

Name

Safiullah

ID

14677

Program

BS SE

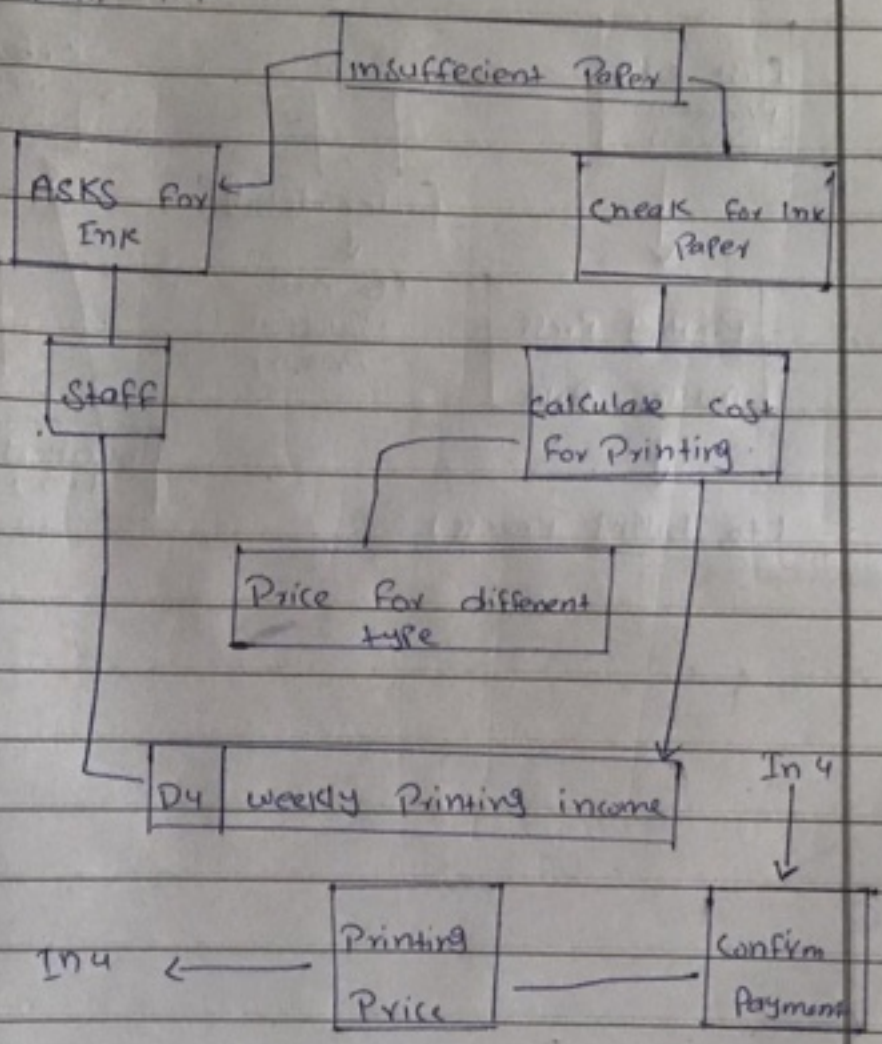
Section

B

Subject

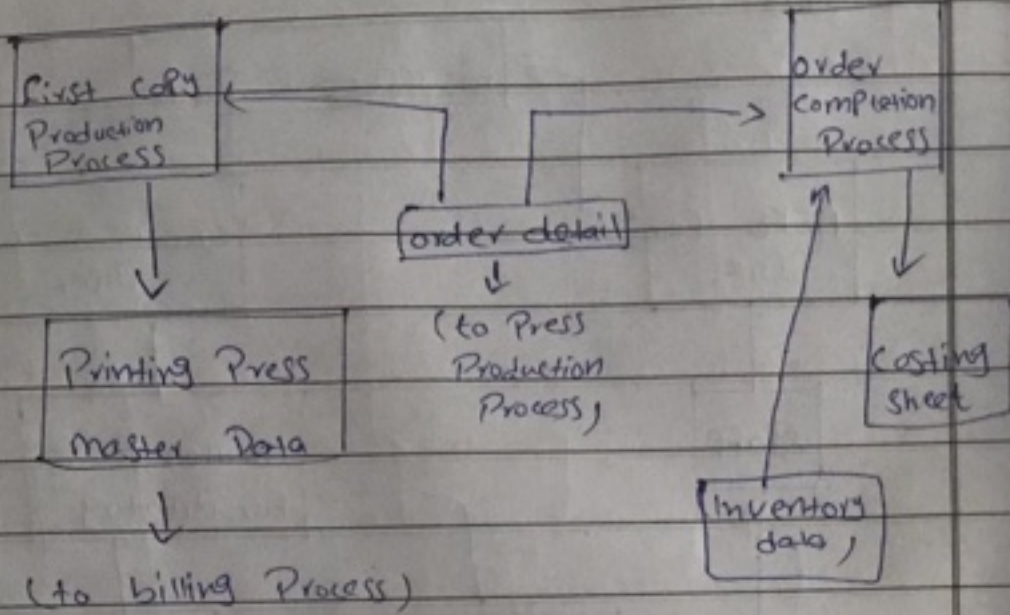
Software Engee

Q: 11  
Part 1.1



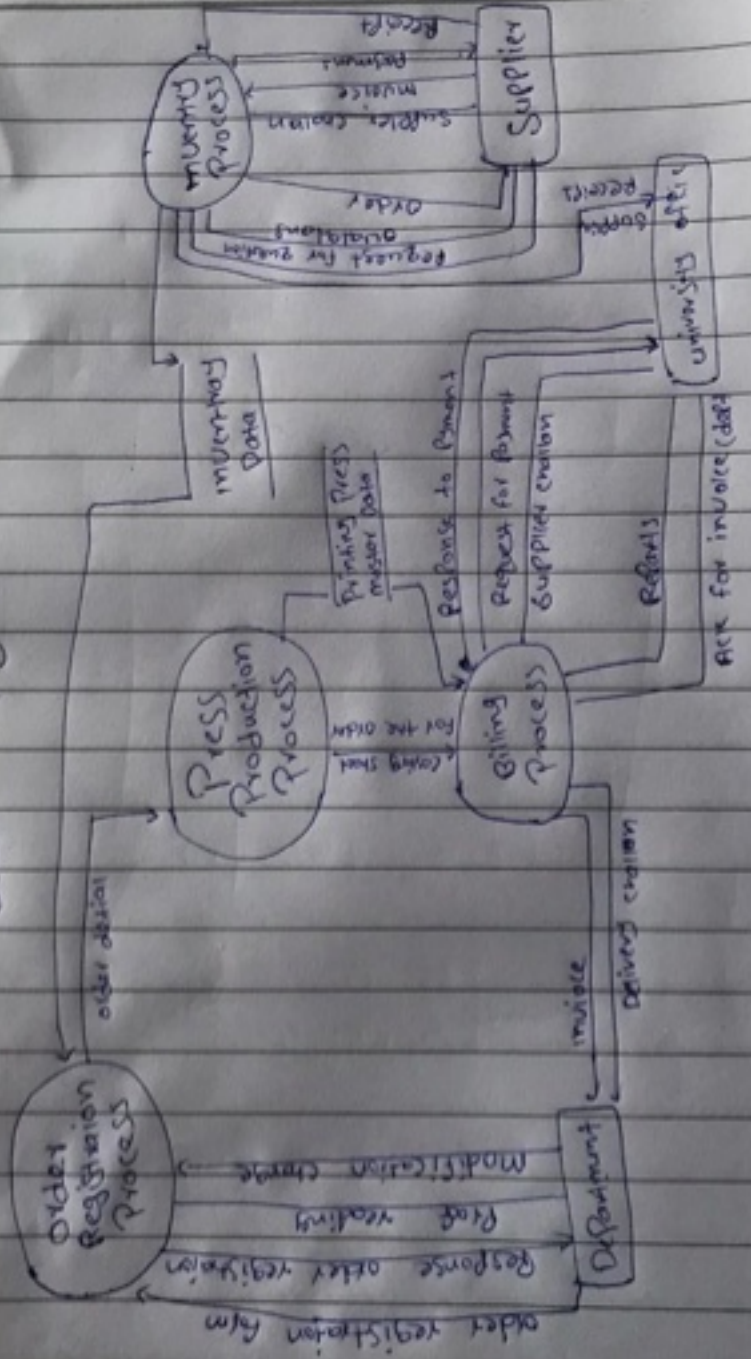
Q. 1

Part 1.2





Question 1.3



Question 2:

Part 2.1

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 meets its requirement expectation or not."

Testing is a part of broader process of software verification and validation. It tester 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 discover further problem with the system.

Q. 2.7

### Unit Testing:

- The most 'micro' scale of testing
- Tests done on Particular functions or code modules.
- Requirement knowledge of the internal Program design and code
- Done by Programmers (not by testers)

Objectives: To test the function of a Program or unit of code such as a Program or module

- To test internal logic
- To verify internal design
- To test Path & conditions coverage
- To test exception conditions and error handling.

When: After modules are coded.

Input: Internal Application Design

- Master Test Plan
- Unit Test Plan

Output: Unit Test Report

Who: Developer.



Methods: white box testing techniques

Tools: Debug

Re-Structure

Code Analyzers

Path / Statement coverage tools

Education: Testing methodology

Effective use of tools

## (2) SYSTEM TESTING:

Objectives: To verify that the system components perform control functions

- To Perform inter-system test
- To demonstrate the system performs both functionally and operationally as specified
- To Perform appropriate types of tests relating to transaction flow, installation, Reliability, Regression etc.

When: After integration testing

Input: Detailed Requirement and External Application Design

- Master test Plan
- System test Plan

Output: System Test Report

Who: Detailed Development team and user

