

**University of Wisconsin Oshkosh  
Computer Science Department**

**CS 342: Software Engineering II  
Spring 2009**

----- Course Syllabus -----

**COURSE SCHEDULE**

		<b>Days</b>	<b>Time</b>	<b>Location</b>
<b>Section 001</b>	Project Meeting	T	3:00-4:00pm	HS
	Project Work or Enrichment Seminar	T	4:00-5:00pm	
		TH	3:00-5:00pm	HS

**INSTRUCTOR** Dr. Kathy (Kate) S. Faggiani  
Computer Science Department  
EMAIL [faggianik@uwosh.edu](mailto:faggianik@uwosh.edu) PHONE (920) 424-2069  
IN-PERSON OFFICE HOURS\*(HS 218): M 11:25am-1:45pm  
TTH 12:00-1:20pm  
Or use Google CHAT: [faggianik@gmail.com](mailto:faggianik@gmail.com) whenever I'm online.  
\*Also available other times by appointment

**PREREQ** Computer Science 271 and 341, each with a grade of C or better.

**COURSE DESCRIPTION**

Software Engineering is the second of a two-semester sequence on the topic of modern Software Engineering tools and techniques. Topics covered include Design Patterns, the Unified Modeling Language (UML), Component-based Software Development, Advanced OO Design and Analysis, Refactoring, and other techniques such as Extreme Programming. An extensive software development project will allow for practical application of the discussed techniques. (Source: [http://www.uwosh.edu/computer\\_science/cs-major/courses/course-descriptions](http://www.uwosh.edu/computer_science/cs-major/courses/course-descriptions) retrieved on 01/27/2009).

**COURSE OVERVIEW**

The conduct of this course is based on the practice of scenario-based learning. Scenario-based learning is based on the theory of situated cognition, which contends that learning occurs as a result of "authentic activities that are common to the community of practice in which the learner is involved". In software engineering, the "community of practice" is the actual conduct of software development projects. Thus, the traditional "classroom" learning environment in Software Engineering II is replaced by a scenario which places the student in a software development effort. The instructor will take on the role of "Project Coordinator"

responsible for guiding process and facilitating meetings. The scenario which forms the basis for the 14 week learning experience is briefly described below.

## THE SCENARIO

The activities and interactions that you pursue in the course throughout the next 14 weeks will be consistent with the following scenario:

Assume you have graduated from the UWO Computer Science program with an emphasis in software engineering. Shortly after graduation, you accepted a position with a software engineering consulting company that specializes in custom software development for third party organizations.

During the hiring process, your new employer made it clear that although you were primarily trained in Java programming with a smattering of C++, the platforms on which you developed would be determined by the needs of the clients that engaged your firm in custom development efforts. Furthermore, you would be expected to research and quickly gain proficiency in any platform/language/tool required under the terms of the project that were specified by the firm's clients.

After negotiating a REALLY high salary and agreeing on a start date for your new job, you took a few weeks off to relax between graduation and the start of your new career. On Tuesday, February 2, 2009, you will begin your new job. You have received an email explaining that in addition to the usual first-day-at-work activities, your work day will include a new project briefing meeting at 3pm. The meeting will be conducted by the project coordinator assigned to handle the administrative tasks related to your project and facilitate interactions with your new client. The purpose of the meeting will be to describe, in general terms, the background of the project and provide you with resources to prepare you for a project work meeting on Thursday, February 4.

In addition to your initial project assignment, as a new software engineer you will be responsible for participation and attendance in a series of "enrichment seminars" suitable for developing the skills, abilities, attitudes, and knowledge of a new college hire. On occasion, you may be asked to prepare and deliver the content of a seminar, at other times the seminar content may be delivered by invited guests or the project coordinator.

Please Note: There are no examinations or individually graded assignments in this course. See the section below on "Assessment of Learning Outcomes" for individual student grade determination.

## MATERIALS AND RESOURCES

**REQUIRED TEXTBOOK** Subramaniam, V. & Hunt, A. (2006). Practices of an Agile Developer. Pragmatic Bookshelf. Available at UWO Bookstore or online via O’Reilly Books and others.

**OTHER RESOURCES** Project Repository – will hold all documents/files/work products created as a part of the project by the project coordinator. This will be established as a Google Group in which membership is mandatory. ALL project work product must be shared, stored, and archived via this group.

Personal Blog – each class participant is required to create a personal blog using any free hosting service available. Popular sites include: blogger.com, wordpress.com, aeonity.com, tBlog.com, blogeasy.com, or any other public blog site. After creation of the blog site, you must email the URL to the project coordinator at [faggianik@gmail.com](mailto:faggianik@gmail.com). You will be encouraged to blog about your experience with the project and scenario-based learning. In addition, you will receive weekly or semi-weekly blog requests to address specific topics. DO feel free to blog your thoughts ideas, and feelings related to your learning and the project at ANY TIME. This should be a blog based on your personal experience with Software Engineering II.

Tools/Technologies/Learning Aids – new items will be added on a regular basis to the project repository. You will be responsible for alternately viewing and submitting materials as they are identified.

## LEARNING OUTCOMES AND ASSESSMENT OF STUDENT LEARNING

Learning outcomes represent the knowledge, skills and abilities, and attitudes/behaviors you will acquire as a result of successful participation in this course. The learning objectives for Software Engineering II are listed below. Please note that each learner’s progress will be assessed at regular intervals using observation and review of work product.

