NAME: ABDUL SALAM

REG ID 14480

DEPARTMENT OF SOFTWARE ENGINREEING

SEMESTER 4TH

SECTION A

Name: Abdul Salam

Reg: ID 14480

Semester ~~8~~ 4th

Subject Software Engineering.

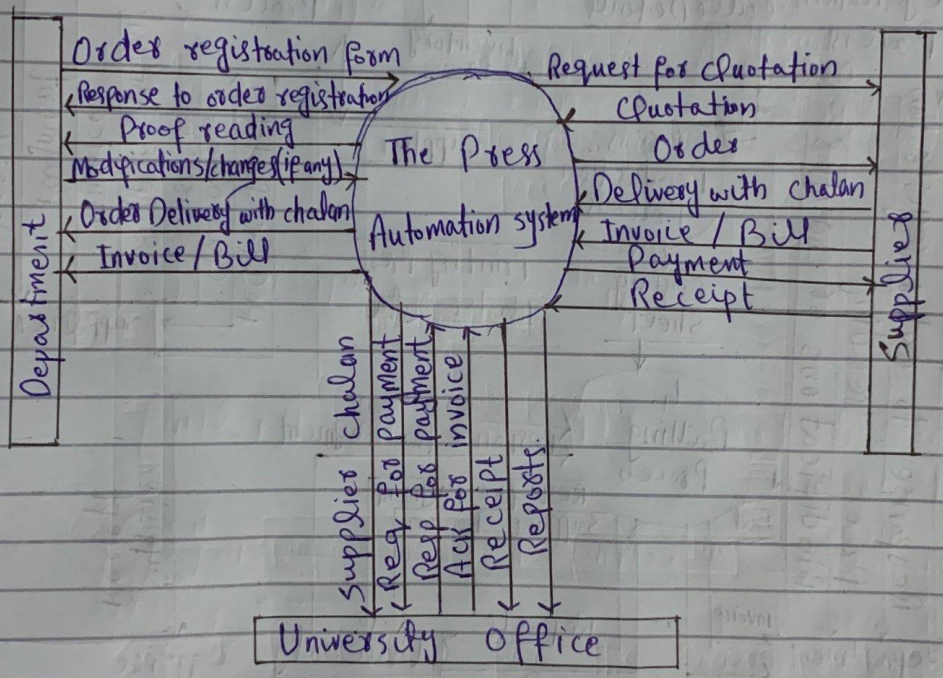
Submitted to Sir Ghassan Husnain.

Name: Abdul Salam ID: 14480 Software Engineering

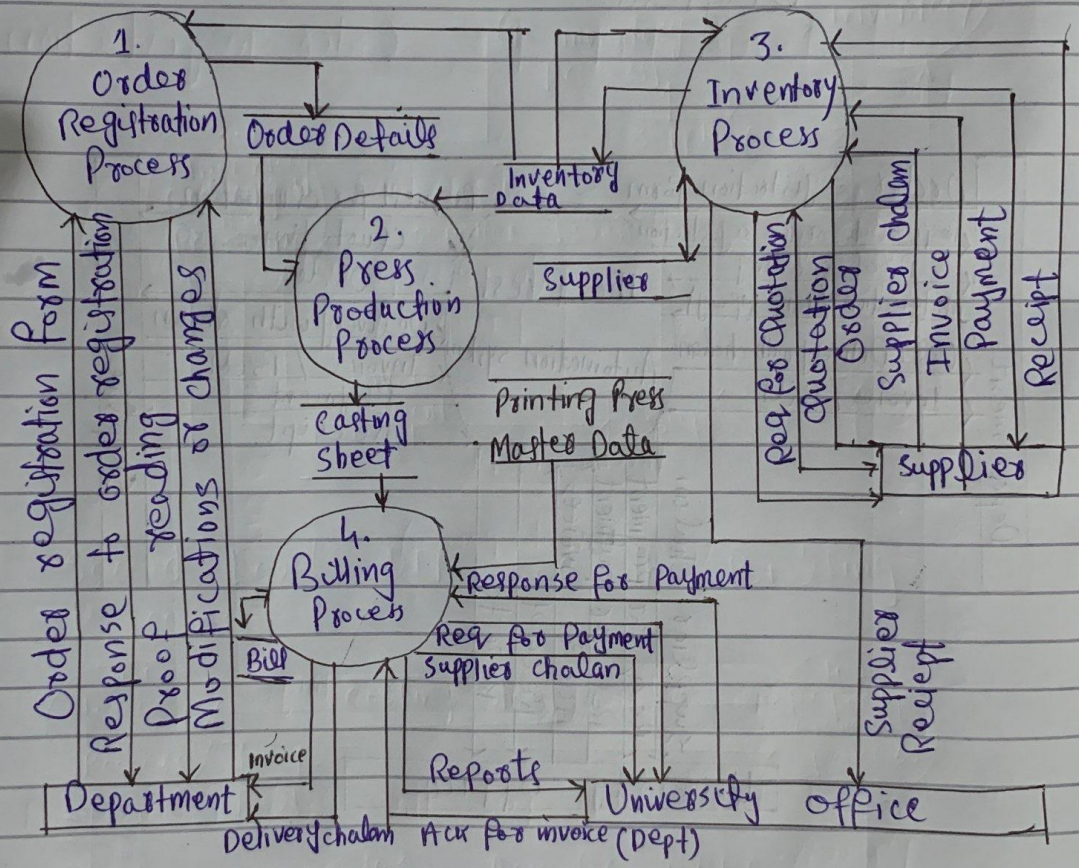
Question 1.1: Case Study.

