

Master Course Syllabus
School of Engineering and Computer Science
Washington State University Vancouver

CS 420
Software Engineering in Practice
3 Semester Hours
(3 lecture hours)

Catalog Description

Developing software in a team environment; project management; unit and integration testing, bug tracking, configuration management, software process models; object-oriented design with UML

Prerequisite Courses

CS 320 – Fundamentals of Software Engineering

Prerequisite Topics

- Experience with an object-oriented programming language (e.g., Java or C++)
- Experience with a software design tool (e.g., Rational Rose, Poseidon, Visio)
- Technical Writing
- Experience with the software development process (requirements analysis, specification, design, implementation, testing)
- Use of UNIX or Windows environment for coding, compilation, debugging and testing

Measured Course Outcomes

Students taking this course will:

1. Peer review teammates based on their ability to contribute their time, ideas, and software engineering skills while working on a team project. *(Contributes to performance criteria **D-1** and **D-2**).*
2. Carry out a design and development process in a team environment to satisfy project requirements for a computer program or system. *(Contributes to performance criteria **C-2** and **D-3**).*
3. Design and execute a test plan to verify a design and validate project requirements. *(Contributes to performance criteria **C-3**).*
4. Deliver at least one individual oral presentation about their role in the team based project. *(Contributes to performance criteria **G-2**).*

Required Textbooks

No Required Text book.

Reference Material

UML Distilled (Third Edition), Martin Fowler, Addison Wesley Inc.

Software Engineering, A Practitioner's Approach, Roger Pressman, Mc Graw Hill, 6th Edition

Major Topics Covered in the Course

1. Students will work on a large software project that requires the coordinated efforts of a team to be successful.
2. Software Engineering in a team environment will include the following topics:
 - a. Team management
 - b. Project planning
 - c. Version control
 - d. Integration testing
 - e. Software maintenance
3. Students will experience, as a team, each phase in a software process model that must include:
 - a. Software requirements and specification including formal analysis
 - b. Object oriented design and programming
 - c. Testing, debugging, and maintenance

Laboratory Projects

	Project Area	Weeks
	Team-based project planning	1
	Team-based software design	2
	Team-based programming	4
	Team-based testing	2

CSAB Category Content

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	0	0	Computer Organization and Architecture	0	0
Algorithm	<u>0</u>	<u>3</u>	Concepts of	<u>0</u>	<u>0</u>

&
Software
Design

Programming
Languages

Oral and Written Communications

Students are required to deliver at least one oral presentation concerning their roles and experiences as a team member working on a software development project. Students are required to create a software requirements specification document (SRS).

Social and Ethical Issues

We discuss ethical issues surrounding team work.

Theoretical Content

	Topic	Hours
Software Testing		8
Formal Analysis		5

Problem Analysis

All student programming assignments require the student to analyze software requirements. Students are required to use at least one formal method to validate their design model. The instructor analyzes problem solutions (both their own and the student's) in class.

Solution Design

This course introduces students to team based software engineering with emphasis on requirements analysis, specification, object-oriented design, implementation and testing (unit and integration). Students, as a team, incrementally develop a software product to give them hands-on experience with software engineering principles. The objective is to experience particular methods in each phase of the software development lifecycle.

CC2001

This course provides coverage of topics in the following areas (hours listed are minimums):

SE1. Software Design [core]	2
SE2. Using APIs [core]	1
SE3. Software Tools and Environment [core]	3

SE4. Software Processes [core]	1
SE5. Software Requirements and Specification [core]	2
SE6. Software Validation [core]	4
SE7. Software Evolution [elective]	3
SE8. Software Project Management [elective]	4

Course Coordinator: Orest Pilskalns
Last Updated: August 31, 2005 (Approved)
Syllabus Version Number: 1.2