

NAME = ABDUL

MATEEN

KHAN

ID = 14452

BS (SE-4) Section A.

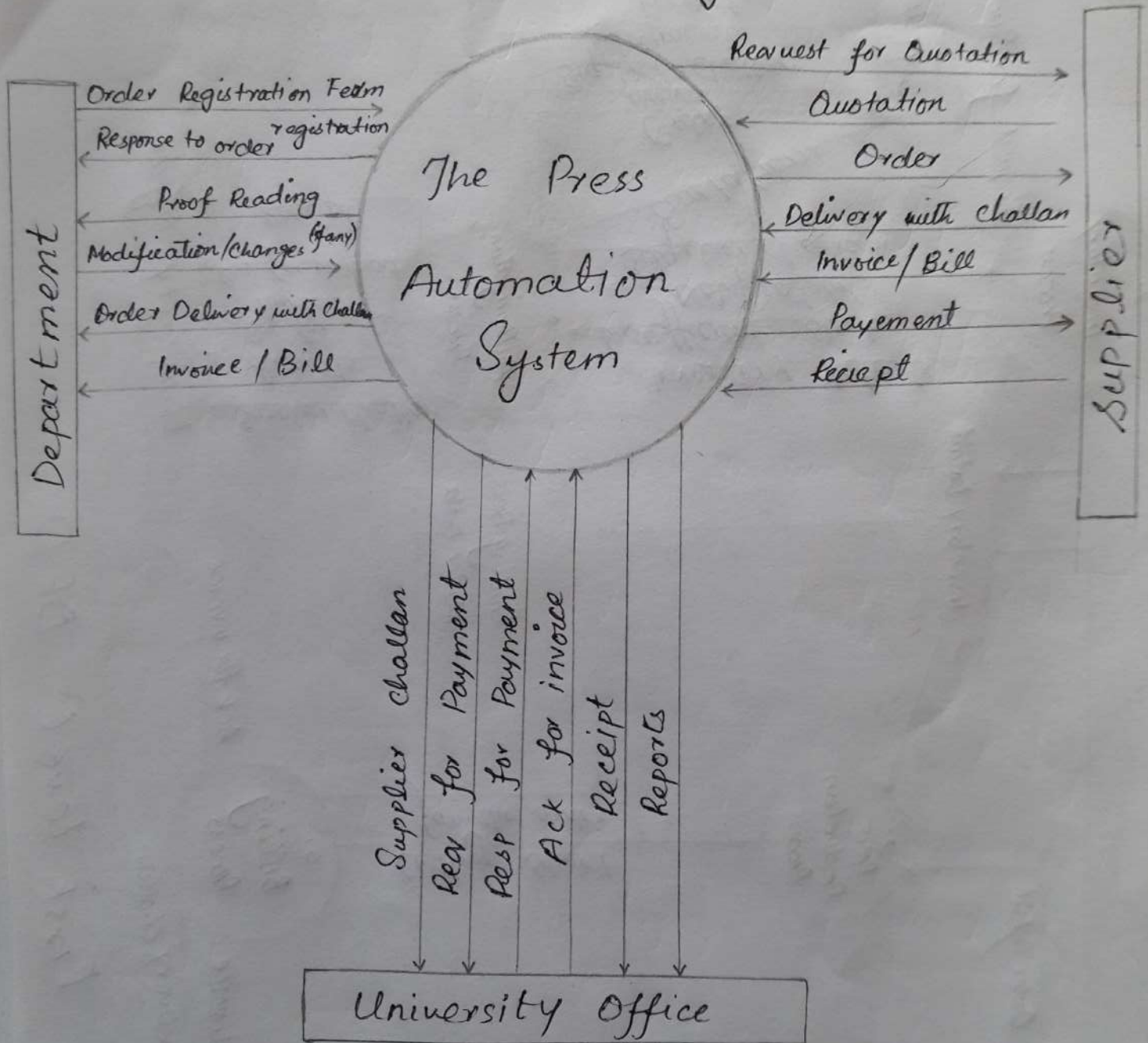
Software Eng.

