STUDENT’S DECLARATION OF ORIGINALITY

By submitting this online test, I declare that this submitted work is free from all forms of plagiarism and for all intents and purposes is my own properly derived work. I understand that I have to bear the consequences if I fail to do so.

Online Test Submission

Course Code: BACS 2103

Course Title: Software Quality and Assurance Testing

Signature: 

Name of Student: Lau Pin Jian

Student ID: 20WMR09285

Programme (eg. RSF, RSD) : RSD

Date: 23/11/2021

|  |  |
| --- | --- |
| **Question** | **Marks** |
| Question 1 |  |
| Question 2 |  |
| Total |  |

**Question 1**

1a) i)

Testing activities should be started as early as possible. Since the above scenario stated that, the tester used up most of the times on creating the test case instead of test the system. Therefore, the tester should start the testing while the system is during the system development period.

Exhausive testing is impossible. Since the above scenario stated that the tester would like to find more errors in the system by considering all inputs for every single function. However, this is not acheiveable for him as he has reached the timeframe of the testing periods. Therefore, the tester should priotize testings based on risk analysis.

The defect clustering. Since the above scenario stated that during the acceptance test, the users found that bugs less frequently found in one or two modules of the system but lot of simple errors were found in most of the connected modules. As the defects are not evenly distributed, although the there were less bugs found in one or two modules doesn’t mean that less bugs will be found in the connected modules.

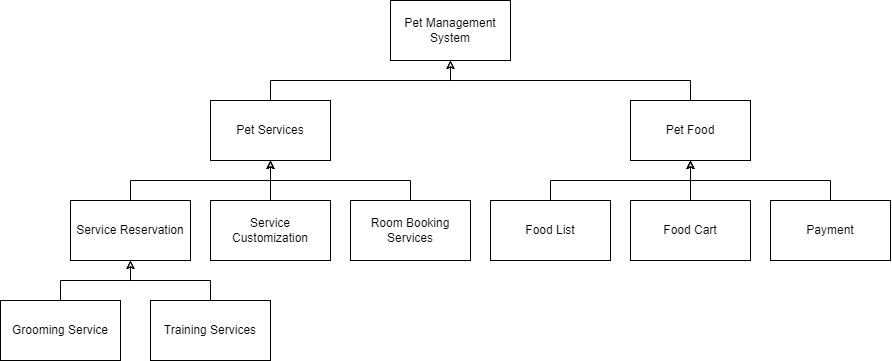
1a) i)

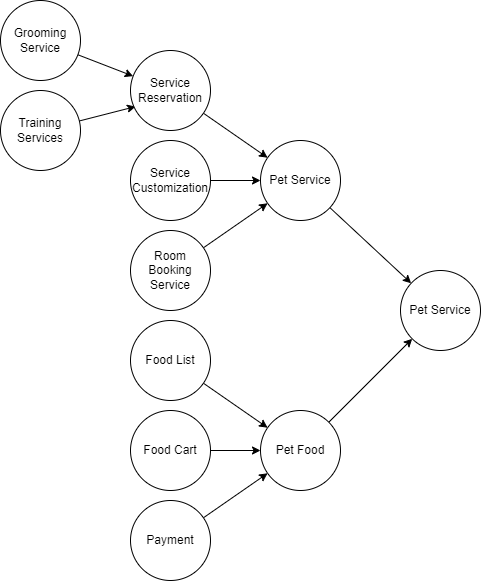
First reason is the programmers are optimistics. It is always a good idea for the tester to test its own software is because the tester have better understanding of the processes. Somehow, they may bring unconscious or unintentional bias into the testing process. Due to the programmers lack of objectivity to be able to test their work. As the programmers may find it harder to put themselves into an end user’s while during the testing the period.

Second reason is the programmers may lack of time during the development process. The testing requires hard work, laser-sharp focus and most of all, dedicated time. Assume the programmes have to do their role such as writing code which takes lots of time, they also have to create strategies and plans, write documentations even conduct unit tests. After all of this, the programmes can’t have full capacity to conduct complex testings anymore.

Third reason is the programmers may lack of experiences when it comes to software testing. Unlike the QA tester, they have vast experience on the testing software and testing processs since they have the capabilities of grasping fully and master the concepts. The programmers may have the capabilities to solve the broken functionalities but not to break an application. Therefore, regarding of less experience, QA testers are most suitable for the testing.

1b) i)





Bottom-up testing is a type of incremental integration testing approach in which testing is done by intergrating or joining two or more modules by moving upwards from bottom to top. In these scenario, the lower modules such as Grooming services and Training services are tested first, then combine modules as a whole module and be tested again for the next level.

Advantages of bottom-up testing are the observation of the test results is easier compare to top-down strategies. Moreover, the test conditions are easier to create with I/O functions.

1b) ii)

System testing is to compare the system or program to its original objectives. For example: -

* Reliability testing, since the system have a payment modules which requires high uptime. Therefore, the system would not easily breakdown and affect the end users that they unable to make payment.
* Performance testing, since the system have payment modules, the end users information is highly confidential. Security testing is to ensure the information of the end users would not be leak when a security breaches were happened.

Acceptance testing is a process of comparing the program to its initial requirements and the current needs of its end users. For example :-

* Alpha test – software is tested unde the supervision
* Beta test – software is tested by customers at his own site without supervision

**Question 2**

2) a)

Module Testing (Non-incremental approach)

- Testing each modules independently, then combining the modules to be tested as a whole program.

2) b)

**Correct** - requirements should define the stakeholders’ needs

**Understandable** – requirements should be easy to be understand by the stakeholders

**Clear and unambiguous** – requirements should be clear enough and unambiguous, which is only one way to interpret the requirement.

Code Review

* Evaluate the code structure together with tester
* Check on the standard and procedues
* To ensure the consistency, compliance to standard and the quality of comments

Code walkthrough

* Normally done in a single meeting and less formal.
* Evaluate the software to:
  + Find anomalies & improve the software product.
  + Consider alternative implementations.
  + Evaluate the conformance to standard and specification.
* Tester and programmers (group of programmers) are present.
* Each test case is mentally executed to inquire about the programmer’s logic and assumptions.

Desk Checking

* Desk checking is the primary testing done on the code.
* Static checking will be done by programmers before compilation or execution.
* If any error is identified, it is going to check by author and will correct the code
* The code is compared with requirements specification or design to see that the designed code is according to client requests.

2) c)

The pesticide paradox. The previous input combinations of test cases are used but the testers unable to find any defects and claims that the function is performing well. As the input combinations were used was same as the previous and the code was reused but still using the same combinations therefore the “invisible” defect was unable to be found.

Testing is context dependent. Testing must be adapted to the risks inherent in the use and environment of the application. No two systems should be tested in the exactly same way. The intensity of testing, test exit criteria should be decided respectively for every software system, depending on its usage environment.

2) d)

Moderator

* The leader of inspection whom plans the inspection and coordinates with the author
* For example, performs the entry check and the follow-up on the rework.

Author

* Describes the background of the work product
* Makes changes to the work product according to the action plans
* To ensure everythings is correct

Specialist

* Provide technical advises and support

Designer,

* show design templates for reviewing