TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2006

MID-YEAR

COMP 301
Software Engineering Principles

Time Allowed: 3 Hours

Instructions: Answer all questions.
Total marks are 150.
Use the marks for each question as a guide as to how long to spend on it.
No calculators are permitted.
Paper dictionaries for translating between English and a foreign language are permitted.

Question 1. General Knowledge [25 marks]

(a) [3 marks]

During the latter half of the twentieth century there was the perception that a software crisis existed. What was perceived to be the central concern in the software crisis?

(b) [4 marks]

Use cases are typically created during a requirements-gathering activity. How might use cases be used during a software testing activity?

(c) [6 marks]

In his book "The Mythical Man-Month", Fred Brooks states that you should "plan to throw away once". Why would developers benefit from planning to throw away once?

(d) [6 marks]

We often use the same notation (e.g. sequence diagrams, class diagrams, ...) in using UML models to document the analysis and design of a system. Even though we are using the same notation in both activities, the content we create for each is different. Discuss an advantage and a disadvantage of using the same notation for both analysis and design.

(e) [6 marks]

Discuss whether or not the following claim is valid: "Microsoft Windows Malicious Software Removal Tool can be downloaded for free from Microsoft's website, so therefore it is Free Software (as defined by the Free Software Foundation)."

Question 2. Requirements [20 marks]

(a) [2 marks]

Identify one benefit of modelling use cases with a use case diagram.

(b) [2 marks]

What do the set of actors represent in the analysis of a system?

(c) [4 marks]

Given that a use case documents a single task performed using the system, why might you have multiple sequence diagrams associated with the use case?

(d)

A use case describes an interaction with the system. A functional model documented with use cases decomposes the system requirements based on the tasks to be supported. Not all requirements map directly to one task however, and not all requirements are functional in nature.

(i) [3 marks]

Describe one technique for documenting a functional requirement that maps to multiple tasks in use cases. Your technique may apply to the use case text, or to the use cases' UML representation, or both.

(ii) [3 marks]

Describe one technique for documenting non–functional requirements in use cases. Your technique may apply to the use case text, or to the use cases' UML representation, or both.

(e) [6 marks]

Use cases decompose a system by task. Discuss how this may or may not help software developers check that each part of their design solves some part of the problem, and that each part of the problem is solved by some part of the design.

Question 3. Design [35 marks]

(a) [9 marks]

Weaknesses in each of the twelve concepts of Extreme Programming are supposedly compensated for by strengths in the other concepts. This means that each concept is linked to some subset of all the other concepts. What set of Extreme Programming concepts would be linked with the concept of having a Simple Design? Why?

(b) [10 marks]

Discuss how design patterns may help or hinder the practice of the values and principles of Extreme Programming.

(c)

Structured Design is a design strategy that was popular in the 1970s and 1980s. Yourdon and Constantine proposed a Structured Design technique called Transform Analysis.

(i) [8 marks]

How does the modelling of data differ between Structured Design and Object Oriented Design?

(ii) [8 marks]

Does Yourdon and Constantine's heuristic of "scope of control versus scope of effect" apply to Object Oriented Designs? If so, why? If not, why not?

Question 4. Implementation and Testing [20 marks]

(a) [5 marks]

What does path coverage mean in the context of software testing?

(b) [5 marks]

Describe the concept of test-driven development.

(c) [5 marks]

Identify one language feature in Java that can be used to implement pre–conditions and post–conditions documented in the design. What is one potential problem associated with significant use of this language feature?

(d) [5 marks]

What is the difference between horizontal and vertical mapping when you are transforming entity objects in a design into a database schema?

Question 5. Lifecycles and Methodologies [25 marks]

(a) [7 marks]

What is the connection between risk analysis and prototyping in Barry Boehm's Spiral Model?

(b) [8 marks]

You are working in a team on a software project using Winston Royce's Waterfall lifecycle model. Describe two characteristics that you would expect the system under development to have if the Waterfall lifecycle model is an appropriate choice for the team.

(c) [10 marks]

Is the claim that the rational unified process is an agile methodology valid? In your answer you should state whether the claim is valid, and discuss why you believe it is valid or not.

Question 6. Free and Open Source Software [25 marks]

(a) [8 marks]

While Free Software and Open Source Software share many similarities and points of agreement, they are subtly different. Discuss one difference between Free Software and Open Source Software.

(b) [17 marks]

Compare and contrast open source development with extreme programming. You should identify at least three similarities and three differences.

****