

Name Asad Hussain.
BSLSE) section (B).

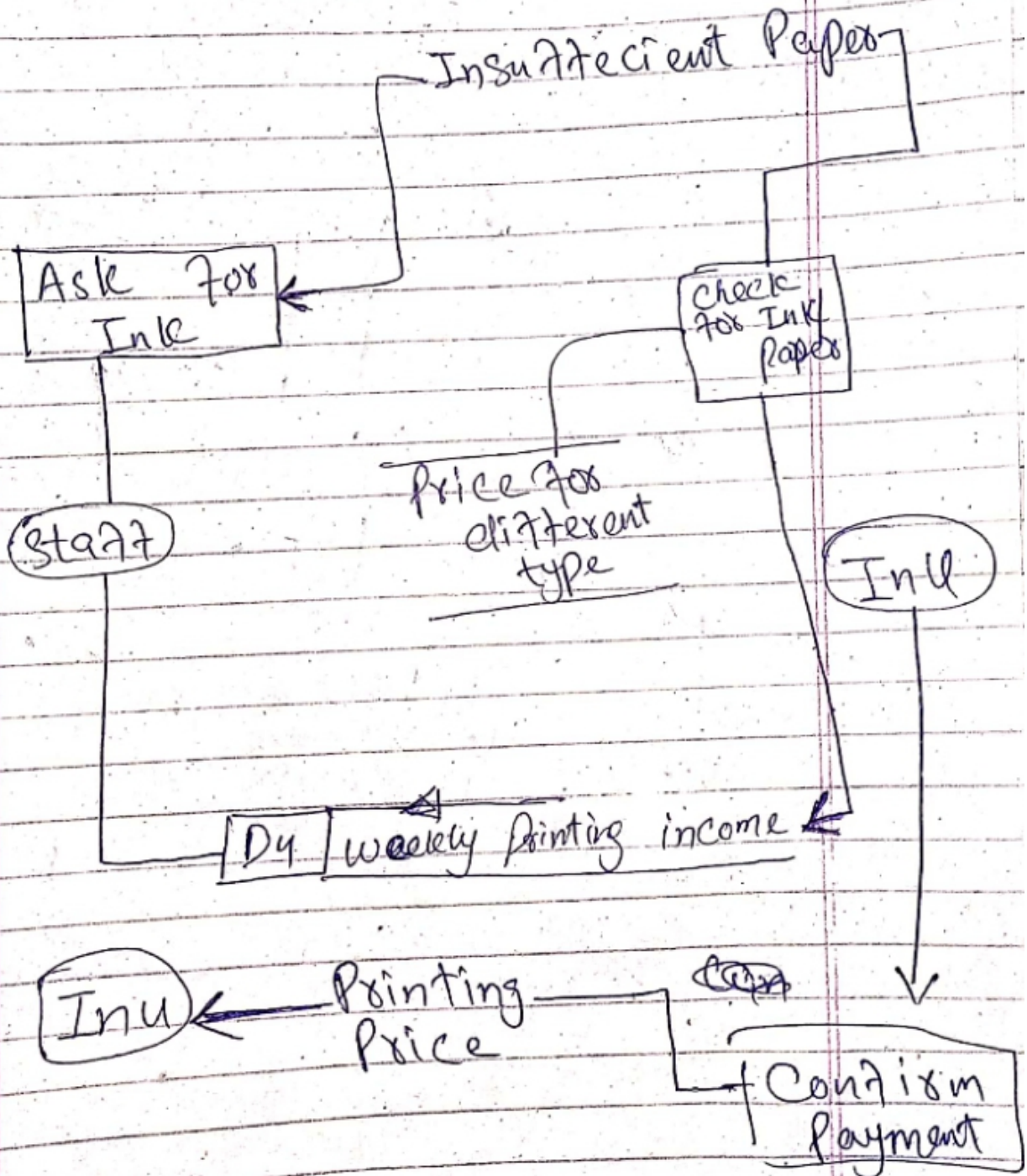
Software Engineering

ID : 14972.