Name = Abdul Mateen Khan

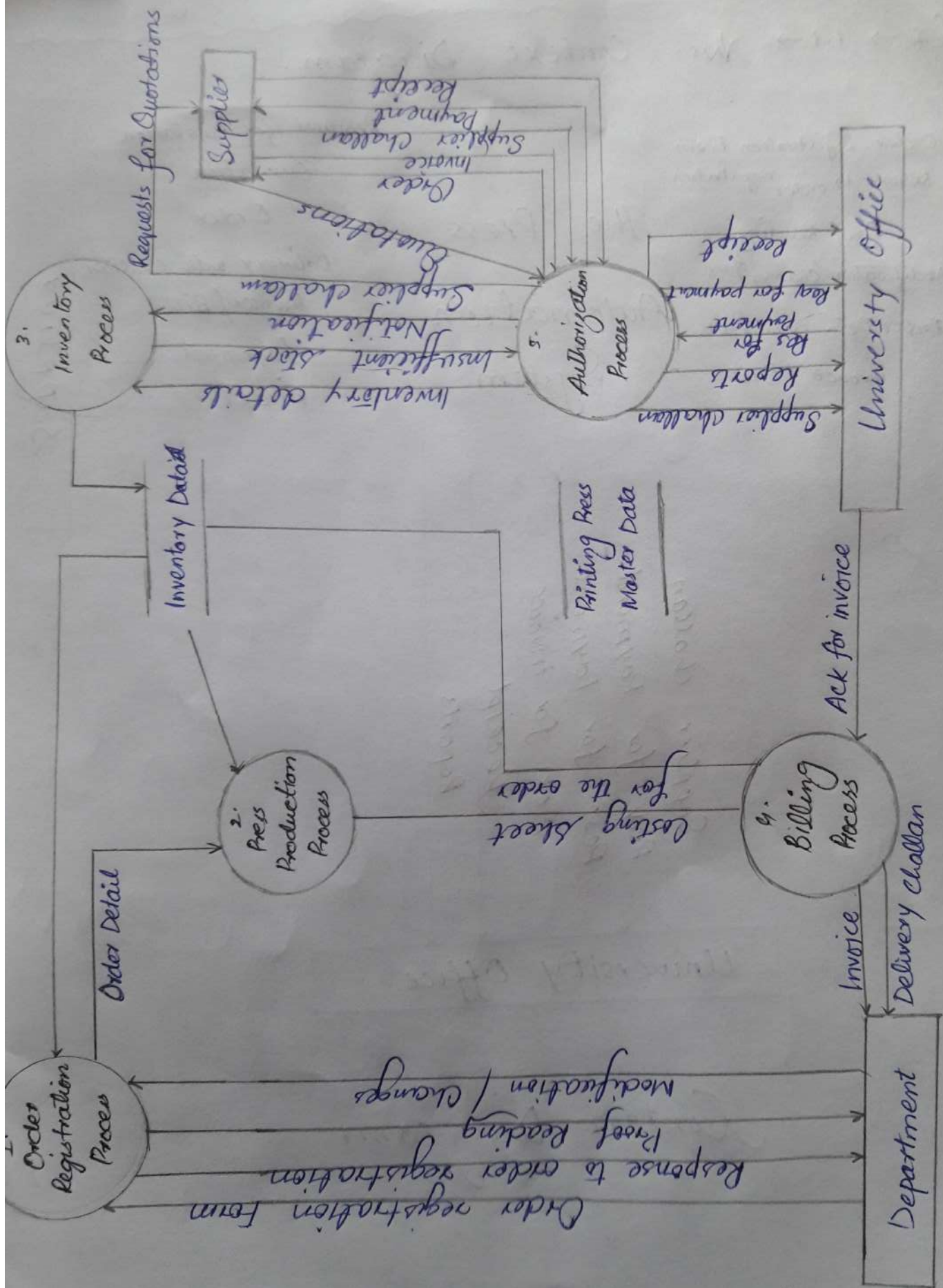
ID = 14452 (BSSE-4) Section A.

Page = ①.

