

NAME # AWAIS MALIK

ID # 14741

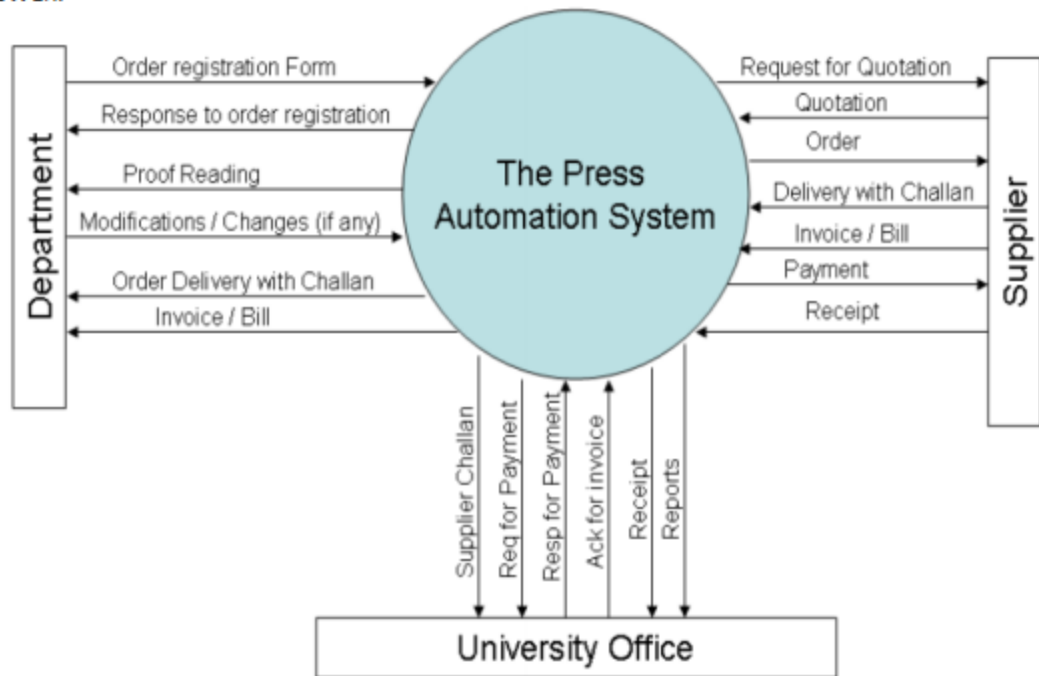
DAPARTMENT # SE

SECTION # B

PAPER # SE

Q1.1

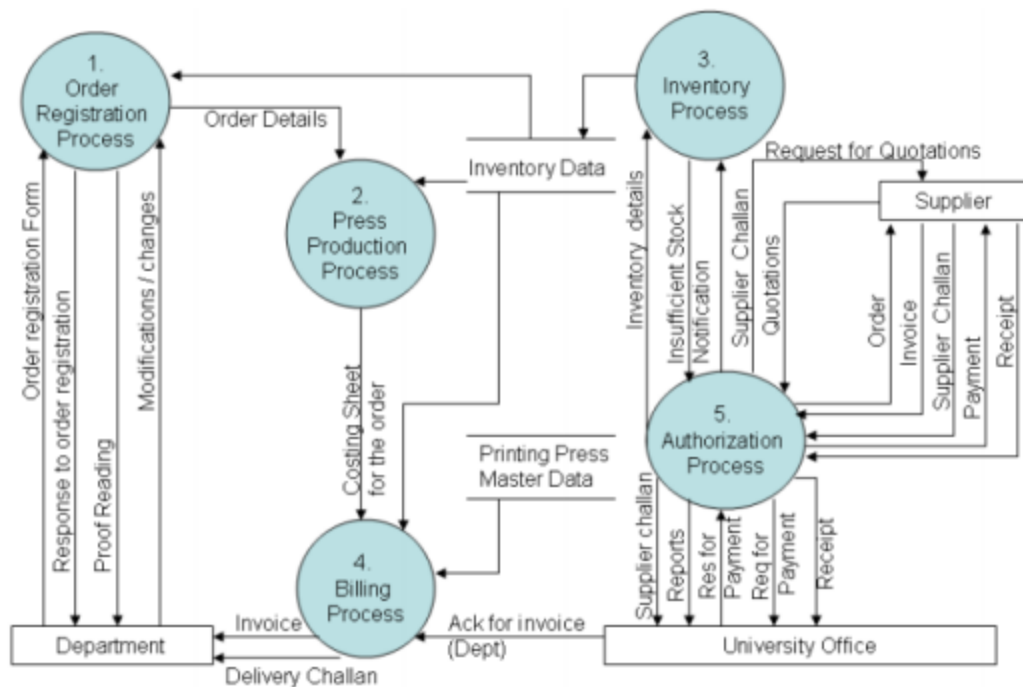
ANSWER:



