

Ali Haider

14259

Sessional Assignment

Course: Object Oriented SE

Instructor: Mam Sana Jahan

Date: June 20, 2020

Question #1 (10 Marks)

In Software Engineering, there is not a single answer to the question “What should be done first, Coding or Modeling?”. Elaborate different scenarios in which all the answers to this questions are justified.

Answer #1: In Software Engineering , Modeling should be done first because

- In most of the cases it is found suitable because changing the models is quite easy then to change the coding.
- It gives a prototype of the product to the customer, so they can change the product according to his need if he/she wanted to.
- By the help of modeling you can easily cope-up with the requirement engineering.
- It is good for testing and other modules to keep modeling before the coding.
- It doesn't effect the actual product

One of the basic notions of the software development process is SDLC models which stands for Software Development Life Cycle models. SDLC - is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. There is no one single

SDLC model. They are divided into main groups, each with its features and weaknesses.

Evolving from the first and oldest “waterfall” SDLC model, their variety significantly expanded. The SDLC models diversity is predetermined by the wide number of product types – starting with a web application development to a complex medical software. And if you take one of the SDLC models mentioned below as the basis – in any case, it should be adjusted to the features of the product, project, and company. The most used, popular and important SDLC models are given below:

- Waterfall model
- Iterative model
- Spiral model
- V-shaped model
- Agile model

Question #2 (10 Marks)

When carrying out Testing of a Software, a number of techniques are used. Why are they so many in number? Name a few popular Testing Techniques in Software Engineering and state the importance of each one.

Answer #2:

Software Testing Techniques help you design better test cases. Since exhaustive testing is not possible; Manual Testing Techniques help reduce the number of test cases to be executed while increasing test coverage. They help identify test conditions that are otherwise difficult to recognize.

A few popular Testing Techniques are:

- Boundary Value Analysis (BVA)
- Decision Table based testing.
- Error Guessing

Boundary Value Analysis (BVA)

Boundary value analysis is based on testing at the boundaries between partitions. It includes maximum, minimum, inside or outside boundaries, typical values and error values.

It is generally seen that a large number of errors occur at the boundaries of the defined input values rather than the center. It is also known as BVA and gives a selection of test cases which exercise bounding values.

This black box testing technique complements equivalence partitioning. This software testing technique base on the principle that, if a system works well for these particular values then it will work perfectly well for all values which comes between the two boundary values.

Guidelines for Boundary Value analysis

- If an input condition is restricted between values x and y, then the test cases should be designed with values x and y as well as values which are above and below x and y.
- If an input condition is a large number of values, the test case should be developed which need to exercise the minimum and maximum numbers. Here, values above and below the minimum and maximum values are also tested.
- Apply guidelines 1 and 2 to output conditions. It gives an output which reflects the minimum and the maximum values expected. It also tests the below or above values.

Decision Table Based Testing.

A decision table is also known as to Cause-Effect table. This software testing technique is used for functions which respond to a combination of inputs or events. For example, a submit button should be enabled if the user has entered all required fields.

The first task is to identify functionalities where the output depends on a combination of inputs. If there are large input set of combinations, then divide it into smaller subsets which are helpful for managing a decision table.

For every function, you need to create a table and list down all types of combinations of inputs and its respective outputs. This helps to identify a condition that is overlooked by the tester.

Following are steps to create a decision table:

- Enlist the inputs in rows
- Enter all the rules in the column
- Fill the table with the different combination of inputs
- In the last row, note down the output against the input combination.

Error Guessing

Error guessing is a software testing technique which is based on guessing the error which can prevail in the code. It is an experience-based technique where the test analyst uses his/her or experience to guess the problematic part of the testing application.

The technique counts a list of possible errors or error-prone situations. Then tester writes a test case to expose those errors. To design test cases based on this software testing technique, the analyst can use the past experiences to identify the conditions.

Guidelines for Error Guessing:

- The test should use the previous experience of testing similar applications
- Understanding of the system under test
- Knowledge of typical implementation errors
- Remember previously troubled areas

- Evaluate Historical data & Test results

Conclusion

- Software testing Techniques allow you to design better cases. There are five primarily used techniques.
- Boundary value analysis is testing at the boundaries between partitions.
- Equivalent Class Partitioning allows you to divide set of test condition into a partition which should be considered the same.
- Decision Table software testing technique is used for functions which respond to a combination of inputs or events.
- In State Transition technique changes in input conditions change the state of the Application Under Test (AUT)
- Error guessing is a software testing technique which is based on guessing the error which can prevail in the code.

Instructions for Assignment Submission

1. Write your names and Ids at the top of each paper of answer sheet.
2. Scan / Take Photo of each paper and save each photo with a number. E.g. photo of page 1 of answer sheet be saved with name 1.jpg, then 2.jpg and so on.
3. Put all answer photos in a folder, name the folder with your Roll Number, Name and Subject Name, e.g. “11512 - Sanaa Jeehan - OOSE”. Alternately, you can also make a PDF file of all the pictures and name it as explained.
4. Zip the folder and upload.