1

:Q No 1:

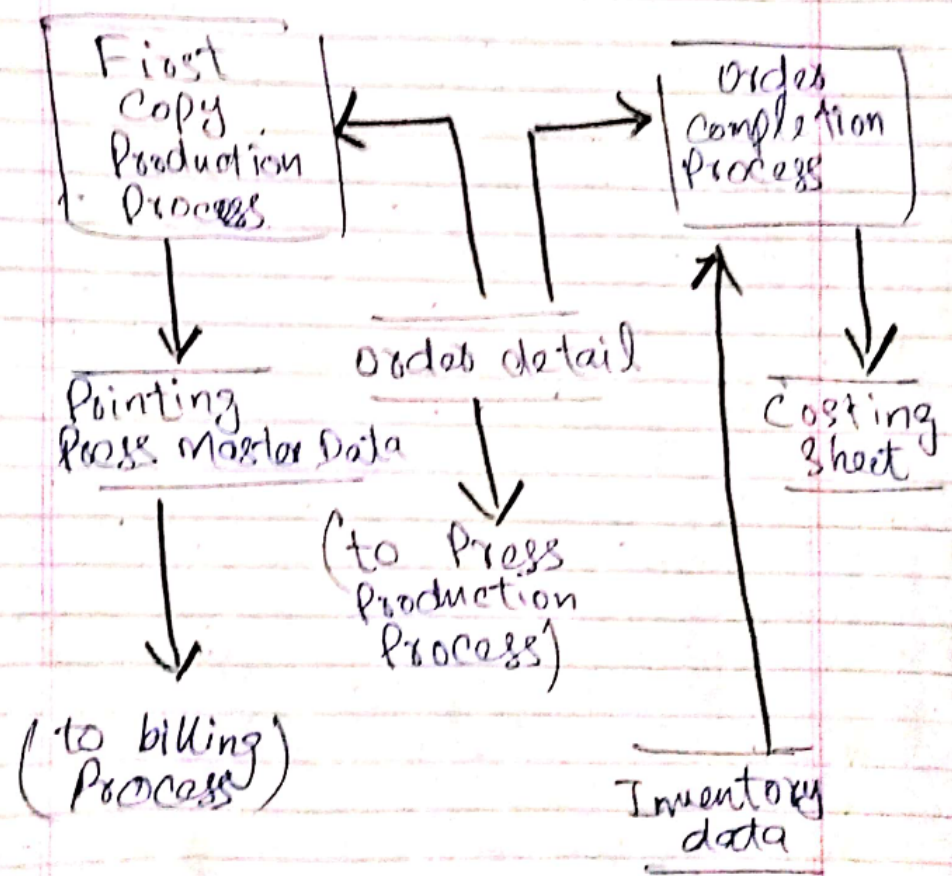
:Q.1.1:-



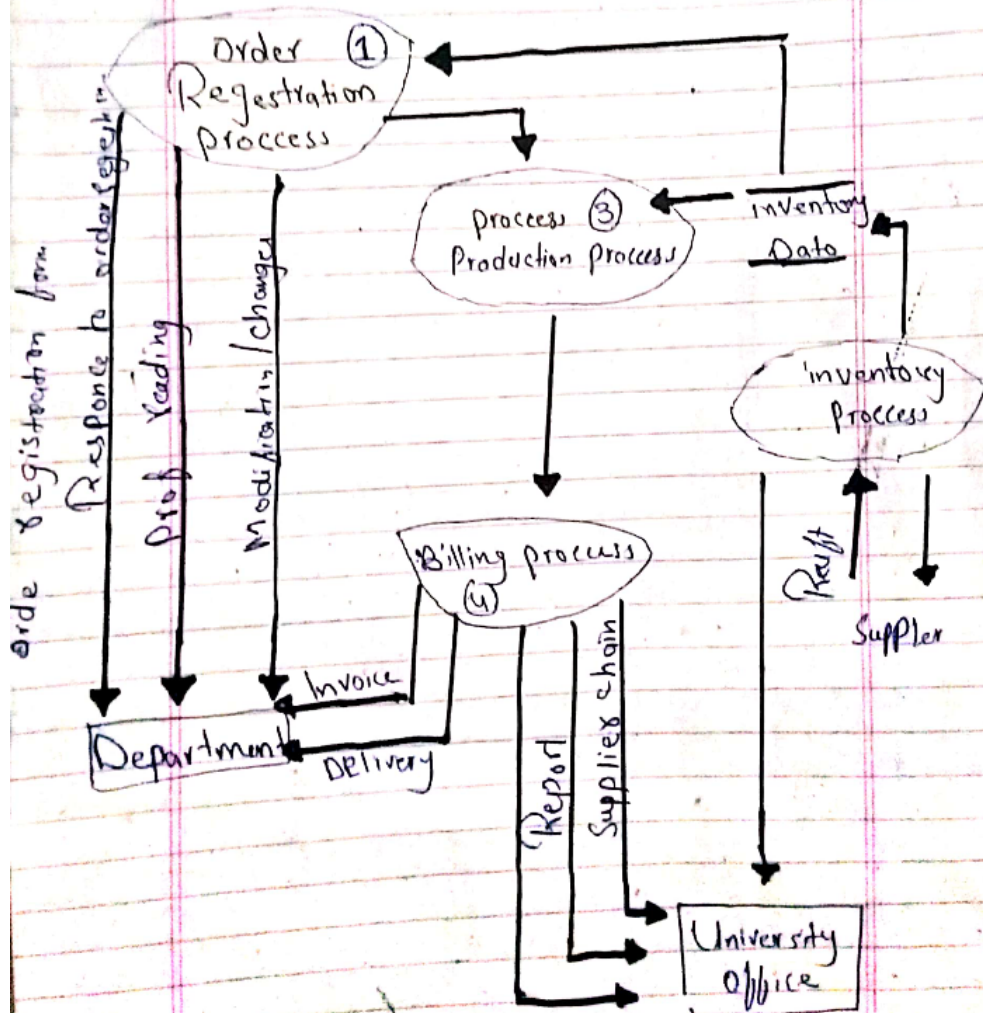
3

:Q No1:

: Q.1. ~~Q.2~~:



Q : 1: 2
: No 1 3:



4

: Question: 2:

: Q. 2.1:

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

: ANSWER:

Testing can detect only the presence of errors, not their absence because the main goal of the testing is.

To observe the behavior of the particular software and to check whether it meet 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

5

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.

:Q2:

:Q.2.2:

Define the following terms.

1. : UNIT Testing:-

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

6

2: SYSTEM TESTING:

* To verify that the system components perform control functions.

* To perform inter-system test.

* To demonstrate that the system performs both functionally and operationally as specified.

* To perform appropriate types of tests relating to Transaction Flow, Installation, Reliability, Regression etc.

3: 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.

7

* Covers all combined parts of a system.

* Tests 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.

8 4: White Box Testing / Structural Testing.

* Based on knowledge of internal logic of an application's code.

* Based on coverage of code statements, branches, paths, conditions.

* Tests are logic driven

* Its 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 operational bounds.

- Exercise internal data structures to ensure their validity.

9

: Q 3:

: Q. 3.1:

: Software Maintenance:-

* Modifying a program after it has been put into use.

* The term is mostly used for changing custom software. Generic software products are said to evolve to create new versions.

* Maintenance does not normally involve major changes to the system's architecture.

* Changes are implemented by modifying existing components and adding new components to the system, in unanticipated ways. These types of maintenance are

: Types of Maintenance:

(1) Maintenance to repair software faults.

* Changing a system to correct deficiencies in the way it meets its requirements.

(2) * Maintenance to adapt software to a different operating environment.

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

(3) Maintenance to add to or modify the system's functionality.

Modifying the system to satisfy new requirements.

Sometimes difficult to distinguish between them?

In practice, there is not a clear-cut distinction between these types of maintenance, when the system adapts to new environment, then add functionality to take advantage of a new environment 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 gives them different name.

"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.

12

: Q3:

: Q.3.2:

: System Re-engineering:-

* Re-engineering or re-writing part or all of a legacy system without changing its functionality.

* Application ble where some but not all sub-systems of a larger system require frequent maintenance.

* Re-engineering involves adding effort to make them easier to maintain. The system may be re-structured and re-documented.

: Re-engineering Costs Factors:

* The quality of the software to be re-engineered.

* The tool support available for

* The extent of the data conversion which is required.

* The availability of expert

Start for re-engineering.

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

