TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

TU Delft

Exam IN3205-I
Software Quality and Testing
Softwarekwaliteit en Testen

June 19, 2009

- There are 5 questions worth of 25 points in total.
- Total number of pages: 2.
- Use of books and readers is not allowed.
- Please write clearly: points may be deducted for unclear or sloppy answers.
- Please don't guess: points may be deducted for incorrect or superfluous answers hiding the correct answer.
- You can answer in English or in Dutch
- Please list your answers in the right order
- The tentative grading scheme is:

Question:	1	2	3	4	5	Total
Points:	5	4	3	4	9	25
Score:						

- If p is the number of points you score, the exam grade E will most likely be determined by

$$E = 1 + 9 * p/25$$

- If the grade for your labwork in 2009 was "good", this can earn you 0.5 extra points.

GOOD LUCK!

1. Let us assume Microsoft's NAV product can be installed for 25 different languages (locales), 10 different major operating system (OS) releases (XP, Vista, various service packs, ...) 5 different major Office releases, and 4 different database servers.

   Since any combination among those four factors can be faulty, we assume Microsoft tries to run its full build and test cycle (which we assume to take 6 hours for a single combination) on a different combination each night.

   (a) (1 point) Assume Microsoft picks one arbitrary combination every night. What are the advantages and disadvantages of this approach?

   (b) (1 point) Explain the pairwise combination testing strategy.

   (c) (1 point) To how many test cases would the pairwise combination testing strategy lead for the given NAV setting? How many cases would be needed for exhaustive combinatorial testing?

   (d) (1 point) In practice, not all combinations may be possible. For example, very new versions of Office may not run on very old OS versions. What are the implications of simply dropping combinations violating such constraints?

   (e) (1 point) What happens if the daily build finds a bug in a new combination after a release-candidate has been shipped, but before the release to manufacturing is out?

2. Automated generation of test data and automated execution of the corresponding test cases is an appealing idea, but requires an *oracle*.

   (a) (2 points) What is an oracle?

   (b) (2 points) Discuss the most important ways of obtaining an oracle, and discuss the advantages and disadvantages of each approach in light of the automated generation of test cases.

3. (3 points) Let class A be a superclass of class B, and assume that this superclass relation adheres to the principles of design by contract. Furthermore, let m be a method implemented in A that is overridden in B. Which of the method implementations, $A.m$ and $B.m$, is the most defensive? Explain your answer.

4. (4 points) While working for software company C, you created a test and integration plan for software system X. Key steps in your plan include ensuring (as close as you can get to) 100% statement coverage for the components used or developed, and a period of 4 weeks intensive integration testing.

   Your manager M studies the plans, and proposes to reduce the period of integration testing to just a single week, based on the argument that with 100% statement coverage there is no need for so much integration testing.

   Write a memo of at most 150 words to your manager, in which you include at least three good reasons in defense of your original plan.

5. While studying the code of a tutorial on service-oriented architectures, you encounter a fragment similar to the one displayed in Figure 1.

   (a) (1 point) Draw a control flow graph for the `spell` method;

   (b) (1 point) Define the statement adequacy criterion.

   (c) (1 point) Discuss with which (minimal) input you can achieve the highest statement coverage.

   (d) (1 point) Define the branch adequacy criterion.

   (e) (1 point) Discuss with which (minimal) input you can achieve the highest branch coverage.

   (f) (1 point) Define the path adequacy criterion.

   (g) (1 point) Discuss with which (minimal) input you can achieve the highest path coverage.

   (h) (1 point) Define the loop boundary adequacy criterion.

   (i) (1 point) Discuss with which (minimal) input you can achieve the highest loop boundary coverage.

```
public void spell() throws IOException {

  DictionaryService dictionary = findDictionary();

  if (dictionary != null) {
    System.out.println("Enter a blank line to exit.");
    String word = "";
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

    while (true) {

      // Ask the user to enter a word.
      System.out.print("Enter word: ");
      word = in.readLine();

      // If the user entered a blank line, then
      // exit the loop.
      if (word.length() == 0) {
        break;
      }
      // Otherwise print if the word is ok.
      else if (dictionary.correctSpelling(word))
      {
        System.out.println("Correct.");
      }
    }
  }
}
```

Figure 1: Code fragment from spell checking service

End of exam