Singapore Institute of Management
BSc in Computing & Information Systems
CIS 210 Tutorial 6 Suggested Solution

[Note: The answers are suggested solutions for your reference. They are not to be treated as model answers but rather a guide on how you might answer the questions]

1. What is the difference between verification and validation? Why is validation so difficult to achieve?

   Answer :

   Verification is to check whether the product is being built right. Need to ensure that the software meets the expectations of the customer.
   Validation is to check whether we are building the right product. Need to check that the software conforms to its specified functional and non-functional specification.

   Validation is so difficult to achieve because it goes beyond checking that the system conforms to its specification. To showing that the software does what the customer expects as distinct from what has been specified. Early validation of software is important. It is easy to make errors and omissions in the system's requirements and in such cases, the final software will probably not meet its customer's expectation. Also, requirements validation is unlikely to discover all requirements and some flaws and deficiencies in the requirements can sometimes only be discovered when the system implementation is complete.

2. What are the different types of maintenance tasks? Explain for the following, which type of maintenance task you will be applying. Justify with reasons.
   a) The company's human resource system is currently running on the HP platform with Unix Version 5.0. HP has informed that there are new versions of the Unix for the HP platform, they will cease to support version 5.0 by end of the year.
   b) The mainframe systems in the Awesome Bank has been around for 20 years now. The systems are getting more and more difficult to maintain as a number of the experienced staff have left and the documentation on the systems have not been updated with the changes in the system. These core systems are essential to the operation of the Bank and on-going changes are still required to meet the business needs.
   c) ABC Training has an internet-based system allowing individuals to study their courses in their own time. They will like to enhance the system to allow the individuals to submit their assignments as well for the supervisors to view and mark and return.

Answer :

The types of maintenance tasks include :

- *Corrective – to remove the defects that survive the testing process*
- *Adaptive – to adjust the software to changes in its hardware and software environments as well as to changes in requirements*
- *Perfective – to incorporate new functions or to improve existing one.*
- *Preventive – to make future maintenance events more viable and less costly consists of reverse engineering and re-engineering.*

a) Adaptive – as the Operating system has to be upgraded and the software has to be adapted to the new operating system
b) Preventive – as the Bank systems need to be undergo preventive maintenance to improve maintainability through reverse engineering or re-engineering or both.
c) Perfective – need to incorporate new function into existing software to meet new business need.

3. Explain which are the 4 stages of testing and which stage of system development it corresponds to. Justify which stage of testing is conducted by the user.

Answer :

4 stages of system development and testing include :

a) *Unit Testing corresponds to Coding stage*
   - as each module is being developed, it is tested in turn and in isolation from the others. Uses white box testing.

b) *Integration Testing corresponds to Design*
   - as Unit tested modules are successively added to an assembled integrated configuration in a stepwise manner until the entire software is assembled. Uses more of black-box testing and less of white box testing.

c) *Validation Testing corresponds to Requirements Analysis* ***(conducted by user)***
   - *The validation criteria elicited along with software requirements are applied to the integration-tested software. Uses black-box techniques exclusively.*
   - *This stage of testing is conducted by user as the users conduct acceptance testing for the system to validate that the system performs according to specifications.*
   - *2 types of testing – alpha testing where tests is in a controlled environment under close scrutiny of the developers who record errors and difficulties. Beta tests by users in actual environment where the software will operate and up to end-users to record and report problems encountered.*

d) *System Testing corresponds to Systems Engineering*
   - Concerned with testing the entire computer-based system. One of the main concerns is to test the interfaces between software, hardware and human components.

4. What is the difference between Tool Bridges and Integrated Project Support Environments types of CASE Tools? If you are the IT Manager, which CASE Tool will you buy? Justify with reasons.

Answer :

Individual Tools all the software engineer to do 1 task only and do not communicate with other tools. There is no link to the Project's central store of information. Tool Bridges rely on the fact that such individual tools dump information into a central store, which allows other tools to retrieve the information. The Individual tools communicate only through data exchange mechanism.

IPSE – is also based around a central store. However it also has standards for portability and interoperability. Thus developers using individual tools that adopt the standards can count on the tools being capable of full, loosely coupled integration with other individual tools that adopt the standards.

As IT Manager will choose to buy IPSE as the IPSE tends to use standards defined by industry-wide consultation and are freely available to developers and users. As such, the best tools for each task, tends to be available and can be purchased without paying for a penalty for integration. With Tool Bridges, additional work needs to be done to allow 1 tool to retrieve the data from a data store that is used by another tool or vice versa.

Also IPSE is based on open standards, and thus the company will not be locked in due to use of proprietary tools. Easy to change tools if the tools are found not to meet needs.

5. For the following, explain whether you will use white box or black box testing. Justify with reasons.
   - You have been assigned to code a program to validate the customer's account number.
   - The individual modules for the Funny game have been unit tested and are now ready for integration testing.
   - You are to be involved in the User Acceptance testing of the Income Tax Filing system.

Answer :

a) White Box testing – to test that the program is working correctly
- *need to test the various paths through the system whether they are performing correctly*
- *Error handling paths are tested*

b) A mixture of white box testing and black box testing but mainly the latter – testing to ensure that the modules are working correctly together
- *need to test the interfaces among the modules are performing correctly*
- *That the error handling is correct*
- *That the system performs correctly for the various functions to be tested*
- *Need to make sure that the inputs to the system produces the expected outputs*
- *To identify missing/incorrect functions*
- *Behaviour or performance errors*

c) Black box testing – mainly testing system functionality against specified system functionality
- *That the system performs correctly for the various functions to be tested*
- *Need to make sure that the inputs to the system produces the expected outputs*
- *To identify missing/incorrect functions*
- *Errors in data structures or external database access*
- *Initialization and termination errors*
- *Behaviour errors*

6. Develop simple coding standards (5 or more) for a system that you will develop for company ABC. The system is to be developed in Pascal. Give examples.

Answer :

Coding standards :

a) Where possible, spell identifiers in full capitalizing the first letter of each word. For e.g. HorizontalVelocity
b) Have Header comments for each function consisting of
   ii. A statement describing the function
   iii. Interface description containing descriptive statement for every argument and sample calling sequences
   iv. A list of subordinate modules
   v. Special points to note
   vi. Development history

E.g.

Function Name : ValidateDate
Function Purpose : To validate the date
Function Interface : ValidateDate(InDate : Date) : Boolean
Where Indate is the date to be checked.
Function returns True if the date is valid. False If otherwise
Subordinate : CheckDateFormat(InDate)
Restrictions on use : None
Development History : Amended on 27/6/1999 by ABC – to cater for year 2000

c) Do not use tabs for indentation. Use 4 spaces for indentation.
d) Declare 1 data item per line
e) Avoid testing negative conditions e.g. If Not (a=b) should be avoided.