Methods: Problem/ Configuration management

Tools: Depends

Education: Testing methodology.

### (3) Black Box Testing.

- No knowledge of internal design or code required
- Tests are based on requirement and functional
- Not based on any knowledge of internal design or code
- covers all combined parts of a system
- Test are data driven (Tests are based on putting some data to check the system)
- it uncovers
  - incorrect or missing functions
  - interface errors
  - Performance errors

#### Types of black Box testing.

Functional testing, System testing

Regression testing, Load testing

Recovery testing, etc.



#### (4) White Box Testing:

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

Q. (3)

Part B.1

Answer:

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 adapts to new environment, then add functionality to take advantage of the 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 sometime gives them different names.

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

"Adaptive maintenance" sometimes means adopting to new requirement

Perfective maintenance sometime 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.

Q. 3

Part 3.2

Answer:

System re-engineering

- Re-structure or re-writing part or all of a legacy system without changing its functionality.
- Applicable where some but not all sub-system of a larger system require frequent maintenance.
- Re-engineering involves adding effort to make them to maintain. The system may be re-structured and re-documented.



## Advantages of reengineering

### • Reduced risk

There is a high risk in new software development. There may be development Problem, Staffing Problems and Specification Problems.

### Reduced Cost.

The cost of re-engineering is often significantly less than the costs of developing new software.

## The reengineering Process