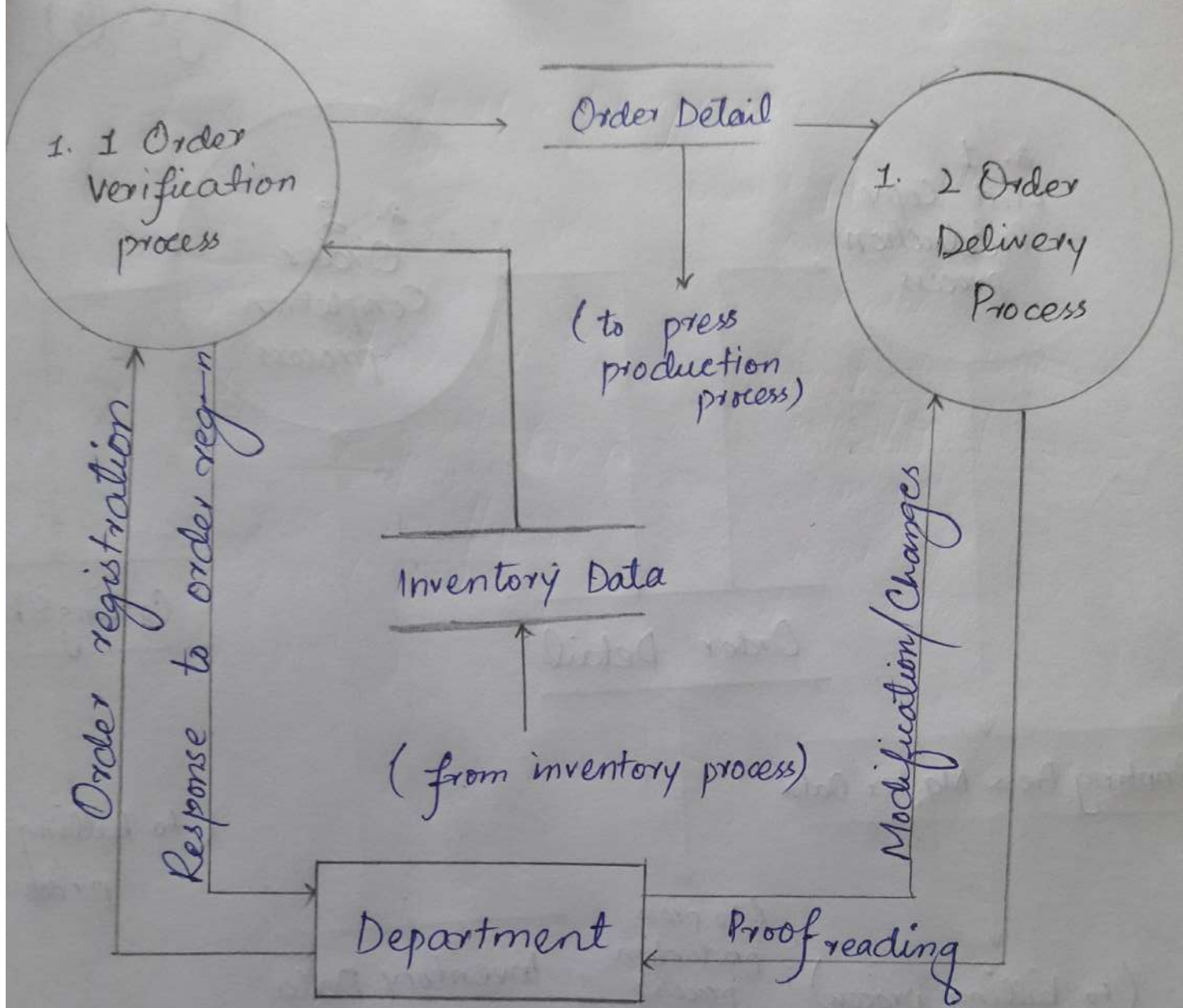
Q1 → 1.1 → Ans. Context Diagram.



