CST 316 Software Enterprise II: Construction & Transition, Spring 2016 (via ASUOnline)

Catalog Description

Best practices in software construction in the context of a team project, including refactoring, defensive programming, unit testing, and configuration and release management.

General Information

Instructor:	Dr. Kevin Gary (kgary@email.asu.edu, *http://www.public.asu.edu/~kgary)*
Class Meeting Time:	January 11, 2016 to March 1, 2016
Final Exam:	Available online February 29, 2016 and March 1, 2016
Office Hours:	MW 3:00-4:30pm, F 10:00-11:00am Arizona time
Schedule Line Number:	11577/12833
Class Website:	Blackboard
Skype:	drkevingary (if I am on you can ping me anytime)

Prerequisites

- CST315 (Software Enterprise I: Tools and Process)
- CST230, or permission of the instructor.
- A desire to learn and participate in class.

Course Description

Welcome (back) to the Software Enterprise! Students in this course will be exposed to software development best practices in the context of a scalable software project. Students learn these concepts through traditional lectures, inquiry-based learning, and in-class discussions based on assigned readings, and then apply these concepts in the context of a multi-semester project and problem-centered learning activities. The Software Enterprise focuses on the application of software processes and best practices to software development. This course focuses on constructing, building, testing, and deploying a software application in a project setting. Content will focus on best practices for implementing software, performing unit, integration, and scalability tests, and deploying software releases. Students in this course collaborate to construct software artifacts according to an agreed upon process.

Course Outcomes

Course Outcome	Description	Program Outcome
CO-1	Students will learn techniques for producing high-quality software code, source code documentation, and unit tests	Tech Competence / Engineering Practice
CO-2	Students will learn techniques and best practices for transitioning software products for release	Tech Competence / Engineering Practice
CO-3	Students will learn how to assess the quality of existing software, understand its structure, and understand the risks of leveraging such software for a desired solution	Design / Problem Solving
CO-4	Students will understand and apply process knowledge to guide the transition of software artifacts across process phases	Critical Thinking & Decision Making / Engineering Practice
CO-5	Students collaborate on a team to overcome obstacles	Teamwork / Communication

Course outcomes describe what you as students are supposed to learn in the course. They are mapped to Program Outcomes, which describe what you should learn in the course of your degree program (If you don't know what your program outcomes are, ask me for a copy). Course Outcomes are supported by Course Topics.

Course Topics

This course covers topics in the area of Quality Software Development Practices via the Software Enterprise pedagogical model. Students will be expected to learn, understand and apply these practices through reading, discussion, reflection, practice, and applying in context. These best practices include:

Topic
Agile software development
Software change management and source code control
Continuous Integration and Testing
Unit Testing
Code Reviews
Software Measures and Metrics
Refactoring to Design Patterns
Static and Dynamic Analysis
Defensive Programming

Other topics will be introduced as needed by the instructor.

Textbook

The *required* textbook for the Software Enterprise sequence of courses is Ian Sommerville's *Software Engineering 9th edition*. This should already be in your possession. Given the nature of the class however, we will also rely on outside readings. Note: there now is a 10th edition of Sommerville's text. If you have not purchased the text yet then please get this version. However I am not asking you to purchase a newer edition of the book if you already have the 9th edition from a prior course.
Recommended: Code Complete 2, Steve McConnell, Microsoft Press; 2nd edition June 2004. ISBN: 0735619670

Grading

Category	Percent of Grade
Project	30% (separate grading criteria provided at project kickoff)
Labs	30% (5% per 6 labs)
Process	20% (incorporating 5 quality practices in your project)
Exam	20% (one final exam)

Project

All students shall participate in a team project. A project kickoff with documented project expectations will be distributed early in the semester. Project grades will be assigned based on 1) quality of the final project deliverables, 2) preparation and quality of intermediate (per sprint) software deliverables and presentations, 3) evidence of faithful execution of the Agile process practices (Scrum), and 4) evidence of consistent effort throughout the semester in support of the team. All projects will participate in the Innovation Showcase at the end of the semester. The project kickoff document will describe project grading and rubrics in detail.

Labs

The Software Enterprise pedagogical model defines a mechanism for introducing software engineering concepts in the context of project work. An Enterprise course is delivered as a series of modules interspersed with project activities and time for reflection. A set of lab modules, derived from the Topics list above, will be delivered in this way this semester. For each module, students are expected to execute a series of steps: *preparation, discussion, practice, apply in context,* and *reflect*. You will be individually assessed on each of these steps for each of these lab modules. Module activities may include a quiz, a practice session, and a project reflection.

Process

The module topics described above are intended to teach you quality software development practices. You are expected to demonstrate FIVE of the quality practices within the context of your team project. More details to come at project kickoff.

Exams

There will be a cumulative final exam worth 20% of the final grade, given during the assigned period for the final exam in this course. Quizzes will not count as part of the exam grade but instead be counted as part of the module grade.

Student Expectations and Responsibilities

- This course is the exact same course offered to on-campus students.
- However, as the nature of online education is distinct, and the timeline for this online course is less than one-half the time as on campus students, there will be customization of my classroom policies.
- These policies will be posted on Blackboard and reviewed via video and discussion forums at the start of the session.

Grade Appeals

Students have the right to appeal a grade in writing. Submit your typed appeal with the graded item, stating the reason for your appeal. All appeals must be turned in no later than one week after the material has been returned in class or on Blackboard.

Cheating and Ethics Policy

This is a team-oriented course, students are expected to support their team to the best of their ability and capability, in a manner befitting the ethics of the software profession. Cheating is strictly forbidden in this class. Cheating is defined as "presenting someone else's work as your own". In this class, any cases of suspected violations will be turned over to the Dean's Office of the Fulton Schools of Engineering who will track violations and determine additional actions. Sanctions may include an E or EX grade in the course. Students should review the student honor code (http://engineering.asu.edu/integrity/honor-code/) and are encouraged to take the honor code pledge. Students are expected to follow ASU's Academic Integrity Policy, https://provost.asu.edu/academicintegrity.

The penalty for cheating in this class is immediate failure of the course. There will be no warnings or exceptions.

The most important thing to remember is to KEEP UP. You earn your grade by executing the project. Trying to do all of your work in the last week of the course will get you a very minimal return as there is no way you can do the project and practices in a single week!