

NAME HAMZA IQBAL

I . D 14784

INSTRUCTOR Ghassan Husnain 7

Program B . S Software

Subject Software Engineering

SECTION B



## Question NO. 2

Define the following terms:-

(i) Unit Testing

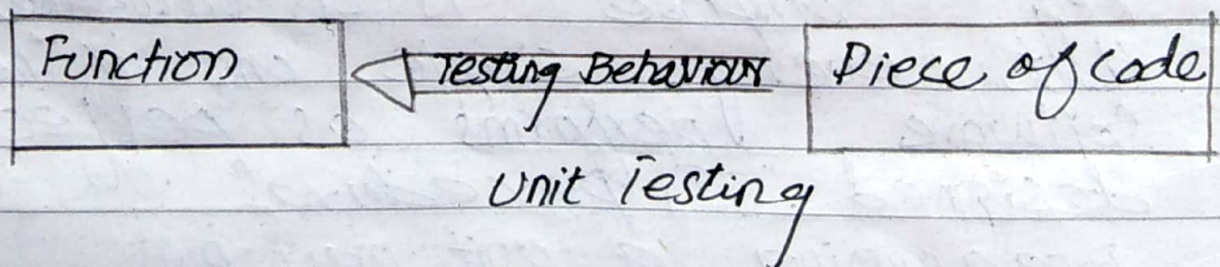
Unit Testing :-

Unit testing is a level of software testing that involves individually testing unit of code to ensure that it works on its own, independent of the other units. Their key purpose is to validate that every single unit of the software performs as perfectly designed. In procedural programming, a unit may be an individual function, programming, process, etc, whereas a technique, which may belong to a base/super class, abstract class or child/derived class.

Unit testing frameworks, stubs, drivers, and mock/fake object are used to help out in unit testing. In short, unit testing is an act used to validate that



separate unites of source code work properly. Example :-  
A function, loop, method or statement in the program is working very well. This type of testing is executed by the developer. In unit testing, individual procedures or functions are tested to guarantee that they are operating in the approved manner and all components are tested individually.





# System Testing :-

Definition :-

System testing is defined as testing of a complete and fully integrated software product. This testing falls in black-box testing wherein knowledge of the inner design of the code is not a pre-requisite and is done by the testing team.

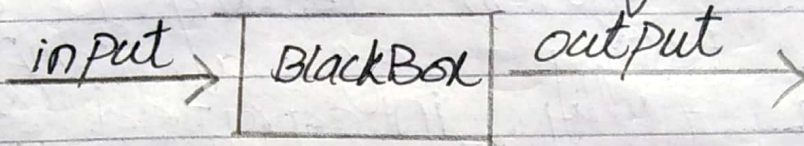
## \* Black Box Testing \*

Black-Box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing. Unit, integration, system and acceptance. It is sometimes



referred to as specification-based testing.

## Black Box Systems



System

Black box · oracle machine

Methods and techniques

Black-box testing · Black boxing

Related techniques

Feed forward · obfuscation

Pattern recognition white

box · white box testing

System identification

Fundamentals

A priori information · Control

systems · open systems

operations research

Thermodynamic systems.



## White Box Testing

White Box testing is a method of software testing that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing).

In white box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the expected outputs. This is analogous of testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system level of the software testing process.

Although traditional testers tended to think of white box testing as being done at the unit level, it is used for integration



and system testing level of the software testing process.

Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration and between subsystems during a system-level test.

## QUESTION NO 2

### PART II

Explain why testing can only detect the presence of errors, not their absence?

⇒ Testing can detect only the presence of



errors, not their absence because the main goal of the testing is: to observe the behaviour of the particular software and to check whether it meets its requirement expectation or not.

Testing is a part of broader process of software verification and validation. It consists of a set of activities where the tester try to make the software behave anomalous in order to detect or anomaly to be later fix. Testing cannot demonstrate the faults other than specified in every circumstance. It is always possible that a test have overlooked could discovered further problem with the system.

---



## Question      No      3

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

⇒ The main types of software maintenance are as follows:

(i) Fault repairs:

Coding errors are usually relatively cheap to correct. Design errors are more expensive as they may involve re-writing several program components. Requirements errors are the most expensive system redesign which be necessary.

ii Environmental adaptation:-

This type of maintenance is required when some aspect of the system's environment such as the hardware, the platform operating system or other support software changes the application system must be modified to adapt



it to cope with these environmental change.

ii) Functionality addition :

This type of maintenance is necessary when the system requirement change in response to organizational or business change, the scale of the other types of maintenance.

Why is it difficult to differentiate between the types of maintenance?

In practice, there is not a clear-cut distinction between these types of maintenance when the system adapt to new environment, then add functionality to take advantage of new environmental features. Software faults are often exposed because users use the system in unanticipated ways. These types of maintenance are recognized but a different person sometimes give



them different names.

"Corrective maintenance" is universally used to refer to maintenance for fault repair.

"Adaptive maintenance" sometimes means adapting to new environment and sometimes means adapting the software to new requirements.

"Perfective maintenance" sometimes means perfecting the software by implementing new requirements in other cases it means maintaining the functionality of the system but improving its structure and performance.