1. Draw a Context diagram for INU Printing Press ?

Context Diagram:

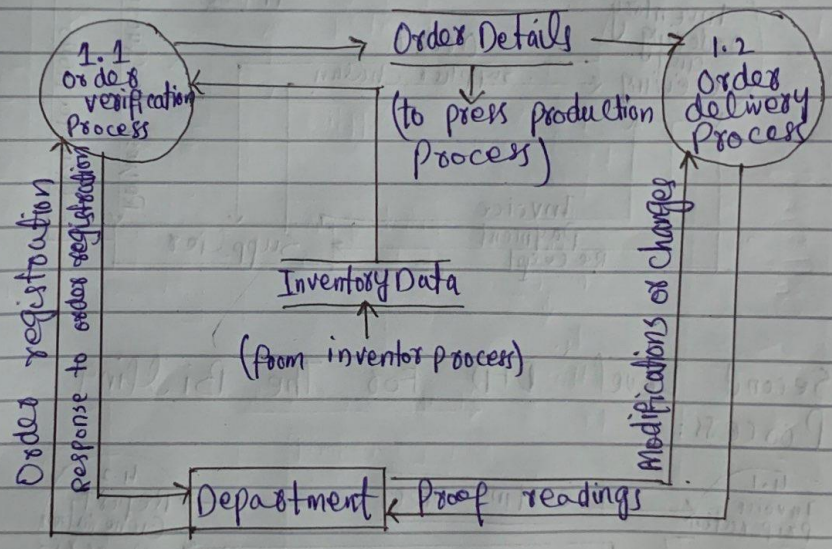


Q12: Draw a Level 1 Data Flow Diagram (DFD) for the above case study?

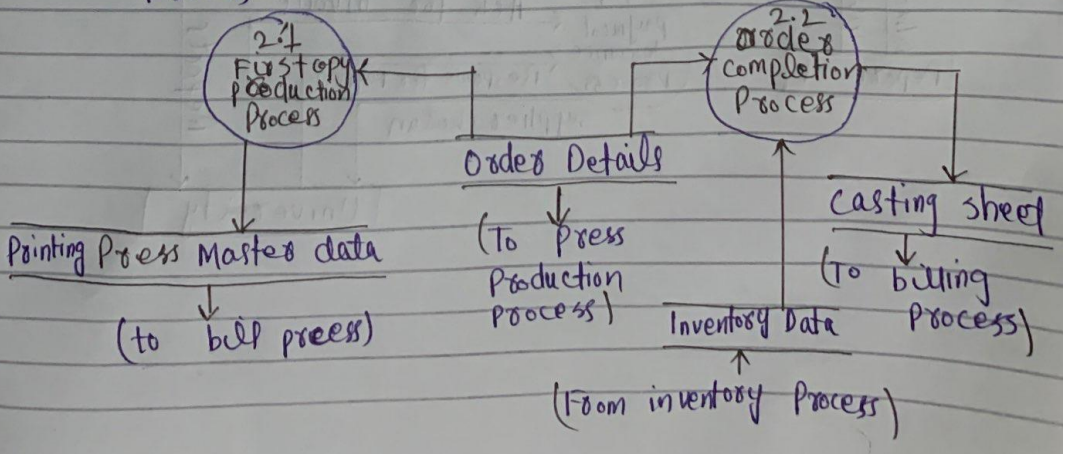


Question 3: Draw a Level 2 DFD for the Order Registration Process, Press production Process, Inventory process, and Billing Process.

