OMSE 517: Agile Software Development

Spring 2010 Syllabus

Instructors:	*Chris Gilmore* grimjack@cs.pdx.edu
Office Hours:	*By appointment only*
Course Materials:	*Via PSU Online (Blackboard)*

About This Course

This course has been designed for graduate level software engineering students who are interested in learning and applying the fundamentals of the Agile Software Development process and the various ways Agile development is implemented in the real world. The course explores Agile concepts both in theory and practice. With the approval of an OMSE advisor, OMSE students may register in this course and count it for credit as an OMSE elective. Students are introduced to the principles and foundations of Agile Development, XP (Extreme Programming) and the SCRUM methodology including terminology, principles, and the Agile Software Development lifecycle. The course explores each of the Agile lifecycle phases in depth including various approaches whereby students can apply Agile concepts and principles. The course also introduces students to day-to-day life on an Agile team as a short iteration is planned and executed.

Course Prerequisites

OMSE 500 is recommended - a basic understanding of the traditional software development models (waterfall, spiral, iterative, evolutionary, incremental, etc.). No previous agile or XP competency or experience is required.

Learning Objectives

Upon successful completion of this course students will:

- Understand and be able to describe the terminology and history of Agile, XP and Scrum
- Be able to describe the Agile lifecycle and how it relates to traditional software processes
- Understand and be able to explain how to conduct test-driven development
- Understand what “Done Done” means and explain how it applies to Agile development
- Be able to describe the relevance and conduct of an iteration demo and the retrospective.

Textbook / Readings

Shore & Warden, The Art of Agile Development, O’Reilly Press, ISBN 0-596-52767-5
Supplemental materials will be posted as required in PSU Online.

General Course Information and Policies

Class Participation: Participation is important. Borderline grade decisions will be influenced by the degree of consistent and substantive participation in class discussions (online and/or classroom).

Online Discussions: Students will conduct online discussions to address most of the learning objectives. Depending on the number of students enrolled, the class may be divided into discussion groups / teams. Instructions/guidance for conducting online discussions will be posted in the course shell (PSU Online / Blackboard).

Academic Integrity Policy: Cheating of any kind will not be tolerated. While it is encouraged for students to study together, the student must ensure that the work they turn in for individual assignments is entirely their own.

Late assignments will not be accepted without providing acceptable rationale for lateness and obtaining the instructor’s approval before the assignment due date.

Grading / Evaluation

Evaluation and grades will be broken down as follows:

Type of assignment	Percentage
Participation in weekly online discussions	10%
Participation and contribution to Scrum meetings (in class and/or online)	20%
Midterm exam or take home assignment	35%
Final exam or take home assignment	35%

Weekly Lesson Plan (readings and assignments)

Wk	Topics Covered	Readings Assignments
1	Introduction: Syllabus and Objectives Intro to Agile & XP: History, terminology, concepts**;** Prepare for the iteration planning meeting	Chapters 1 & 2 http://en.wikipedia.org/wiki/Scrum_(development)
2	Iteration planning meeting: General discussion; Divide task into stories; Determine tasks; Closing	None this week
3	More about Agile: Lifecycle; Is agile always right? The team and other concepts	Chapters 3 & 4
4	Thinking in Agile: Pair programming; Work environment; Root cause analysis; Retrospectives	Chapter 5
5	Collaborating: Trust; Sit together; Real Customer Involvement; Ubiquitous Language; Stand-Up Meetings; Coding Standards; Iteration Demo; Reporting	Chapter 6
6	Releasing: What is "Done Done"?; No Bugs; Version Control ; Ten-Minute Build; Continuous Integration; Collective Code Ownership; Documentation	Chapter 7 Exam #1 handed out
7	Planning: Vision; Release Planning; The Planning Game; Risk Management; Iteration Planning; Slack; Stories Estimating	Chapter 8 Exam #1 due
8	Developing (not just coding anymore): Incremental Requirements; Customer Tests; Test-Driven Development; Refactoring; Simple Design; Incremental Design and Architecture; Spike Solutions; Performance Optimization; Exploratory Testing	Chapter 9
9	Going on from here: What to do next? How to get better at Agile Development	Chapters 10, 11, 12, 13
10	Demo and retrospective: Closing thoughts	Chapters 14 & 15 Exam #2 handed out, due finals week (11)