SE2890 Software Engineering Practices

Dr. Walter Schilling

Spring, 2012-2013

On the midterm and final exam, you are permitted one 8.5 x 11 sheet of paper with notes.

The following paragraph describes a system which you have been asked to design and implement. This system will be used for several questions on the exam. Read through the system description so you are familiar with it and think about the potential design questions which can be asked.

> A software system, "MSOE Ride" is to be developed to allow students at MSOE to share rides home and to other locations on weekends. At its highest level, it will allow two students to coordinate a trip to a common location or a location which is on the way to another location.
>
> The system, as it is designed, will interact with the existing LDAP database for student authentication, as all students must log into the system.
>
> If a student on MSOE has a car, they can sign up to be a "designated chauffeur". By signing up to be a "designated chauffeur", a driver certifies that they have a valid license, consent to a background check, and consent to having their insurance verified. If the student fails at any of these tasks, they will not be permitted to be a designated driver.
>
> A designated "designated chauffeur" means that the driver can "advertise" on the MSOE ride system when they are going to a location and how many "passengers" can be taken. To do this, they will enter their ultimate destination and a range of departure times from campus.
>
> A student who wishes to take a trip can log onto the system and find a "designated chauffeur". The student will indicate where they would like to go and the time range for departure. The system will then try to find a matching driver. If a driver is found, the system will setup for the driver and passenger to negotiate on a rate for the trip. If a driver is not available, the system will keep the request in case a new chauffeur adds a trip which would match.
>
> During negotiation, the driver and passenger determine the exact departure time and location, as well as how much the passenger will pay the driver for gas. This will result in either the trip being setup or not. Once a trip is setup, calendar reminders will be sent to the driver and passenger using the existing MSOE Outlook system.
>
> When a trip is completed, both the passenger and the driver will be asked to evaluate the passenger and the driver on their punctuality. The driver will also be assessed for their safe driving skills. When the evaluations are completed, the system will automatically transfer funds from the passengers student account to the driver's student account using the existing MSOE Bursar's system.
>
> Both passengers and drivers are allowed to enter a profile. The profile would indicate the desired sex of the driver or passenger, whether smoking is allowed or not, the style of music desired, and other related items.
>
> The administrator of the system shall have the ability to monitor system usage as well as ban users who abuse the system.

1. Week 1 - Introduction
   (a) Lecture 1: Introduction
      i. Recognize the economic impacts of software failure
      ii. Discuss the ramifications of software failure
      iii. Understand that to improve software quality, we must learn from our past failures
      iv. Explain the concept of the CMM and CMMI process initiatives.
   (b) Lecture 2: Software Development Processes
      i. Explain the difference between general software and real-time software
      ii. List the three tracks of software engineering lifecycles
      iii. Define the term software process and software methodology

iv. Compare and contrast the aspects of the Waterfall model, the V model, spiral models, and other software development models.
v. Explain the ROPES Software Development Process (Rapid Object Oriented Process for Embedded Systems)
vi. Explain the relationship between Systems Engineering and Software Engineering
vii. Define the concept of an Architecture

2. Week 2 - Requirements and Use Cases

(a) Lecture 1- Requirements
i. Recognize the relationship between different types of requirements within the realm of software engineering
ii. Compare and contrast functional requirements, non-0functional requirements, and constraints.
iii. Critique the wording of a requirement and constraints.

(b) Lecture 2 - Use Cases
i. Define a use case
ii. Interpret the meaning of a use case diagram.
iii. Define Actor
iv. Explain the relationship between Use Case Diagrams and Use Case Scenarios
v. List the items present in a use case scenario
vi. Construct a use case scenario for a given problem
vii. Explain how the level of detail in use cases may change throughout the phases of the software development process.

3. Week 3

(a) Lecture 1 - Introduction to Version Control and Configuration Management
i. Explain the concept of a version control system
ii. Recognize the importance of disciplined configuration management to the successful completion of a software development project
iii. Understand the model of a client server configuration management system.
iv. Define repository
v. Define local working copy
vi. Explain the concepts of checking out, committing, and updating.
vii. Explain the concept of merging code files

(b) Lecture 2 Object Domain Analysis
i. Explain the purpose for Object Domain Analysis.
ii. Explain the relationship between the use case model and the Object Model
iii. Explain the mechanism used to connect object domain models with use case models
iv. Apply key strategies for identifying objects within a problem domain
- Noun Strategy
- Services
- Physical Devices
- Persistent Information

4. Week 4 Object Domain Analysis

(a) Lecture 1 Domain Modeling - Part 2
i. Define aggregation and inheritance in terms of UML.
ii. Critique a model for correct usage of UML.
iii. Explain the difference between compact and expanded versions of UML.
iv. Review Aggregation and composition in UML (Lab Review)
v. Construct an object domain model for a given problem based upon a use case

(b) Lecture 2 Defining Object Behavior
i. Explain the relationship between simple, state, and continuous behaviors.
ii. Define entry actions, exit actions, and activities.
iii. Explain the transaction syntax for transitions.

iv. Define guard condition.
v. Define object state behavior using a UML state charts
vi. Represent concurrent state machine behavior using Harel state machines.

5. Week 5 Defining Object Behavior

(a) Lecture 2 Defining Object Behavior

i. Not to be covered on the exam.

(b) Lecture 2- Introduction to Software Reviews

i. Explain the concept of a software review
ii. Illustrate the flow for a typical software review
iii. Explain the differences between an inspection and a walkthrough.
iv. Recognize the applicability of checklists to improving the review process
v. Identify the risks of reviewing at too fast or too slow of a rate.