CONTEXT DIAGRAM

Q1.2

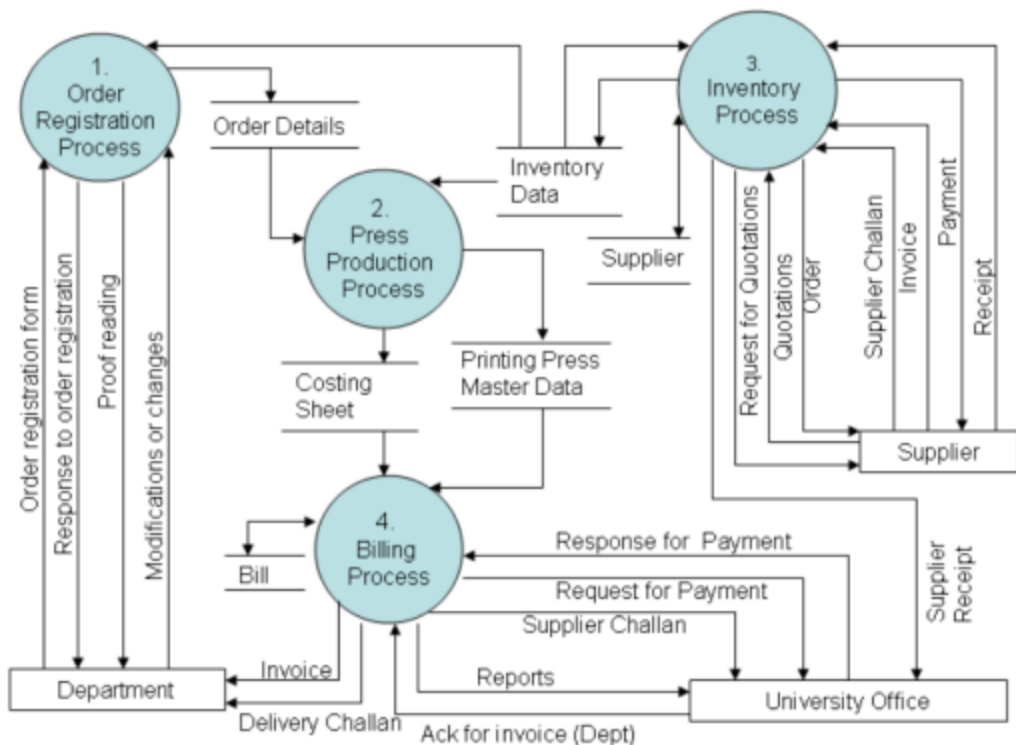
ANSWER:



Level 1 Data Flow Diagram (DFD)

Q1.3

ANSWER:



Level 2 Data Flow Diagram (DFD)

Question 2:

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

Answer:- Testing can only detect the presence of errors, not their absence because the main goal of testing is to observe the behavior of 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 testers 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 discover further problem with the system -

Assume that exhaustive testing of a program - where every possible valid input is checked is impossible (true for all but trivial programs) Test Cases either do not reveal a fault in the program or reveal a program fault. if they reveal a program fault then they demonstrate the presence of an error. if they do not reveal a fault, however this simply means that they have executed a code sequence that - for the inputs chosen is not faulty. The next test

of the same code sequence with different inputs could reveal a fault.

Q2.2 Define the following term.

1) Unit Testing:-

- * The most micro scale of testing.
- * Tests done on particular functions or code module.
- * Requires knowledge of the internal program design and code.
- * Done by programmers (not by testers)

2) System Testing:-

System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

Black box testing :-

- + No knowledge of internal design or code required -
- + Tests are based on requirements and functionality
- + Not based on any knowledge of internal design or code
- + Test are data driven (Tests are based on putting some data to check the system) -
- + It uncovers
 - incorrect or missing functions
 - interface errors
 - Errors in data structures or external database access
 - Performance errors -
 - initialization and termination errors -

White box testing:-

* Based on knowledge of internal logic of an application Code

* Base on coverage of Code Statements branches paths, Conditions

* Test are logic driven-

* it ensures

- All independent paths within a module have been exercised at least once

- Exercise all logical decisions on their true and false

Sides-

- Execute all loops at their boundaries and within their operational bands-

- Exercise internal data structures to ensure their validity

Question #3 :-

Q3.1)

Answer

Types of Software maintenance:-

* Maintenance to repair software faults:-

Changing a system to correct deficiencies in the way meet its requirement- Requirements errors are the most expensive to repair because of the expensive system redesign which be necessary -

* Maintenance to adapt software to a different operating environment:-

Changing system so that it operates in a different environment (computer, OS, etc.) from its initial implementation

The
This type of maintenance is required when some aspect of the systems 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 changes.

- * Maintenance to add or modify the system's functionality:-
- * Modifying the system to ~~st~~ satisfy new requirements.

