CS3450 – Exam Study Guide

General Comments

The midterm will cover everything contained in the textbook or covered in class up through September 30th. It will include a variety of questions, namely fill-in-the-blank, true-false, multiple-choice, short answer, and matching questions. There will also be at least one question that asks you to draw one of the diagrams we talked about in class (e.g., use case diagram, class diagram, sequence diagram) The test will take place in the testing center between October 4th and October 8th. We will not have class on Friday, October 4th in order to have time to take the exam. You will have 2 hours (or whatever the testing center allows) to complete the test.

This test will be a closed book and closed notes exam. Please read each question carefully before forming your answer.

Topics to Study

Chapter 1: Software and Software Engineering Basics

Purpose of Software Engineering

Basic steps for successful software engineering project (requirements gathering, high-level design, low-level design, development, testing, deployment, maintenance, wrap-up)

Chapter 2: Before the Beginning (Document Management)

Importance of documentation

Version Control

- What is it and why do we use it?
- Features: Reversion, Change/Bug Tracking, Branching, Merging
- Git process
- Git command reference

Chapter 3: Project Management

Importance of Project Management

PERT charts

Critical Path

Gantt charts

How to improve Time Estimates

Chapter 4: Requirement Gathering

Functional and Non-Functional Requirements

Unified Modeling Language (UML)

Prototypes

Chapter 5: High-level Design

Software Architecture Types (Monolithic, Client/Server, Component-based, Service-oriented, Data-centric, Event-driven, Distributed)

Class Diagrams

Use Case Diagrams

Sequence Diagrams

Chapter 7: Development

Self-documenting code
Good programming practices

Chapter 8: Testing

Levels of testing (unit, integration, automated, component interface, system, acceptance)
Testing techniques (e.g., Black-box testing, White-box testing)
Testing pyramid

Chapter 9: Deployment

Cutover
Staging

Chapter 10: Metrics

Defect Analysis
Attributes - Examples
Metrics
Indicators

Chapter 11: Maintenance

Categories of Maintenance Tasks (Perfective, Adaptive, Corrective, Preventative)
Refactoring and why you might refactor code
Why does maintenance take so much of a project’s total cost?

Chapters 12, 13, & 14: Process Models

Four Types of Approaches (Predictive, Iterative, Incremental, Agile)
Waterfall, Waterfall with feedback, Sashimi, Incremental waterfall, Spiral, Cleanroom
Rapid Application Development
Extreme Programming
SCRUM
Test-driven development
Continuous Integration