Stevens Institute of Technology
SSW 555 Syllabus

SSW 555 Agile Methods for Software Development

Overview

In software problem areas that require exploratory development efforts, those with complex requirements and high levels of change, agile software development practices are highly effective when deployed in a collaborative, people-centered organizational culture. This course examines agile methods, including Extreme Programming (XP), Scrum, Lean, Crystal, Dynamic Systems Development Method and Feature-Driven Development to understand how rapid realization of software occurs most effectively. The ability of agile development teams to rapidly develop high quality, customer-valued software is examined and contrasted with teams following more traditional methodologies that emphasize planning and documentation. Students will learn agile development principles and techniques covering the entire software development process from problem conception through development, testing and deployment, and will be able to effectively participate in and manage agile software developments as a result of their successfully completing this course. Case studies and software development projects are used throughout.

Prerequisites

Programming experience in an object-oriented language, preferably Java.

Cross-listed with: *CS 555*

Learning Goals

After taking this course, the student will be able to:

- describe several agile methods for software development
- describe the origins and motivations of the Agile Manifesto
- practice pair programming
- perform test-first development with a unit testing framework such as JUnit
- create and critique user stories for system requirements
- refactor code and tests to meet changing needs
- develop and monitor project backlogs
- measure and monitor velocity of development
- compare and contrast plan-driven versus agile methods
- construct tailored agile processes that best fit the technical and market demands of a modern software project

Pedagogy

The course will employ lectures, quizzes, homework and team projects.

Required Readings

There is no textbook. Readings from journals and web sources will be assigned for each week.

Assignments

1. Weekly Reports - Each student needs to keep track of their time and report their status at the end of each week.
2. Quizzes – There will be a quiz on the reading and lecture notes due at the end of each week.
3. Homework – Homework must be completed by the required date and submitted via the course website.
4. Project – Students will work on teams on a semester-long project with regular deliverables.

The assignments and their weights are as shown below:

Weekly Reports	5
Quizzes	10
Homework	25
Project	35
Final Exam	25
TOTAL	100

Final grades will be determined by the following scale:

Grade	Score
A	93 - 100
A-	90 - 92
B+	87 - 89
B	83 - 86
B-	80 - 82
C+	77 - 79
C	73 - 76
C-	70 - 72
F	< 70

Course Schedule (Note: subject to change! The course schedule will be maintained on the course website)

Wk	Topic	Reading(s)	Homework	Project
0	Orientation			P0: experience, preferences
1	Introduction	[Boehm2002], [RUP 2003]	H1: Agile Manifesto	P1: create GEDCOM file
2	User Stories	[Ambler 2004], [Grenning 2002]	H2: User Stories	P2: write program to read GEDCOM file
3	Scrum	[Schwaber 2013]	H3: Scrum	P3: modify program to store data
4	Testing	[Gamma 1998]	H4: Test-first tool	P4: Sprint 1
5	Pair Programming	[Williams 2001]	H5: Pair programming	
6	Refactoring	[Fowler 1999] 2-3	H6: Refactoring	P6: Sprint 2
7	Lean	[Poppendieck 2012]	H7: Lean	
8	Dynamic Systems Development (DSDM)	[Clifton 2003]	H8: DSDM	P8: Sprint 3
9	Feature-Driven Development (FDD)	[Palmer 2013]	H9: FDD	
10	Crystal	[Wikiversity on Crystal]	H10: Crystal	P10: Sprint 4
11	Scaled Agile Framework (SAF)	[Leffingwell 2009]	H11: SAF	
12	Disciplined Agile Delivery (DAD)	[Ambler 2013]	H12: DAD	P12: Sprint 5
13	DevOps	[Mueller 2011]		
14	FINAL EXAM			

References

[Ambler 2004] "Introduction to User Stories" by Scott Ambler, 2004, <http://www.agilemodeling.com/artifacts/userStory.htm>

[Ambler 2013] "Going Beyond Scrum: Disciplined Agile Delivery" by Scott Ambler, disciplinedagileconsortium.org/Resources/Documents/BeyondScrum.pdf

[Boehm 2002] "Get Ready for Agile Methods, with Care" by Barry Boehm, *IEEE Computer 35(1)*, 64-69, January 2002.

[Clifton 2003] "What is DSDM?" by Marc Clifton and J. Dunlap, www.codeproject.com/Articles/5097/What-Is-DSDM

[Fowler 1999] *Refactoring: Improving the Design of Existing Code* by Martin Fowler, Kent Beck, John Brant, William Opdyke and Don Roberts, Addison-Wesley, 1999, ISBN 0201485672. - available online at Stevens Library (only need to read chapters 2 and 3)

[Gamma 1998] "Test infected: Programmers love writing tests" by E. Gamma and K. Beck, *Java Report 3(7)*, 51-56, July 1998, <http://junit.sourceforge.net/doc/testinfected/testing.htm>

[Grenning 2002] "Planning poker of How to avoid analysis paralysis while release planning" by James Grenning, <http://renaissancesoftware.net/papers/14-papers/44-planing-poker.html>

[Leffingwell 2009] "The Big Picture of Enterprise Agility" by Dean Leffingwell, http://scalingsoftwareagilityblog.com/wp-content/uploads/2009/11/the-big-picture-of-enteprise-agilitywhitepaper.pdf

[Mueller 2011] "What is DevOps?" by Ernest Mueller, James Wickett, Karthik Gaekwad, and Peco Karayanev, http://theagileadmin.com/what-is-devops/

[Palmer 2013] "Feature Driven Development" by Stephen R. Palmer, <http://www.step-10.com/SoftwareProcess/FeatureDrivenDevelopment/index.html>

[Poppendieck 2012] "Lean Software Development: A Tutorial", *IEEE Software 29(5)*, 26-32, 2012.

[RUP 2003] "Rational Unified Process: Best practices for software development teams" by IBM staff, 2003, <http://www.ibm.com/developerworks/rational/library/253.html>

[Schwaber 2013] *Scrum Guide* by Ken Schwaber and Jeff Sutherland, www.scrum.org

[Wikiveristy on Crystal] "Crystal Methods" Wikiversity, en.wikiversity.org/wiki/Crystal_Methods

[Williams 2001] "In Support of Student Pair-Programming", *Proc. of 32nd ACM SIGCSE Technical Symposium on Computer Science Education*, 327-331, 2001.