## Question \* NO \* 3

### Part II

What are the principal factors that affect the costs of system reengineering?

Also briefly explain the reengineering process with the help of diagram.



## Software Re-Engineering:-

Software Re-Engineering is the examination and alteration of a system to reconstitute it in a new form.

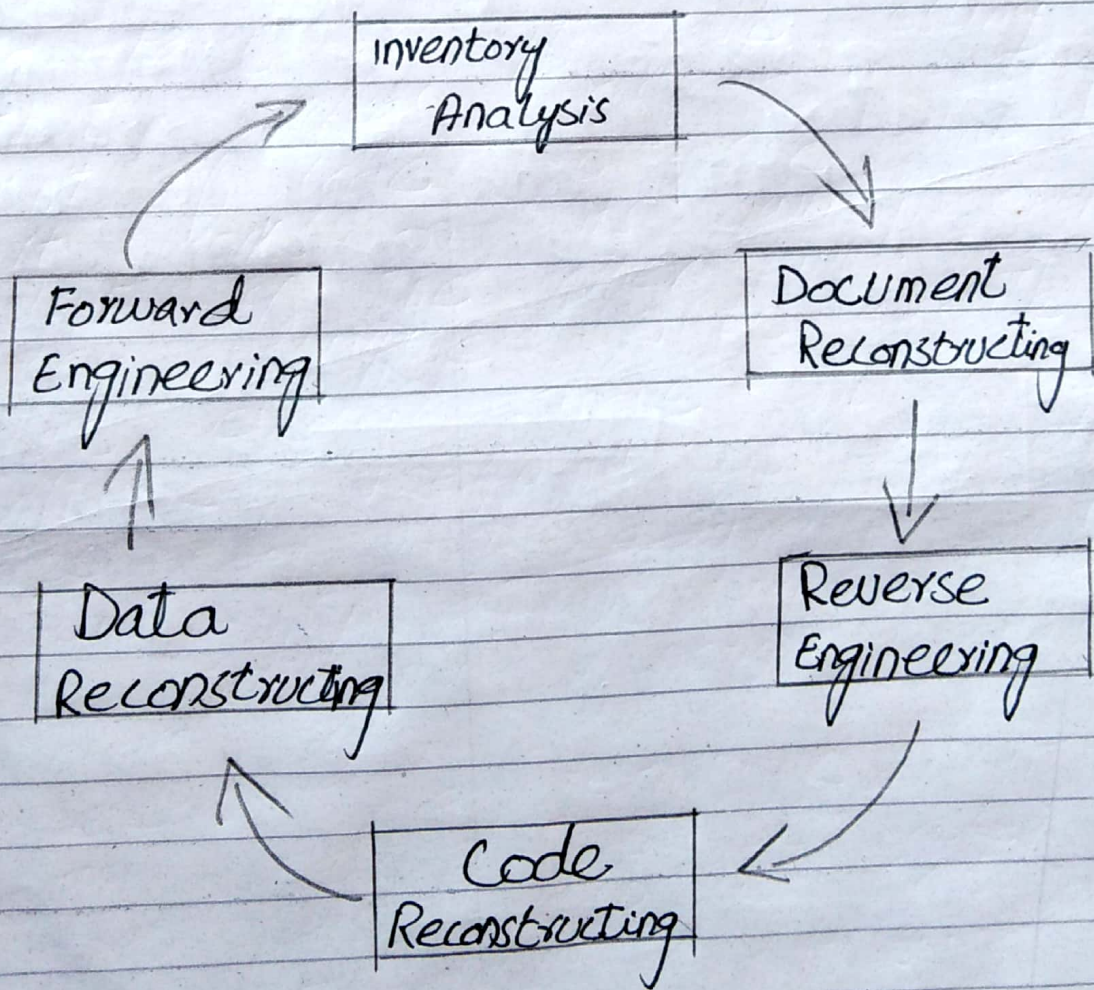
The principles of Re-Engineering when applied to the software development process is called software re-engineering. It affects positively at software cost, quality, service to the customer and speed of delivery. In software Re engineering, we are improving the software to make it more efficient and effective.

### Re-Engineering Cost factors:-

- The quality of the software to be re-engineered.
- The tool support availability for engineering.
- Extent of the data conversion which is required.

