Exam in Testing

Justin Pearson

December 2009

Cover Sheet

Problem no.	Solution provided	Max	Your points
1		6	
2		8	
3		9	
4		8	
5		5	
6		9	
	Total:	45	

Name : .....

Pers.no. : .....

Exam Rubric

All answers to be written in English or Swedish. A mark of 50% is required for a 3, the marks 4 and 5 are evenly distributed in the interval from 50% to 100%. I will not be able to come to the exam to answer questions, if you have trouble understanding a question then please state clearly any assumptions that you have to make to answer the question.

Hjälpmedel:

Pen,pencil, ruler,rubber, dictionary, calculator.

1. General Questions on Testing.

   (a) Describe the difference between *validation* and *verification*. (**points 2**)

   (b) Describe the difference between a *Software Fault* and a *Software Failure*. (**points 2**)

   (c) Is it ever possible to completely test a piece of software for correctness? You should provide some justification for your answer (that means saying yes or no is not enough). (**points 2**)

2. Consider the following fragment of java code:

```
public void displayAnswer(int x, int lang) {
 if (lang == 0) { System.out.println("The Answer is :"); }
 if (lang == 1) { System.out.println("A vlasz :"); }
 if (lang == 2) { System.out.println("Mae fy hofrenfad yn llawn
llyswennod: ");}
 if (lang > 2) {System.out.println("Universal Translator offline.");}
 if (lang < 3) {
   if (x==42) {System.out.println("9 * 8"); }
   if (x != 42) {System.out.println(x);}
 }
}
```

   (a) Draw a Structural Graph that represents the above piece of source code. (**points 2**).

   (b) For you graph, enumerate a complete set of paths that guarantee node coverage. (**points 2**).

   (c) Construct a finite set of test cases that guarantee node coverage. For each test case indicate which execution path is covered. (**points 2**)

   (d) For the above fragment of code is there any difference in requiring complete edge coverage rather than node coverage? Explain your answer. (**points 2**)

3. More on Coverage Criteria. Consider the following fragment of java code:

```
public  int max(int[] t) {
    int maximum = 0;
```

```
    for (int i=1; i<t.length; i++) {
        if (t[i] > maximum) {
            maximum = t[i];   // new maximum
        }
    }
    return maximum;
}//end method max
```

(a) Draw a graph representing the above piece of source code. (**points 2**)

(b) Enumerate all the prime paths of the graph you have constructed. (**points 2**)

(c) Derive tests cases that cover all the prime paths that you have identified. (**points 3**)

(d) The above piece of code has two failures. First it does not handle empty arrays properly but secondly it does not return the correct result if the array only contains negative values. Construct a test case for the second case. (**points 2**).

4. Logic Coverage. Given that the variables `a,b,c,d` are integers consider the following expression:

```
(( a >= b) && (b > 42) ) ||  (a != b) || ( a+b == c+d)
```

(a) Define *predicate coverage* (**points 2**) and for the above expression derive a set of test cases that give predicate coverage. (**points 2**)

(b) Define *active clause coverage* (**points 2**) and for the above example derive a set of test cases that give active clause coverage. (**points 2**)

5. Input Space Partitioning.

(a) For an input domain define what a partition of a domain is. (**points 2**)

(b) Consider the fragment of code in Question 3. Is the following a partition of the input domain? (**points 3**).

- Input `t` is `null`.
- Input `t` is the empty array.
- Input `t` is a array of size 1.

- Input `t` is a array of size greater than 1.
- Input `t` contains a negative value.

6. General Practical Questions on Testing.

   (a) When developing a piece of software from a requirements specification. At this stage what are the major test action and design goals. (**points 2**)

   (b) What happens during integration testing? (**points 2**)

   (c) It is important to keep a good set of tests cases for regression testing, but as the project ages the regression suite can become too large. As a test engineer explain what criteria you would use to choose which test cases remain and which test cases can be taken away. (**points 3**)

   (d) Discuss why it is often useful to have as a test engineer somebody who has not actually developed the component under test. (**points 2**)