**Table 1. CS 342 Learning Outcomes and Assessment**

<b>Course Outcome</b>	<b>Assessment Activity</b>
Apply agile development processes and techniques to a significant software development project.	Observation of project steps/process
Develop a work breakdown structure and task set for a given project/project phase.	Observation of work product produced in team meetings
Use appropriate techniques to identify and refine requirements for a significant software development project.	Review of work product posted in project repository

<b>Course Outcome</b>	<b>Assessment Activity</b>
Apply a variety of software engineering methods, techniques, and tools to the development of a significant software project.	Observation of project tasks/work product produced
Work effectively as a member of a development team.	Observation in team meetings, review of blog comments, review of work product.
Establish an atmosphere of trust and mutual commitment with co-developers.	Observation in team meetings
Establish an atmosphere of trust and mutual respect with user representatives.	Observation of interaction with users – in-person, online discussion
Conduct effective technical walk-through and code review sessions with peers.	Observation in meetings
Establish effective communication plans with clients and co-developers.	Review of work plans; monitoring of communication issues.
Create and contribute work product to project repository as assigned.	Comparison of work product produced to commitments made in work breakdown/work schedule
Document personal project experience, including key project elements and learning moments, for post-mortem project review.	Review of personal blog
Create and fulfill commitments to project tasks.	Comparison of results produced to commitment
Distinguish among different software testing strategies, and identify an appropriate software testing strategy for a given project.	Activity in enrichment seminar
Seek timely assistance from others when individual efforts to resolve problems have been unsuccessful.	Observation and review of personal blog
Ability to design and create software that meets the defined needs of users.	Review of project results by client

The learning outcomes described in Table 1 relate directly to the Computer Science Program Objectives found at: [http://www.uwosh.edu/computer\\_science/cs-major/program-objectives](http://www.uwosh.edu/computer_science/cs-major/program-objectives)

### **ASSESSMENT OF LEARNING OUTCOMES**

To determine student progress toward achievement of learning outcomes in the course, direct observation of student performance in project meetings, contribution in enrichment seminars, description of experiences from personal blogs, and content of presentations will be reviewed. Students will receive continuous and on-going feedback (both written and oral) on their performance

to encourage them to move toward the desired learning outcomes. These assessments are primarily formative and designed to guide student learning. Summative assessment will be conducted by user representatives and other third parties who will be enlisted to provide feedback on the final project deliverable. In general, grades will be assigned based on the following:

**A student who earns an A in this course:**

- Attends all project meetings and enrichment seminars, and consistently makes worthwhile contributions.
- Is prepared to participate in every meeting.
- Takes the initiative to identify risks and problems in the project, suggest mitigation activities, and clearly communicates these to the team.
- Posts regularly, both assigned and extemporaneously, to his/her personal blog, ideas and musings that demonstrate synthesis of software engineering methods and techniques with the course project; postings reflect actual experience without judgment (i.e. that the post will NOT be construed as a “positive” or “negative” post)
- Actively participates in all project meetings.
- Takes the initiative to take responsibility for project tasks.
- Commits to producing assigned results by a given date.
- Consistently produces high quality work that is posted to the project repository before the commitment date.
- Demonstrates consistent use of appropriate software development methods and techniques.
- Encourages and supports participation in the project by all team members, and treats all team members with respect.
- Maintains clear and respectful communication with all user representatives.
- Suggests and supports the use of appropriate software engineering techniques.
- Produces final project result that meets all requirements and is of high quality – user/third party rate the project results as “A”

**A student who earns a B in this course:**

- Attends all project meetings and enrichment seminars, and occasionally makes worthwhile contributions.
- Is prepared to participate in most meetings.
- Takes the initiative to identify risks and problems in the project, and communicates these to the team.
- Posts regularly to his/her personal blog ideas and musings that demonstrate familiarity with software engineering methods and techniques.
- Actively participates in most meetings.
- Takes responsibility for assigned project tasks.
- Commits to producing most of assigned results by a given date.
- Consistently produces an acceptable quality of work that is posted to the project repository by the commitment date.

- Demonstrates use of appropriate software development methods and techniques.
- Treats team members with respect.
- Communicates effectively as needed with user representatives.
- Supports the use of appropriate software techniques as determined by the team.
- Produces a final project result that is rated “B” by users/third party

**A student who earns a C in this course:**

- Attends all project meetings and enrichment seminars, but makes little contribution.
- Is occasionally prepared to participate in meetings.
- Approaches risks and problems in the project passively, willing to let team members address them.
- Posts to personal blog occasionally with general ideas/musings that demonstrate some degree of insight or synthesis of software engineering methods/techniques with project.
- Passively participates in project meetings
- Occasionally shows initiative and takes responsibility for selective tasks.
- Occasionally commits to producing assigned results by a given date.
- Produces work of somewhat acceptable quality or is occasionally incomplete; may miss due dates or does not post work to project repository.
- Demonstrates marginal use of software techniques and somewhat complies with those determined by the team.
- Treats team members with indifference.
- Has minimal communication with user representatives.
- Avoids use of appropriate software techniques agreed to by the rest of the team.
- Produces final project result that is rated “C” by users/third party.

Students not “fitting” into one of the above three grade categories (or “crossing” categories, i.e. AB or BC) will receive an F in the course.

**COURSE CALENDAR**

A course calendar is superfluous in scenario-based project course. The project schedule will be developed by the project team and posted in the project repository, as will task commitments and due dates, enrichment seminar requirements, and other key activities or events in the course.