This type of maintenance is necessary when the system requirements change in response to organizational or business change. The scale of the changes required to the software is often much

much greater than for the other types of maintenance.

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

in practice there is no 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 fault are often exposed because users use the the system in Unanticipated ways. These types of maintenance are recognized but a different person sometime gives them different names.

Q3.2,

Answer :-

Reengineering Cost factors :-

* The quality of the software to be reengineered.

* The tool support available for reengineering.

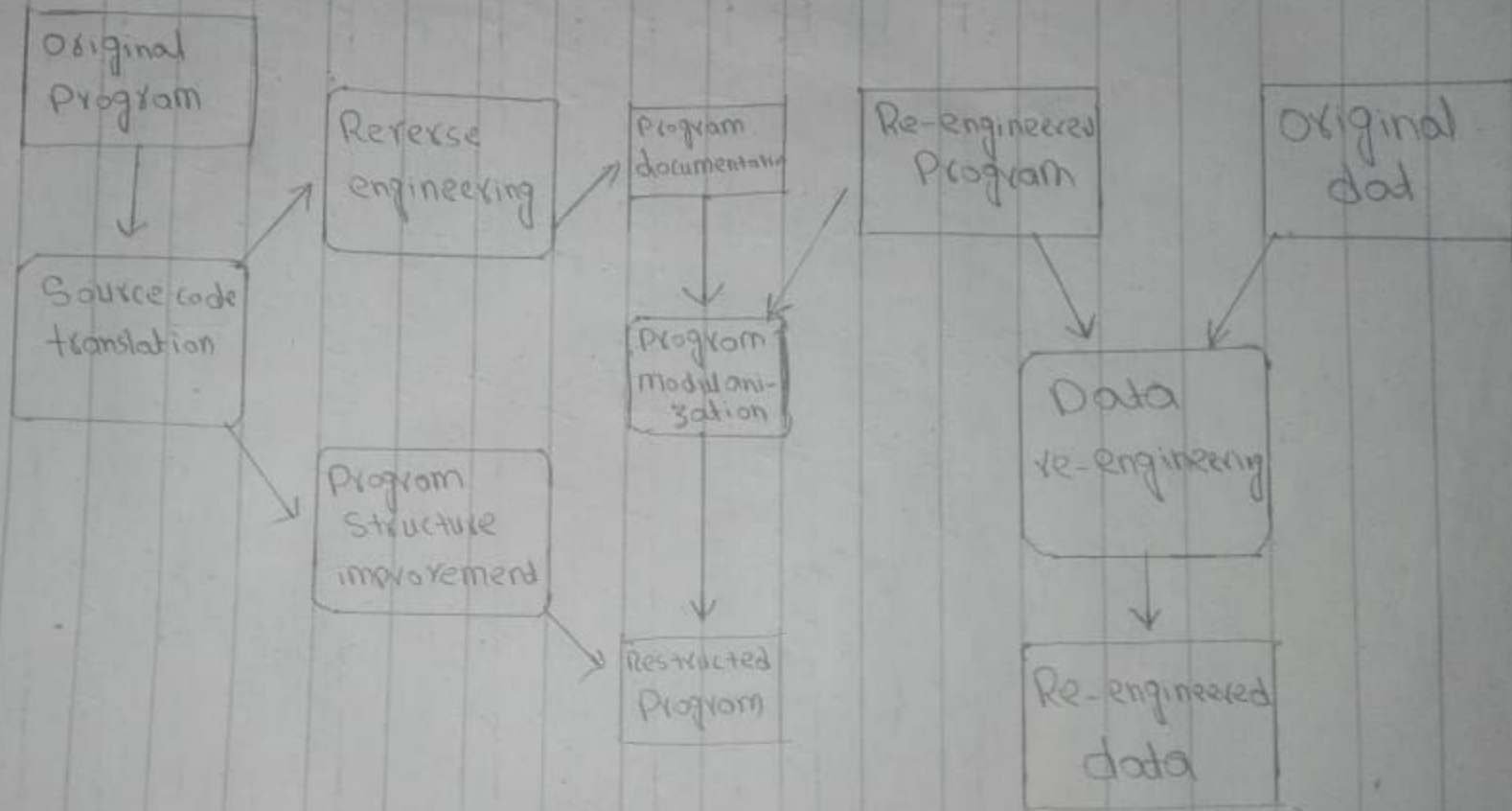
* The extent of the data conversion which is required.

* The availability of expert staff for reengineering.

- This can be the problem with old systems based on technology that is no longer widely used.

The Reengineering Process:

Diagram:-



Reengineering process activities:-

- * Source Code translation
Convert code to new language
- * Reverse Engineering:-
Analysis the program to understand it
- * Program Structure improvement:-
Restructure automatically for understandability.
- * Program modularisation.
Reorganise the programme structure.
- * Data reengineering:-
Clean up and restructure system data