University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

Spring, 22/23 TRIMESTER EXAMINATIONS

COMP47480

Contemporary Software Dev

Module Coordinator: Assoc Professor Mel Ó Cinnéide

Student Number

Seat Number

Time Allowed: 120 minutes

Materials Permitted in the Exam Venue:
English language dictionary (hard copy)
Foreign language dictionary (hard copy)

Materials to be Supplied to Students:
12 Page Answer Booklets

Instructions to Students:
Each question carries 20 marks.

Attempt any four questions.

Question 1

(a) What are the essential features of an Agile software development process?
(5 marks)

(b) Explain why Agile has become the dominant process in software development.
(5 marks)

(c) Explain what an *executable baseline architecture* is, and the benefits that accrue from creating one.
(5 marks)

(d) Explain briefly what DevOps is, and the problems it aims to solve.
(5 marks)

Question 2

(a) Explain briefly how the three key UML diagram types (*use case*, *class* and *interaction*) combine to describe a software design. Draw a sample use case diagram, a class diagram of the associated domain and an interaction diagram for a single scenario of one use case.
(15 marks)

(b) Explain why UML is not widely used in the software industry.
(5 marks)

Question 3

In relation to software testing, write a note on each of these topics: (i) test-driven development, (ii) basis path coverage and (iii) mutation testing.
(20 marks)

Question 4

(a) Explain the Liskov Substitution Principle and show with an example the impact this principle has on how methods can be overridden in a subclass. How does this principle relate to the Refused Bequest code smell?
(10 marks)

(b) Explain the Law of Demeter, provide a code example of it being violated and explain the benefits of following this principle.
(10 marks)

Question 5

(a) In relation to the State design pattern: (i) What is the intent of the pattern? (ii) Draw a UML class diagram that describes the typical structure of the pattern. (iii) Describe three interesting issues related to the applicability or implementation of this pattern. (iv) How does this pattern relate to the Feature Envy code smell?

(15 marks)

(b) Explain the risk of overengineering when using design patterns, and how it can be avoided.

(5 marks)

Question 6

Write a note on refactoring, including (among other topics): descriptions of several individual refactorings, a discussion of the role of refactoring in software development, a discussion of the role unit tests, code smells and design patterns play in refactoring and accounts of any interesting refactoring episodes from your own experience.

(20 marks)

oOo