

SOFTWARE ENGINEERING

(MC – 310)

ASSIGNMENT – 5

AIMAN SIDDIQUA

2K18/MC/008

Ques. Explain reverse engineering and re-engineering. What are the principal factors that affect the costs of system re-engineering?

Solution. Software Re-Engineering is the examination and alteration of a system to reconstitute it in a new form. The principles of re-engineering when applied to the software development process is called software re-engineering. It affects positively at software cost, quality, service to the customer and speed of delivery. In Software Re-engineering, we are improving the software to make it more efficient and effective.

Reverse engineering is a process of design recovery. Reverse engineering tools extract data, architectural and procedural design information from an existing program.

Re-Engineering cost factors:

- The quality of the software to be re-engineered.
- The tool supports availability for engineering.
- The extent of the data conversion which is required.
- The availability of expert staff for Re-engineering.

Ques. What is software maintenance? Explain why a software system that is used in a real-world environment must change or become progressively less useful.

Solution. Software Maintenance is the process of modifying a software product after it has been delivered to the customer. The main purpose of software maintenance is to modify and update software applications after delivery to correct faults and to improve performance.

A software system that is used in real world environment must change because of the following reasons:

- **Market Conditions** - Policies, which changes over the time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger a need for modification.
- **Client Requirements** - Over the time, customers may ask for new features or functions in the software.
- **Host Modifications** - If any of the hardware and/or platform (such as operating system) of the target host changes, software changes are needed to keep adaptability.
- **Organization Changes** - If there is any business level change at client end, such as reduction of organization strength, acquiring another company, organization venturing into new business, need to modify in the original software may arise.

Ques. Write short notes on

(a) Alpha and Beta testing

Solution.

ALPHA TESTING is a type of acceptance testing, which is performed to identify all possible bugs/issues before releasing the product to the end-user. Alpha test is a preliminary software field test carried out by a team of users to find out the bugs that were not found previously by other tests. Alpha testing is to simulate a real user environment by carrying out tasks and operations that actual users might perform.

BETA TESTING is a type of acceptance testing; it is the final test before shipping a product to the customers. Beta testing of a product is implemented by "real users "of the software application in a "real environment." In this phase of testing, the software is released to a limited number of end-users of the product to obtain feedback on the product quality. It allows the real customers an opportunity to provide inputs into the design, functionality, and usability of the product.

(b) System testing and debugging,

Solution.

Software testing is a process of identifying defects in the software product. It is performed to validate the behavior of the software or the application compared to requirements. In other words, we can say that the testing is a collection of techniques to determine the accuracy of the application under the predefined specification but, it cannot identify all the defects of the software.

Debugging is the action where the development team or a developer implements after receiving the test report related to the bugs in the software from the testing team. In the software

development process, debugging includes detecting and modifying code errors in a software program.

In the debugging process, the developer needs to identify the reason behind the particular bug or defect, which is carried out by analyzing the coding rigorously.

(c) Functional testing

Solution. Functional Testing is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification. In functional testing, each function tested by giving the value, determining the output, and verifying the actual output with the expected value. Functional testing performed as black-box testing which is presented to confirm that the functionality of an application or system behaves as we are expecting. It is done to verify the functionality of the application.

Functional testing is also called as black-box testing, because it focuses on application specification rather than actual code. Tester has to test only the program rather than the system.

(d) Structured and unstructured maintenance.

Solution. Unstructured maintenance wades straight into the source code and makes changes based on that alone. Structured maintenance examines and modifies the original design, and then reworks the code to match it. Clearly structured maintenance is a more reliable and (usually) a more efficient process. Unfortunately, it's not always possible.

Ques. What are the objectives of testing? Explain why testing can only detect the presence of errors, not their absence. What are the different levels of testing and the goals of the different levels?

Solution. Software testing is an activity which aims at evaluating the quality of a software product and also to improve it by identifying defects. Software testing strives to achieve its objectives but has certain limitations. However, adherence to the established objectives ensures effective testing.

Objectives of software testing:

- To check whether software which builds, it is as per the requirement or not.
- Finding defects from the software before customers find them out.
- Defects get a fix from the developer.
- Preventing defects
- Gaining confidence about the level of quality

Testing is a part of a broader process of software verification and validation. It consists of a set of activities, where the testers try to make the software behave anomalously in order to detect an anomaly to be later fixed. Testing cannot demonstrate the faults other than specified in every circumstance. It is always possible that an overlooked test case could conceal further problems with the system.

There are mainly four Levels of Testing in software testing:

- **Unit Testing:** checks if software components are fulfilling functionalities or not.
- **Integration Testing:** checks the data flow from one module to other modules.
- **System Testing:** evaluates both functional and non-functional needs for the testing.
- **Acceptance Testing:** checks the requirements of a specification or contract are met as per its delivery.

Ques. Suggest five possible problems that could arise if a company does not develop effective configuration management policies and processes.

Solution. Five possible problems that could arise if a company does not develop effective configuration management policies and processes are:

- New versions of software systems cannot be created effectively as they change. Developers cannot keep track of the changes to the software.
- Controlling the costs and effort involved in making changes to a system is difficult.
- Wrong version of a system may be delivered to the customers or forget where the software source code for a particular version of the system or component is stored.
- If someone leaves the company, protecting investments in software and the ability to reproduce a build with the correct components or continue development on a project is difficult.
- Ineffective quality management process because configuration management may be seen as part of a more general quality management process.