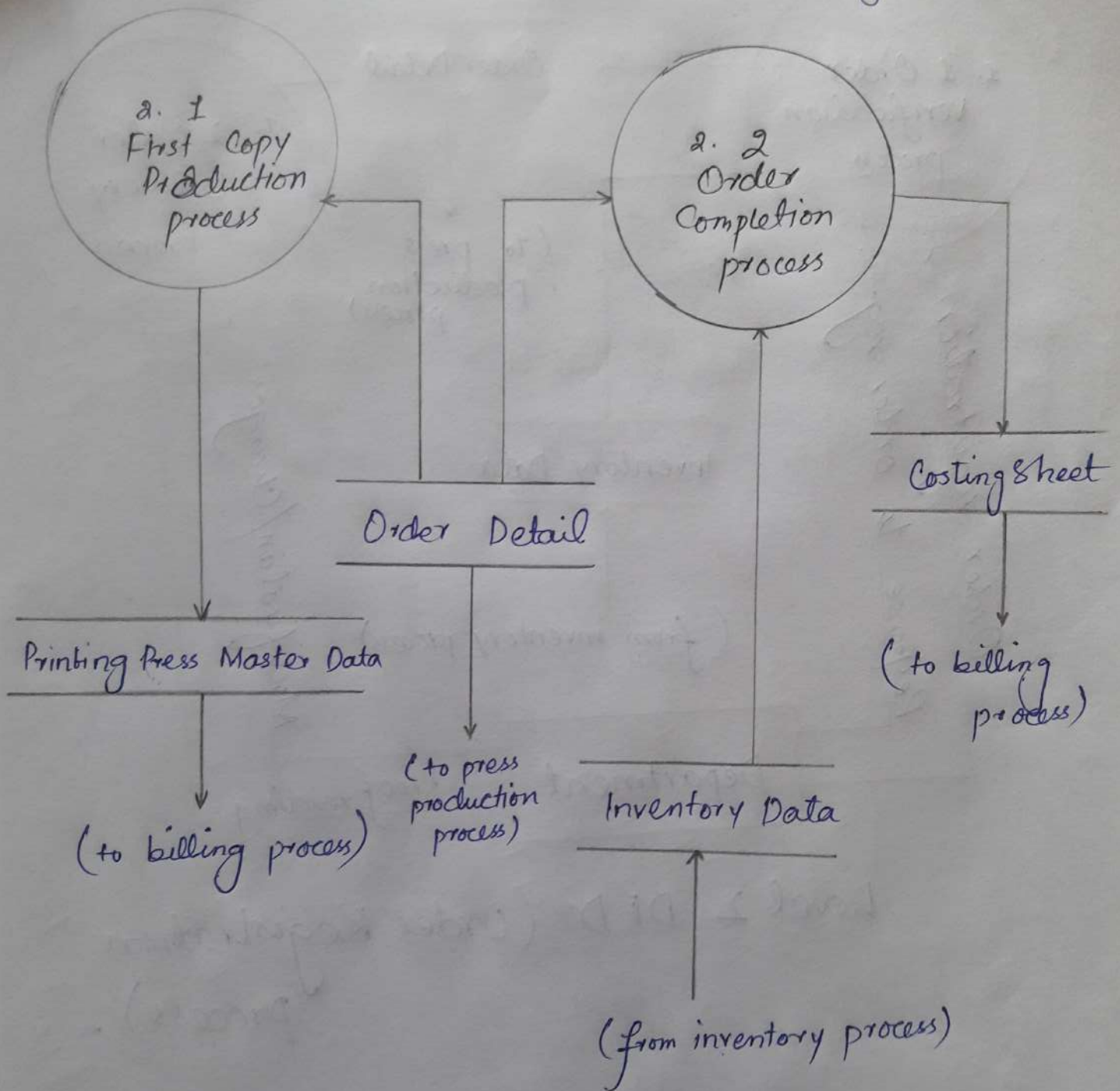
Context Diagram



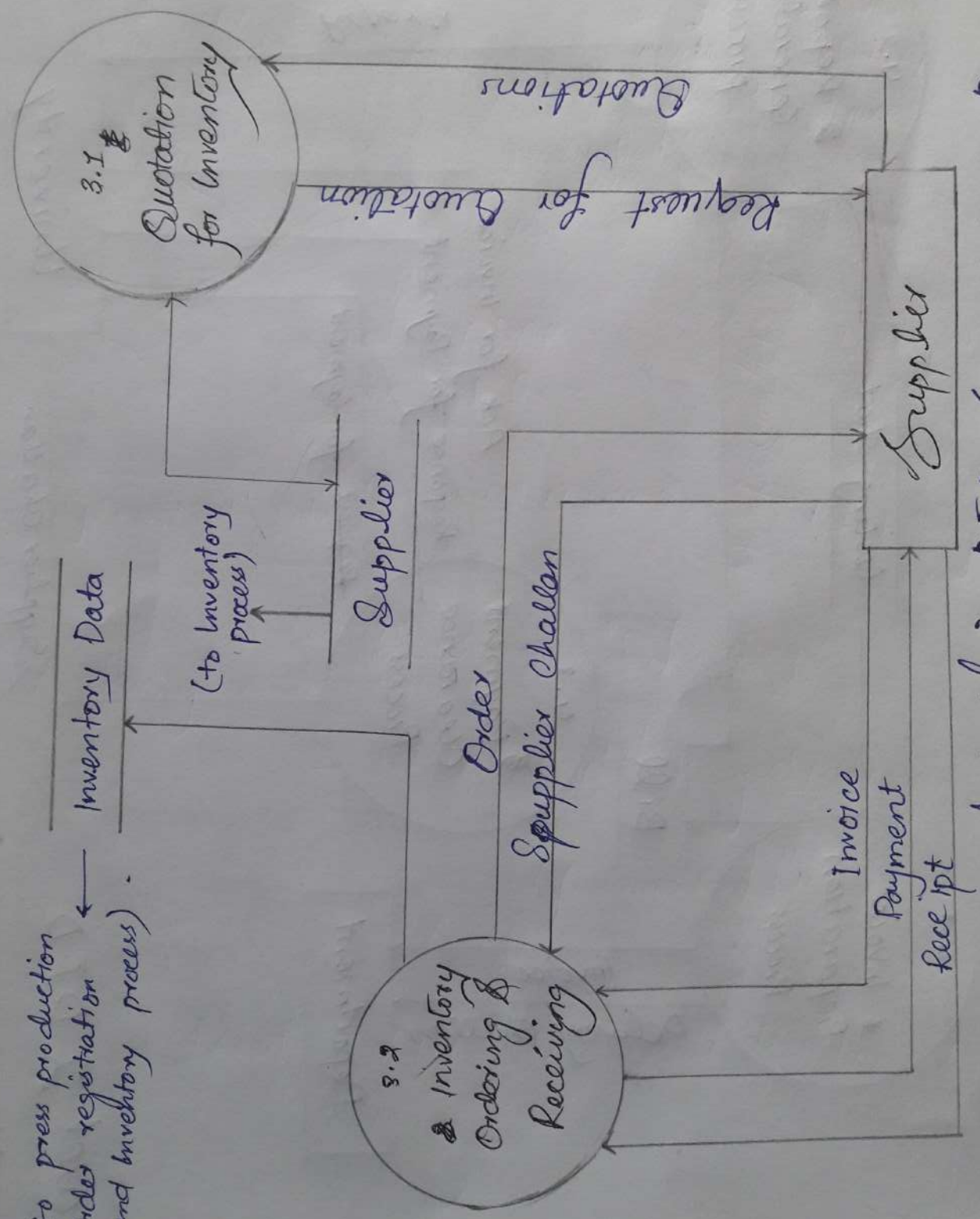
First level DFD



Level 2 DFD (Order Registration process).

