COMP5222 Assignment 1

Qu Xiaofeng 09903198R

COMP 5222
Software Testing and Quality Assurance
Assignment 1
Due date: Nov 2, 2012 Total Mark: 60

Question 1

(a) Why is it impossible to guarantee that an arbitrary software application is error free? Give 3 reasons. (3 marks)

"Human make errors in their thoughts, actions and making decisions. Error are a part of daily life."

It can be analyzed in different levels in software development, such as system design, function planning and coding.

In system design level, the communication with clients is always very difficult. People didn't know what they really want. The common case is that the client comes with only a general idea. The design of whole system will be tested for several times by making prototypes, system framework charts, function lists and etc. The case is even worse in function planning level. Functions can be too hard to implement then alternate ways should be taken, or other supportive functions should be added. These all require recursive coding and tests. The errors are always very stupid in coding level and it is really a tar pit consuming lots of time and cost.

Even for a properly run software application, it is impossible to guarantee its quality. There are several possible errors which are very hard to test.

1. Time

Sometimes the time could be a "time bomb", for example, Millennium bug. This error occurs only when it comes to a certain time.

2. Overburden

When the data is too much or the running time is too long, software fails. Maybe it lasts for hours or even days, or it may be years or decays.

3. False Use

The most often and damaging example should be using rm -rf as a unix or linux admin in root. Or some unexpected input data or input sequence.

(b) List 5 characteristics of a test case that is "better" than another test case. (5 marks)

1. Straight to the target

A test case should be aiming to a certain object. Maybe a certain condition, a certain line of code, or a certain boundary. Each failed test case should lead the test engineer to the right point of the defect.

2. Make sure the key component is tested

Due to time/cost limit, the test case testing key component is better than the one testing components never used.

3. Test boundary cases specified by the requirements

There are many possible inputs. The boundary test cases are much better than random test cases.

4. Do not repeat

A test case should be independent with others. There is no need to test multiple times on one same condition except stress test.

5. Cause the program to exercise certain error-prone situations.

Make sure the program will not collapse even the test case is not in requisition.

(c) Identify 5 key process factors that influence the effort required for doing testing. (5 marks)

1. Maturity of the software development process;
2. Quality and testability of the testobject;
3. Test infrastructure;
4. Skills of staff members;
5. Quality goals and test strategy.

Question 2

The following module of the Beer system calculates the cost of wine orders. The wine is sold in boxes of 12 bottles, and is of three types: A. Shiraz, B. Cabernet Sauvignon, and C. Lambrusco. The code is as follows:

```
Procedure CostOfWine (in: no_of_boxA, no_of_boxB, no_of_boxC;
                      out: cost_to_customer) is
Begin
 1    cost_of_boxA = 60;
 2    cost_of_boxB = 80;
 3    cost_of_boxC = 100;
 4    if (no_of_boxA == 2)
 5        discount = 5;
 6    elsif (no_of_boxA == 1 AND no_of_boxB == 1)
 7        discount = 8;
 8    elsif (no_of_boxC == 3)
 9        discount = 10;
10    else
11        discount = 0;
12    ifend
13    cost_to_customer = ( (no_of_boxA * cost_of_boxA) + (no_of_boxB * cost_of_boxB)
                         + (no_of_boxC * cost_of_boxC) ) * (100 - discount) / 100;
14    return (cost_to_customer);
end Procedure CostOfWine;
```

(a) Draw a control flow chart for procedure CostOfWine. (4 marks)

The control flow chart is shown as Figure 1.

(b) Calculate the cyclomatic complexity of procedure CostOfWine. (2 marks)

The cyclomatic complexity is v(g) = e - n + 2 = 16 - 14 + 2 = 4

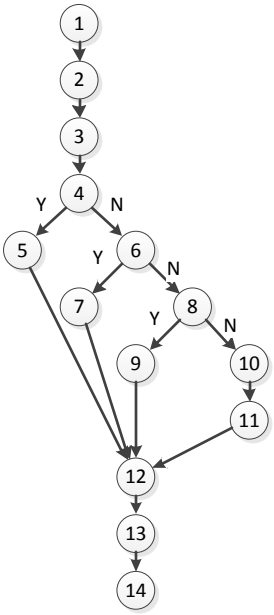

Figure 1: control flow chart

(c) Create test cases that test all the basis paths and provide a set of input values and expected results (in terms of values for cost_to_customer). (5 marks)

path	input	expected result
1-2-3-4-5-12-13-14	(2,0,0)	114
1-2-3-4-6-7-12-13-14	(1,1,0)	128.8
1-2-3-4-6-8-9-12-13-14	(0,0,3)	270
1-2-3-4-6-8-10-11-12-13-14	(0,0,1)	100

Question 3

From an IBM study, verification prior to coding is 50% effective, and after coding is 80%. It is 10 times as costly to correct a defect after coding as before, and 100 times as costly to correct a production defect.

