

Q2.1 Explain why testing can only detect the presence of errors, not their absence.

When a software is created, it runs through a process of testing. Testing is actually a set of activities that focus on ~~det~~ detect software abnormality in a way that the errors can be discovered and corrects. The aim of software testing is to observe if the software behavior meet the requirements that they are expecting. Testing demonstrates to the developer that the software fullfils its requirement and it is a way to find out if it behaves in a incorrect or different way from the specifications.



Q.2.2 Define the following.

1. Unit Testing :-

It is a level of software testing where individual units of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or few inputs and a single output.

2. System Testing :-

System testing is a black box technique performed to evaluate the complete system. In system testing, the functionalities of the system are tested from an end-to-end perspective.

3. Black box Testing :-

It is defined as a testing technique in which functionality of the Application Under Test is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based



M. Zakria Salem

Roll. 14459

entirely on software requirements and specifications. In Black box Testing we just focus on inputs and outputs of the software system.

#### 4. White box Testing:-

It is testing of a software solution's internal structure, design and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability and security. White box testing is also known as clear box testing, Open box testing, transparent box testing, glass box testing. It is usually performed by developers.



Q3.1 Briefly describe the three main types of software maintenance. Why is it sometimes difficult to distinguish between them?

The three main types of software maintenance are:

- a) Corrective maintenance or fault repair.
- b) Adaptive maintenance or environmental adaptation.
- c) Perfective maintenance or functionality addition.

a) Corrective maintenance:-

It is concerned with fixing errors that are observed when the software is in use.

b) Adaptive maintenance:- Adaptive maintenance

is concerned with the change in software that takes place to make the software adaptable to new environment such as to run the software on a new operating system.

c) Perfective maintenance:-

It is concerned with the change in the software that occurs while adding new functionalities in the software.



M. Zakria Salem

Roll. 14459

Q3.2 What are the principal factors that affect the costs of system re-engineering? Also briefly explain the re-engineering process with the help of diagram

i. The quality of the software to re-engineered:

The lower of the quality of the software and its associated documentation the higher re-engineering costs.

ii. The tool support available for re-engineering:

The use of case tools to automate most of the program changes is normally cost affective to re-engineer a software.

iii. The extend of data conversion required:

If re-engineering requires large amount of data to be converted, this significantly increases the process cost.

iv. The availability of expert staff:-

if the staff is responsible for maintaining the system can't be involved



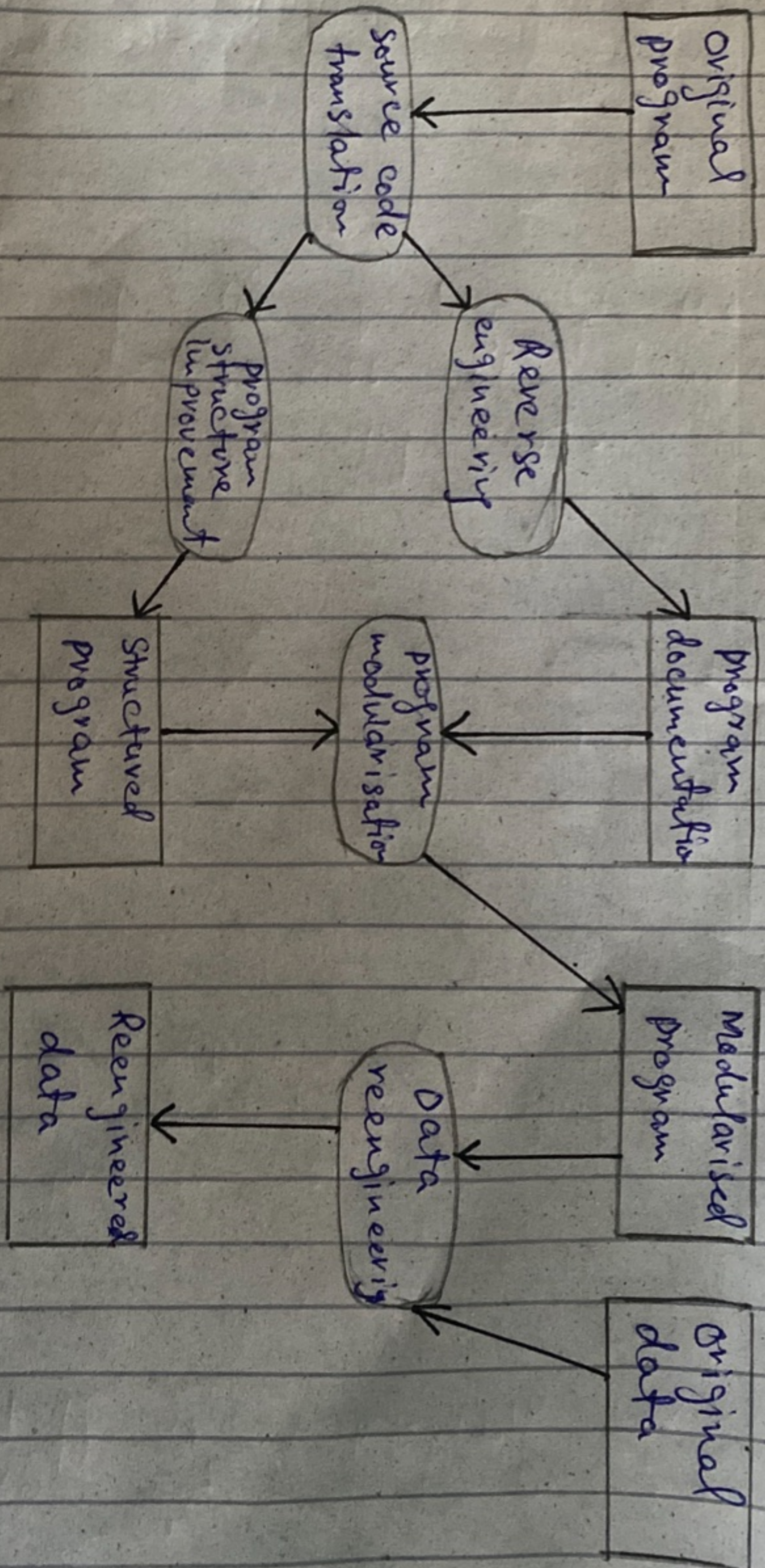
M. Zakria Salem

Roll. 14459

in the re-engineering process, this will increase the costs. System re-engineers will have to spend a great deal of time understanding the system.



# The Re-engineering Process





M. Zalaria Salem

Roll. 14459

Q11:

Context Diagram