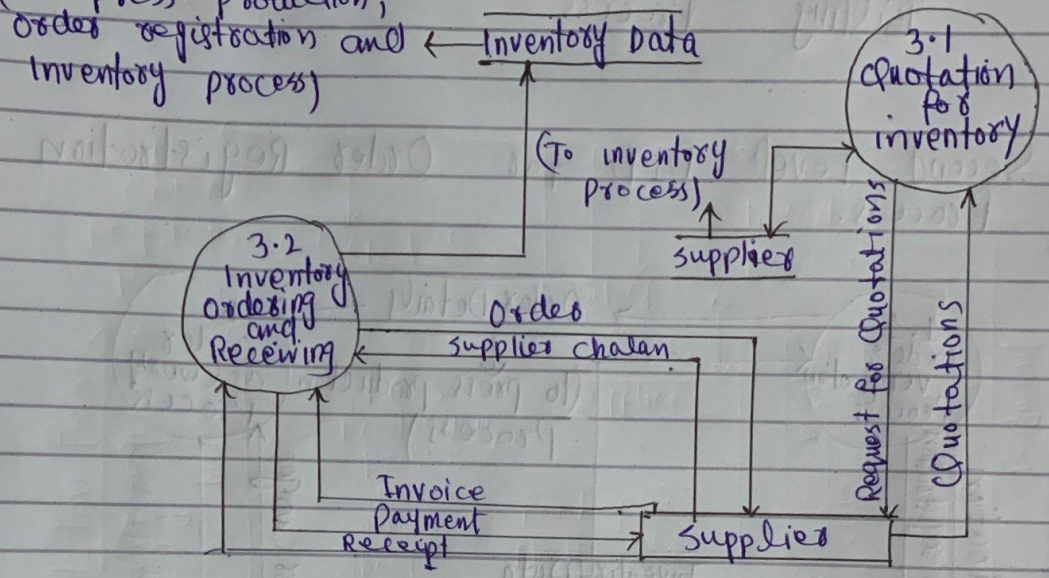
Second Level DFD for Order Registration Process.



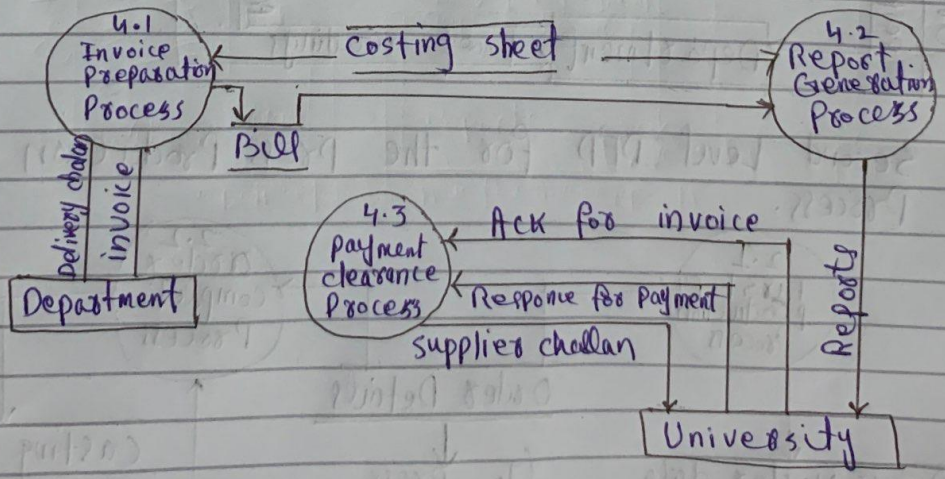
Second Level DFD for the press Production Process.



Second Level DFD for the Inventory Process.



Second Level DFD For The Billing Process:



Question 2:

Q. 2.1: Explain why testing can only detect the presence of errors, not their absence?

RESULT:

Softwar testing is Observing softwar behavior to meet its need.

Testing is a set of activities that the tester tries to demonstrate to the software such behavior to detect an error or disorder, which can then be fixed.

Q. 2.2: Define the following terms.

1. Unite testing.

In computer programming, Unit testing is a software testing method that tests one or more sets of computer program modules with source code, associated control data, usage process, and operation procedures. Are they suitable for use.

2. System Testing:

System testing is the level of testing that verifies a complete and fully integrated software product. The purpose of system testing is to evaluate end-to-end system specification. In general, software is only one aspect of a larger computer-based system. Finally, the software interferes with other softwares/hardware systems. System testing is actually a series of different tests, whose sole purpose is to use a complete computer-based system.

