

Syllabus: CSCI 430 Software Engineering

Kevin Buffardi

February 2015

Overview

In this class, we will explore software engineering practices and tools for you to develop as mature software developers. This class is project and team oriented, so you need to be active and collaborative in both the discussion meetings and the lab section.

Prerequisites: CSCI 311 for CSCI/CINS/APCG majors or EECE 337 for Engineering majors; ENGL 130 or JOUR 130 (or equivalent) all with a grade of C- or higher

This course is required for the Computer Science BS degree and is worth 3 credits.

Instructor

Kevin Buffardi, kbuffardi@csuchico.edu, Off hrs: OCNL 220 Tu/Th 1-3pm.

Required Materials

Laptop computer; No textbook required

Learning Outcomes

By completing this course, students will be able to:

Demonstrate knowledge of and first-hand experience with Agile Development and related methods; Contribute to a team software development project and

exhibit effective group collaboration and communication; Demonstrate proficiency with software engineering tools, particularly for software testing and revision control; Understand the fundamental software design and architectural patterns and apply them appropriately; Evaluate software designs for their qualities, including maintainability and extensibility; Reflect upon software engineering practices, evaluations, and experiences

The course addresses following broader CSCI Learning Outcomes:

Introduces, Practiced, Assessed

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline. **(P)**; b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution. **(A)**; c. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs. **(A)**; d. An ability to function effectively on teams to accomplish a common goal. **(A)**; e. An understanding of professional, ethical, legal, security and social issues and responsibilities. **(P)**; f. An ability to communicate effectively with a range of audiences. **(A)**; g. An ability to analyze the local and global impact of computing on individuals, organizations, and society. **(P)**; h. Recognition of the need for and an ability to engage in continuing professional development. **(P)**; i. An ability to use current techniques, skills, and tools necessary for computing practice. **(A)**; j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. **(A)**; k. An ability to apply design and development principles in the construction of software systems of varying complexity. **(A)**

Topics

Topics for this class include: agile development, software testing, revision control, software design patterns, anti-patterns, team management, communication