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

EXAMINATIONS — 2007

MID-YEAR

COMP 301

Software Engineering Principles

Time Allowed: 3 Hours

Instructions: Answer all questions
Total marks are 180.
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 [30 marks]

(a) [6 marks]

Developers often use version control tools such as CVS or Subversion (SVN). List three ways that version control tools can assist during the maintenance activity.

(b) [6 marks]

Describe the fundamental difference between white box testing and black box testing.

(c) [4 marks]

Is Free Software (as defined by Richard Stallman and the Free Software Foundation) "free as in speech" or "free as in beer"?

(d) [6 marks]

Use cases and scenarios created during the requirements analysis activity can be used in the system testing activity. Why is it useful to reuse use cases and scenarios in the system testing activity?

(e) [8 marks]

Yourdon and Constantine's transform analysis technique is a form of structured design. One step in the technique is to identify the afferent and efferent data elements in a data flow diagram. What is the purpose of identifying the afferent and efferent data elements?

Question 2. Patterns [20 marks]

(a) [5 marks]

What is a design pattern?

(b) [5 marks]

What is an anti-pattern?

(c) [10 marks]

A developer has a problem of ensuring that a class only has one instance, and that the class provides a global point of access to the instance.

Identify the pattern you would recommend for this problem and describe the solution and consequence components of that pattern.

Question 3. Aspect- & Object- Oriented Design [20 marks]

Consider the following simple problem description:

A University needs to manage student and course information. The database administrator can add a teacher to a course. A teacher of a course can add a student to the course and later assign a grade to that student. A student may request a list of the courses they are in, and a teacher can request a list of the courses they teach. Whenever a teacher is added to a course, or whenever a teacher adds a student to a course or assigns a grade to a student, that event must be logged in a file. The database administrator can also view the file. Lastly, a student may request a list of the grades they have been given so far.

(a) [3 marks]

Identify a probable cross cutting concern in the problem description above.

(b) [12 marks]

Consider how your probable cross cutting concern would be addressed in an object-oriented system, and how it would be addressed in an aspect-oriented system. Discuss two reasons why aspect-oriented proponents believe their approach is better than the object-oriented approach for addressing your cross cutting concern.

(c) [5 marks]

Discuss one way in which software created using the object oriented approach may be more reusable than software created using the structured design approach.

Question 4. Deployment & Maintenance [20 marks]

(a) [10 marks]

You have developed a system that meets the core requirements defined in the requirements analysis document agreed to by the client, and now you need to deploy the system to the client's machines. Describe two additional software systems you would ideally use (or create specially) for deploying your system.

(b) [10 marks]

Three categories of maintenance are corrective, adaptive and perfective. For each of the three categories, discuss whether or not the Y2K bug (also known as the "Millenium Bug") is an example of that category.

Question 5. Testing [35 marks]

(a) [4 marks]

Identify a testing-specific software tool that is useful when beta-testing software.

Note: The tool you identify does not need be *only* useful during beta-testing, but should have some specific functionality that is useful when dealing with issues arising from "testing in the wild".

(b) [11 marks]

Discuss whether the following statement is true: "white box testing can help find functionality specified in the requirements that is missing in the implementation".

(c) [10 marks]

If you were following an agile methodology such as extreme programming, in what order would you perform the following tasks when coding a software component (such as a Java class) using a unit testing suite (such as JUnit)? Justify your answer.

- create the component's interface.
- create the component's implementation.
- create the component's test cases.

(d) [10 marks]

A method for calculating insurance takes two integer parameters called *age* and *health rating*. The *age* parameter should be a value from 18 to 85. The *health rating* parameter should be a value from 1 to 10. A developer who has been asked to unit test this method has used equivalence classing and boundary analysis to come up with the following test case inputs:

Test Case	*Age*	*Health Rating*
1	19	9
2	17	0
3	83	10
4	120	15

The developer believes that if the method has the right behaviour for these test case inputs (i.e. signals if the parameters are invalid or performs the calculation if the parameters are valid) then the method is likely to be correct for all other possible test case inputs.

What are two problems with the developer's belief?

Question 6. Software Lifecycles [35 marks]

(a) [10 marks]

In "The Mythical Man Month", Fred Brooks claims that developers should "plan to throw away once". How does this claim relate to Winston Royce's original version of the Waterfall Lifecycle?

(b) [15 marks]

Weaknesses in each of the twelve concepts in Extreme Programming are supposedly compensated for by the 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 Continuous Integration? Justify your answer.

The twelve concepts are:

- Collective Ownership
- 40-hour Work Week
- On-site Customer
- Coding Standards
- Metaphors
- Small Releases
- Planning Game
- Simple Design
- Refactoring
- Pair Programming
- Continuous Integration
- Testing

(c) [10 marks]

Define each of the four phases in the Rational Unified Process. Each definition should only be a sentence or two long.

Question 7. Free & Open Source Software [20 marks]

(a) [10 marks]

Discuss whether the following statement is true: "in practice, everyone who uses an open source application has equal control over the direction of that application's future development".

(b) [10 marks]

What is one advantage and one disadvantage of developing using an open source methodology?

****