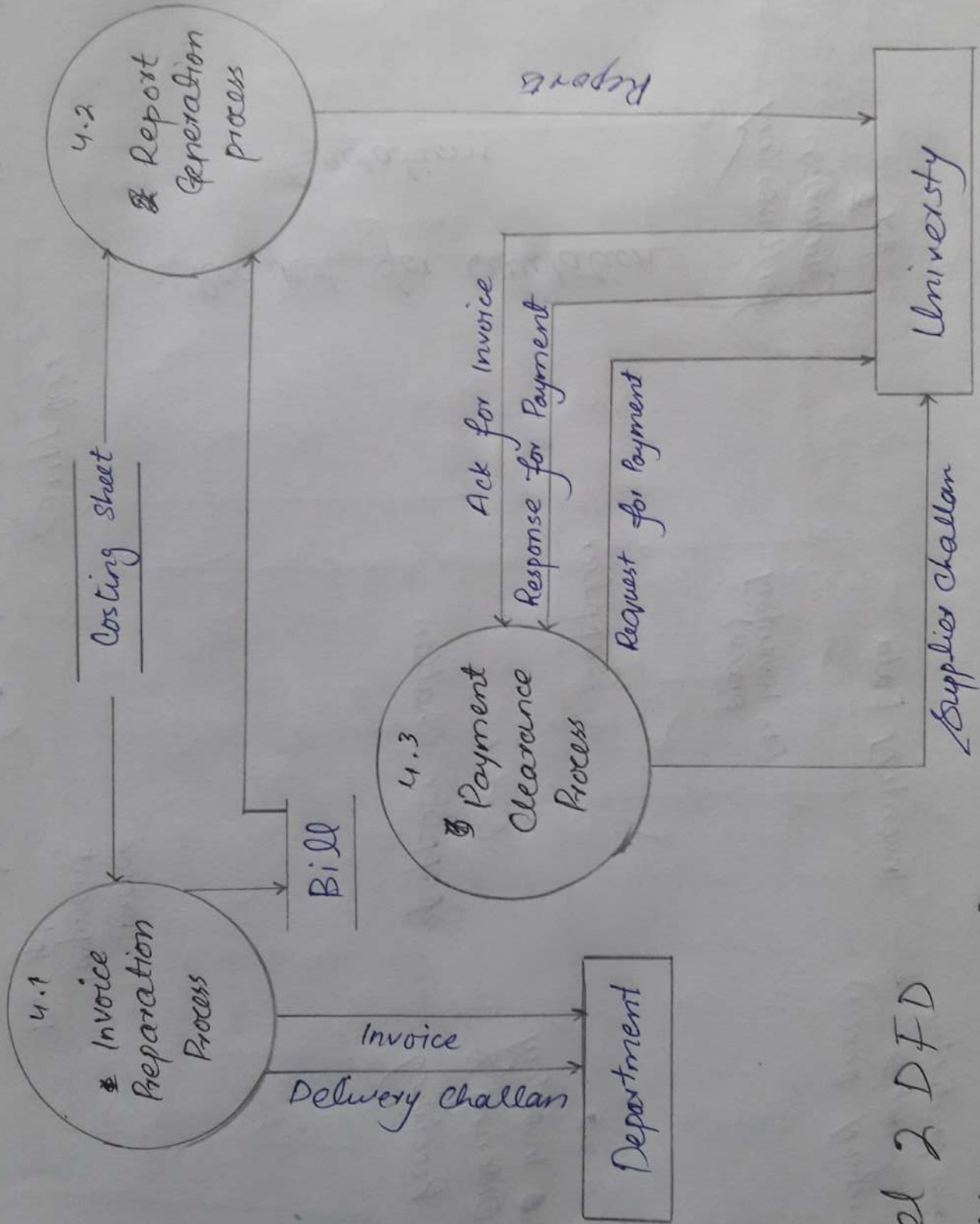


Level 2 DFD (Press Production Process).



(to press production Order registration and inventory process).

Level 2 DFD (Inventory Process).



Level 2 DFD
(Billing process)

Q2 → 2.1 Explain why testing can only detect the presence of errors, not the absence?

ANS:- Testing can only detect the presence of errors, not the absence because the main goal / target of the testing of software is to observe the software behaviour to meet its requirement expectation or not.

Testing is a part of broader process of software verification and validation. Testing cannot demonstrate the faults other than specified in every circumstance. Test case 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 that inputs chosen is not faulty.

Q2 → 2.2 → Define the following terms?

Ans:- ① Unit testing:-

The most 'micro' scales of testing and tests are done on particular function or code modules, it requires a knowledge of the internal program design and code and done by programmers.

② System testing:-

It is a level of software testing in which the purpose/objective of the testing is to verify that the system components perform control functions and is based on overall requirements specification.

③ 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.

④ White-box testing:-

White box testing is a method of software testing that tests internal structure of workings of an

application, as opposed to its functionality.

Q3 → 3.1 → Briefly describe the three main types of software maintainance, why is it sometime difficult to distinguish between them?

Ans: - There are four types of software main-
tainance but we will discuss three.

→ Corrective software maintainance:-

Corrective
Software maintainance is what one would typically associate with the maintenance of any kind. Correct software maintainance address the errors and fault within software application that could impact various parts of your software, including the design, logic, and code.

→ Adaptive software maintainance:-

Adaptive soft-
ware maintainance becomes important when the environment of your software changes. This can be brought on by changes to the operating system, hardware, soft-
ware dependencies, Cloud storage, or even

changes within the operating system. sometime, Adaptive software maintenance reflects organization policies or rules as well.