Two category of Software testing

- Black box testing.
- White box testing

Blaek box testing:

Black box is defineed as a techniqne in which the functionality of an application under test (AUT) is tested without knowledge of internal code structure, implementation details, and software's internal lines. This type of testing is entirely based on software requirements and specifications. Blackbox testing focuses on the input and

output of the software system without worrying about the software's internal knowledge.

White box testing:

White box is testing the architecture design and coding of a software solution. In this type of test, the code appears to the tester. It mainly focuses on validating the flow of input and output through the application, improving design and usage, and strengthening security. White box testing is also known as clear box Testing, Open box Testing, Structural Testing, Transparent Box Testing, Code Based Testing and Glass testing. This is usually done by developers.

Q.3.1: Briefly Describe the three main types of software maintenance

Ans: (A) Correctional maintenance or fault repair. Changes to the system are intended to correct occurs that are reported to be program bugs or specification errors or errors.

(B) Adaptive management or environmental adaption. Modifying software to adapt to your environment, e.g changes to other software systems.

(C) Complete maintenance or operational adelition. This includes adeling new functionality or features to the system.

It is sometimes difficult to detect them, because the same changes involve three types of maintenance. For example: A bug that was reported in the system was repaired by upgrading some other software, and then adapting the system to use this new version (Correction + compatible). New software may adal additional functionality and add new features to take advantage of it, as part of custom management.

Q.#3.2 What are the prineipal factors that affect the cost of system re-engineering.

Answer: Cost of Re-engineering depend on the the extent of the work that is carried out. Cost increases from left to right so that source code translation is the cheapest option and re-engineering as part of the architectural migration is the most expensive.

- Apart from the extent of the re-engineering, the principal factors that affect re-engineering cost are:

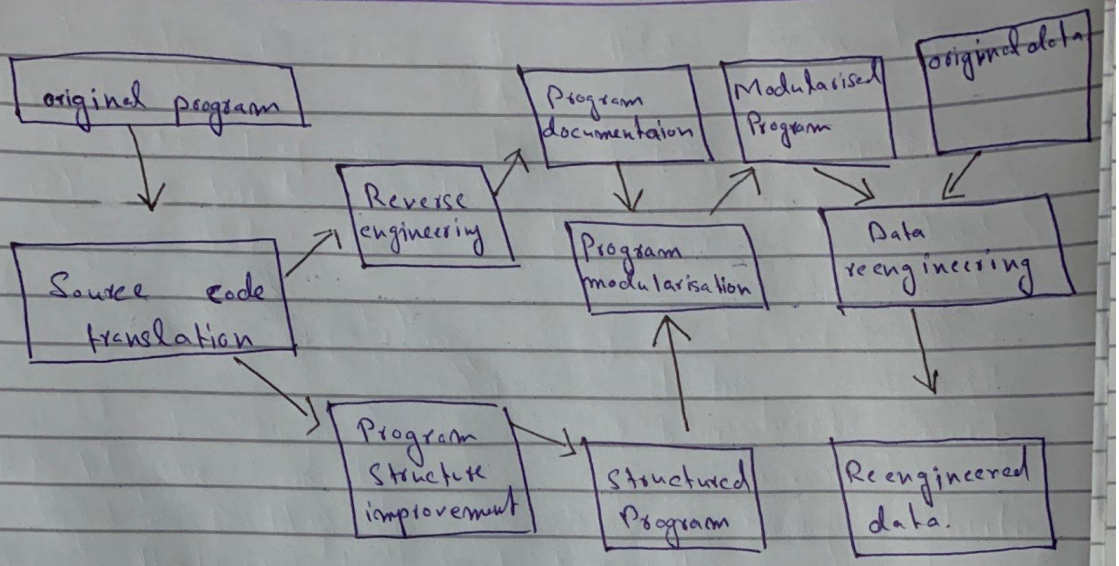
1- The quality of the software to re-engineered: The lower of the quality of the software and its associated documentation (if any), the higer re-engineering cost.

2. The tool support available for re-engineering: The use of CASE tools to automate most of the program changes is normally cost effective to re-engineer a software.

3. The extent of data conversion required: If engineering requires large volumes of data to be converted, this significantly increases the process cost.

4. The availibility of expert staff: If the staff responsible for maintaining the system can't be involved in

re-engineering process, this will increase the costs, System re-engineers will have to spend a great deal of time understanding the system.



Re-engineering Process:

The input to the process is a legacy program and the output is the modularized version of the same program. As the same time as program re-engineering, the data for the system may also be re-engineered. The activities in this re-engineering process are. Source Code Translation, Reverse Engineering, Program Structure Improvement, Program Modularization, Data Re-engineering