumstances and any anticipated impact these might have on coursework. Students who have anticipated exceptional circumstances such as secular or religious holiday observances, medical treatment, or work-mandated travel should notify the instructor as early as possible of these circumstances and any anticipated impact these might have on coursework. In both unanticipated and anticipated cases, a suitable plan for dealing with the coursework impact is agreed upon by the student and instructor. In some cases, suitable documentation of the exceptional circumstances may be requested by the instructor. In both unanticipated and anticipated cases, students must inform their fellow team members of their inability to participate in the project and make suitable arrangements to make up for missed contributions to the project.

Grade Responsibility. Every effort is made to provide the student with the resources and support needed to succeed in the course. Grades are assigned fairly and impartially based on the coursework submitted by the student, without regard to external circumstances such as GPA goals or employer tuition reimbursement minimum grade requirements. It is the student's responsibility to earn his or her final grade. Please do not ask for a grade which you do not earn.