→ Perfective software maintenance:-

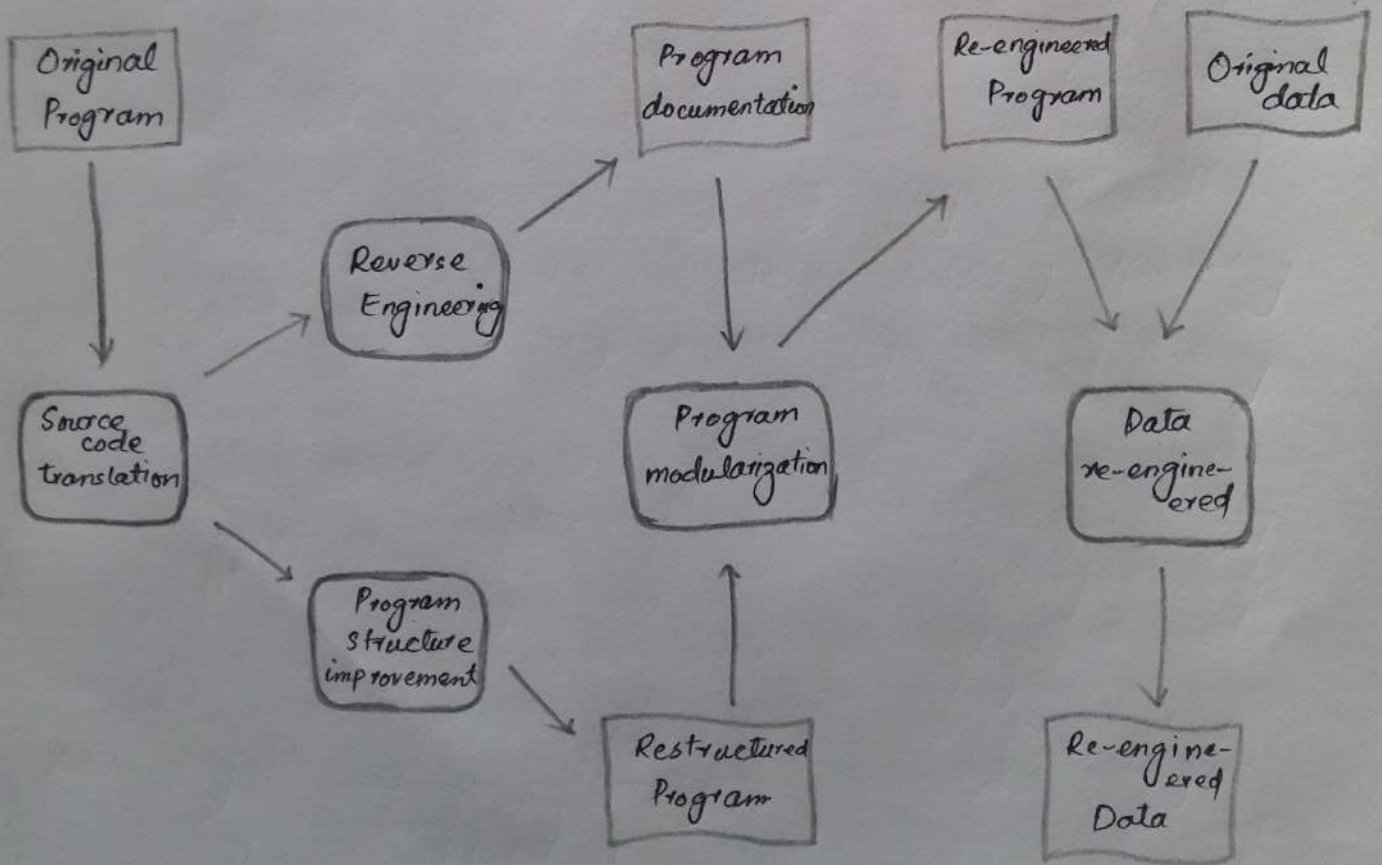
Perfective software maintenance focuses on the evolution of requirements and features that existing in your system.

⇒ They are sometime difficult to distinguish because the same set of changes may cover all three types of maintenance. For example, a reported fault in the system may be repaired by upgrading some other software and then adapting the system using to use new version (corrective + adaptive).

Q3 → 3.2 → What are the principal factors that affect the costs of system reengineering? Also briefly explain the reengineering process with the help of diagram?

Ans:- Factors that affect the costs of system reengineering :-

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



- Source code translation.
 - convert code to a new language.
- Reverse Engineering.
 - Analyse the program to understand it.
- Program structure improvement.
 - Restructure automatically for Understandability.
- Program modularisation.
 - Reorganize the program structure.
- Data Re-engineering.
 - Clean-up and restructure system data.