COMP 425
Software Engineering

Possible Text: Object-Oriented and Classical Software Engineering by Stephen R. Schach, McGraw Hill 2011.

Objectives: In this course students learn how to use an iterative and incremental process to develop software that is fault-free, maintainable, delivered on time that satisfies all the requirements.

Structure: Class will be divided into 3-4 person teams. Each team will be provided a set of high-level requirements to be implemented, integrated, and tested. Each team's requirements will be implemented using the software engineering processes covered in the course. Activities will begin with project planning and culminate in a review of the team results. Each team will be required to develop and maintain documentation, use UML through all stages provide three presentations during the development of the project. The first two presentations will review the objectives and architecture of the selected software system. The final review will compare actual results to initial plans and then develop a set of lessons learned for future projects (what worked well, what did not work well, and for elements that did not work well, recommendations for improving the process). Teams may define their own project (subject to instructor approval) or select a project from a list provided by the instructor. Teams will also be allowed to select their own development environment and languages, but must be able to demonstrate the application and provide access to the source code to the instructor.

Grading: Final grades will be based on the following percentages.

First presentation ……..……..…25 %
Second presentation…….. 25 %
Documentation…….. 25 %
Software quality, testing and project legacy……..…….25 %

Attendance: Students are expected to attend each class.

Academic Dishonesty:
Acts of dishonesty will be handled in accordance with SMSU's academic dishonesty policy. While you are encouraged to collaborate when working on homework assignments, you should not share your finished work with someone else nor ask someone else to share theirs with you.

Course Contents

This outline is only meant as a rough guide for the semester. Some topics listed may be abbreviated or eliminated. Additional topics may be introduced. The material covered for each topic, the exact nature of the assignments and the pace of the course will be dependent on the overall progress of the class.

1. Software Engineering Overview
2. Project Planning and Management
    a. Lifecycle Models
    b. Estimation
3. Requirements Specification and Management
    a. Techniques and Tools
    b. Functional Requirements
    c. Performance Requirements
4. Development Tools and Methods
    a. Documentation
    b. Peer Reviews/Inspections
    c. Configuration Management
    d. Case Tools
    e. Metrics
5. Software Architecture/Top-Level Design
    a. Architecture Styles
    b. Reuse
    c. Commercial Off-the-Shelf (COTS) Software Products
6. User Interface Design
    a. Techniques
    b. Use of Patterns
    c. Guidelines
7. Detailed Design
    a. Design UML
    b. Data Structures
    c. User Interfaces
    d. Algorithms
8. Implementation
    a. Coding
    b. Unit testing and debugging
9. Independent/Formal Testing
    a. Test Levels
    b. Test Planning
    c. Test Execution and Reporting
    d. Analysis of Test Results