Introduction

In this course, the written exam will count 50% of the final grade, and the remaining 50% of the final grade comes from the compulsory exercises.

So, your final grade of this course will be:

Points you get from this written exam + points you get from the compulsory exercises.

If you feel that any of the problems require information that you do not find in the text, then you should

- Document the necessary assumptions
- Explain why you need them

Your answers should be brief and to the point.

You need to put all your answers to a .pdf file and upload the .pdf file to Inspera before the examination time expires.

Problem 1 – Use case (5 points)

Many ATM (Automated Teller Machine) of banks allow users to withdraw cash from a bank account and to store money to a bank account. **Based on the use case template taught in this course, write a use case** “store cash to a bank account.”

Storing cash to a bank account means that:

- The users can put some money in the ATM
- The ATM can identify whether the money is true/faked
- The ATM can sum up the total amount of the money
- The ATM can add the total amount of the money to the user’s bank account

Problem 2 – Goal-oriented requirement engineering (5 points)

Based on goal-models taught in this course, make a goal-model of the control system of the “autonomous bus” using soft goal types. The goal-model should have at least three layers.

(Note: You can use whatever software to draw the goal-model. Handwriting drawing is not acceptable).

Problem 3 – System test (12 points)

Suppose you will lead the system test of a web application that allows users to buy/sell stock shares in real-time. Your task is to explain to your top-level managers the purpose of each type of the following system test and give them at least one test case example.

- Robustness test
- Scalability test
- Stress test
- Load and stability test
- User acceptance test
- Operational acceptance test

Problem 4 – Code review and testing (8 points)

Use the bugs and code smells your group identified in the exercises as examples to explain the pros and cons of testing and code review, and explain why testing and code review complement each other to assure software quality. You need to list at least two bugs your group identified through testing, and at least two code smells your group identified through code review as examples.

Problem 5 – Bug tracking and software process improvement (10 points)

Suppose the current bug tracking system of your company has only three fields:

- Bug description (free text)
- Bug fixing description (free text)
- Name of the person who fills in the information

Your company wants to achieve the following goals through improving data collection and data analysis of the bug tracking system:

- Reduce the number of severe bugs
- Improve bug-fixing efficiency

Your task is to use the GQM (Goal-Question-Metrics) approach (and possibly combined with the IBM Orthogonal Defect Classification) taught in this course to propose data fields to be added to the bug tracking system to collect the data for achieving the goals. You should also briefly explain how you are going to analyze the collected data.

Problem 6 – OSS and software Ecosystem (10 points)

Suppose you have developed a software education game to teach kids mathematics, and you have published the software as open source. Based on what you have learned from the OSS lecture, propose your strategies to attract more open source developers to help you grow the software game fast and successfully.

Your strategies should cover the following aspects. You should briefly explain what strategies you choose to use and why you choose a particular strategy.

- License type (1 point)
- Software engineering practices (5 points)
- Social process (4 points)