

ICS 52 - Introduction to Software Engineering
Final Exam – Fall, 2009

First Name: _____ Last Name: _____

| | | | | | | | |
|---|---|---|---|---|---|---|-------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | Total |
|---|---|---|---|---|---|---|-------|

1. (11 points) Chapter 1 of the textbook (“Introduction”) discusses several prominent software system failures. We also discussed a few software non-success stories from recent news in lecture. The Scientific American article “Command and Control” is about a software project that eventually ended up being quite successful. Select one software failure story from the textbook or lecture, and compare it to the “Command and Control” project. Focus on **one** quality, process, or principle that has been covered in ICS 52 and explain how that concept helps understand the failure and the success. Your answer should be detailed and specific.

2. (16 points) You have been assigned to test a Pepsi Cola soda machine (such as the one outside the ICS building) using a black box approach. Your task is to focus on features important to a typical customer, specifically the functionality of selecting, paying for, and receiving a bottle of soda.

(a) What is the input domain of this functionality?

**The (possibly empty) set of coins deposited in the coin slot;
the bill entered in the bill entry area, if one was entered;
the soda selection buttons pressed;
and whether the coin return button was pressed.**

(b) What is a basis for dividing the input domain you described into subdomains?

**The sum of the coins and bill deposited. Assume the
coin return button was not pressed, and exactly one
soda selection button was pressed.**

(c) Using the basis from (b), name three or four subdomains.

I will assume that all sodas have the same price, which is P.

- 1. Sum is zero.**
- 2. Sum is less than P.**
- 3. Sum is equal to P.**
- 4. Sum is greater than P.**

(d) For each subdomain from (c), give a test case input and the expected output.

Let P be 75 cents.

- 1. Input: no coins or bills. Expected output: No soda.**
- 2. Input: two quarters. Expected output: No soda.**
- 3. Input: three quarters. Expected output: Requested soda.**
- 4. Input: one dollar bill, no coins.**

**Expected output: Requested soda, any combination of
coins totaling 25 cents in coin return slot.**

3. (24 points, 4 points each) Define the following terms, as used in software engineering:

A. Function point.

One measurable unit of (or, A way of measuring) a software system's complexity and capacities; used to measure software size.

B. Unit test.

Determining whether a small piece of software functions correctly; often performed by the programmer. (-2 in many cases because "test" was used, undefined, in the definition.)

C. Regression testing.

Application of previously developed test cases, applied after software maintenance to make sure new errors have not been introduced. (Here we allowed "test" in the definition; 2 pts for clearing indicating the same test cases are used repeatedly.)

D. Static analysis technique (for testing)

An approach to quality assurance which does not involve program execution. Much of this is done by the compiler. Exaamples are reading, walkthroughs and inspections, and correctness proofs.

E. Equivalence partitioning.

Dividing the set / domain of inputs into subsets based on some criterion.

F. Testing oracle.

A mechanism for determining whether a test output is correct / for determining the expected output.

4. (12 points) Describe in a few words each quadrant of the spiral model.

3 points each; see page 62

A. Upper left (northwest).

B. Upper right (northeast).

C. Lower right (southeast).

D. Lower left (southwest).

5. (12 points) Since testing is never complete, we need to have some criterion for determining whether a certain amount of testing is adequate. Define the following three categories of test adequacy criteria:

3 points each, see p. 409

Coverage-based testing

Fault-based testing

Error-based testing

Which category or categories apply to homework 4? Explain why.

3 points for this part

White-box: coverage-based

Black-box: coverage-based or fault-based

6. (15 points) Consider the following phrases: “a module’s interface,” “a user interface,” “the Java keyword `interface`.” What is the shared meaning in these three uses of the word “interface”?

The interface is the border or point of interaction between two separate systems or parts of a system.

In light of the shared meaning you’ve identified, briefly define each of the three phrases.

a module’s interface

The services one module provides to another module.

a user interface

The way a human user interacts with and controls a computer system.

the Java keyword `interface`

A contract or guarantee that a module will have certain methods in its module interface.

7. (10 points) Someone wrote that “every software quality must be verified.” Choose a software quality other than correctness, and briefly describe how it might be verified.

Performance can be verified by exercising the software under realistic conditions and with the expected volume or rate of inputs, and measuring whether the usage of CPU, memory, disk space, and other resources is within the specified limits.