Compare doing verification only after coding vs. during all phases.

Assume

- There are 5 phases: Requirements, Design, Code, Test, and Production.
- 20 errors introduced at Requirements
- 20 errors introduced at Design
- 20 errors introduced at Code

(a) Work out the cost of fixing all defects when verification is done at the Test phase. (4 marks)

Given the condition as described in the question, we can draw a table comparing the debugging cost and defects detection rate like this.

Stage	Errors Introduced	Debug Cost	Defects Detection Rate
Requirements	20	1	50%
Design	20	1	50%
Code	20	1	50%
Test	0	10	80%
Production	0	100	100%

The question (a) performs verification only at the Test Phase. Then errors are found only in Test phase at the percentage of 80%. The actual cost could be calculated like this(assuming the cost of correcting a defect before coding is 1).

Stage	Previous Bugs	Total Bugs	Detected	Solving cost
Requirements	0	20	0	0
Design	20	40	0	0
Code	20	60	0	0
Test	60	60	48	480
Production	12	12	12	1200

And the total cost is 480 + 1200 = 1680.

(b) Work out the cost of fixing all defects when verification is done at each of Requirements, Design, Code and Test phases. (8 marks)

Question (b) performs verification at Requirements, Design, Code and Test Phase. Then there are bugs found in every phases. The actual cost could be calculated like this.

Stage	Previous Bugs	Total Bugs	Detected	Solving Cost
Requirements	0	20	10	10
Design	10	30	15	15
Code	15	35	17.5	17.5
Test	17.5	17.5	14	140
Production	3.5	3.5	3.5	350

And the total cost is 10 + 15 + 17.5 + 140 + 350 = 532.5.

Question 4

A bug tracking tool offers many useful functions to the users to report, track, classify bug/defect reports, etc. Take a close look at Bugzilla and list 8 key functions of this tool that help with bug tracking. (8 marks)

In the offcial website of Bugzilla, it is descripted that the benefits of it as follows:

1. Improve communication

In my short experience, the web sever and email combined structure really helps the communication between users and developers. And also all the bugs and commets are numbered. The communicaion is really tracable by talking about bug 555, or the comment 9.

2. Increase product quality

Of cause, keeping bugs tracable and giving users the oppotunity to report bugs through the whole life cycle of the software improves the product quality greatly.

3. Improve customer satisfaction

Every error and question is recorded and treated seriously. This feature satisfy the need of customers.

4. Ensure accountability

Flexible structure and various ways of communication made Bugzilla a reliable system. It is a safe and mature bug tracing system.

5. Increase productivity

This can be seen in several different fields.

1. Every bug is clearly numbered and the history is logged. This saves lots of time finding previous work and avoid the confusions.
2. The status of every bug is clearly stated. The priority and risk level is shown by users' voting. And all corresponents are automatically added to the CC list. Changes on the bug status are sent to all people concerning it.
3. Bugzilla can adapt to multiple situations

The core service of Bugzilla is remaining flexible. Different databases or front ends can be added. It can be modified to suit new environment like small business.

The major functions behind all these features are listed below.

1. Searching function
2. User-configurable email notifications of bug changes
3. Full change history

4. Inter-bug dependency tracking and graphing
5. Attachment management
6. A robust, stable relational database management system back-end
7. Web, XML, email and console interfaces
8. Completely customisable and/or localisable web user interface
9. Extensive configurability

Question 5 (16 marks)

iMove Company is headquartered in Hong Kong and has a loyal customer base of over 2000 throughout Asia. It has its own distribution centers and maintains a fleet of delivery vehicles. Its information systems are all centrally maintained by the Information Systems (IS) staff.

The CEO of iMove wants to expand their customer base and reduce the cost of doing business. Each employee at iMove has internet access through the corporate hub. The most common use of the internet is for e-mail with sales staff (about 10) and other business partners (about 25). The company maintains a website that provides information on the company and its products.

The CEO has decided to implement a mobile solution for inventory management, which will provide its sale staff and business partners up-to-date data on its inventory any where any time. iMove has contracted out the development to your company. Write a system test plan for the system you are developing for iMove. The document's audience is mainly your internal test team and also the client manager.

The test plan should include:

- Test strategy
- Test objectives and scope
- Test phases
- Test schedule and milestones
- Entry criteria for each milestone's phase
- Tasks to be performed during each phase

- Deliverables and exit criteria for each Milestone
- Roles and responsibility
- Required testing resource

You will be graded on the following items:

1. Readability and organization of your test plan.
2. Thoroughness of test plan.

Page Limit: 12 pages.

Reference: IEEE Test Plan template