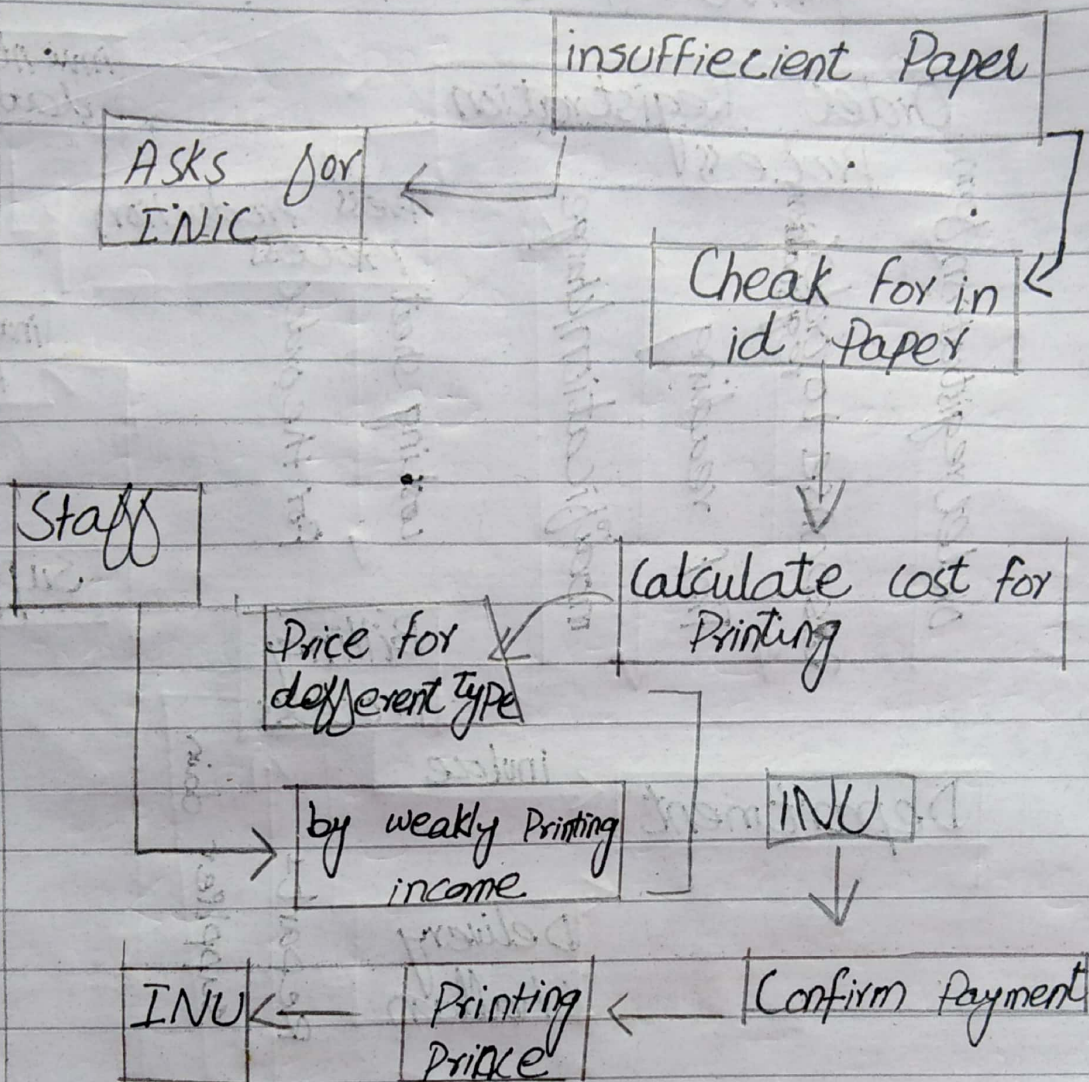


- The availability of expert staff for V&E engineering.

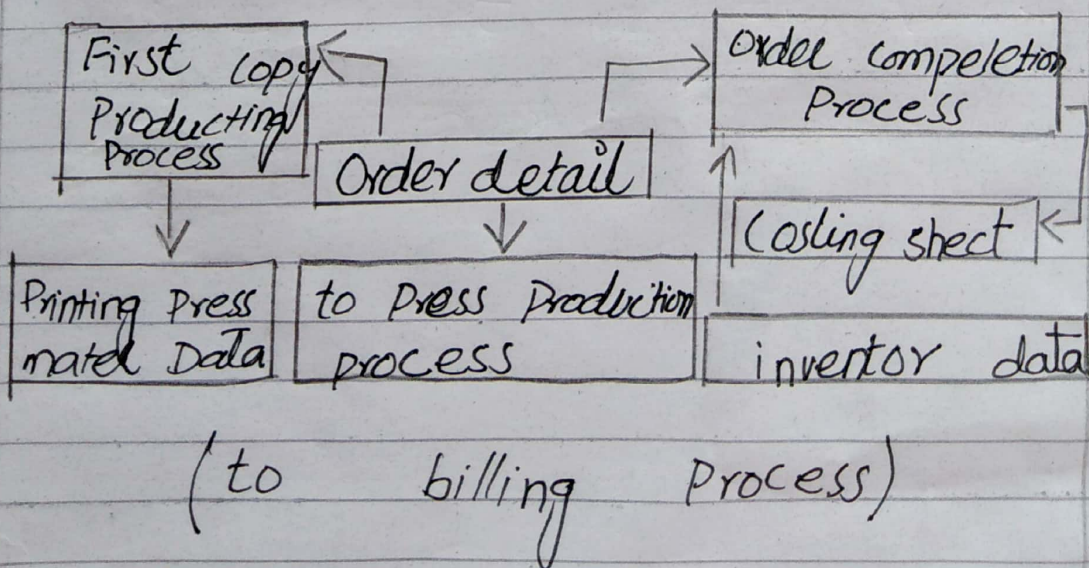




# Question 2 01.1



# Question # 1.2





Q#NO#1.3

