Name	Hilal Ahmad
Id	14728
Class	Bs (SE) 4th semester
Section	B
Paper	software Engineering
Instructor	Guisson Husain
Date	6/22/2020

Question 1: **Case Description**

INU Printing Press, a part of a University undertakes printing jobs from different departments of the University. The press can take printing related orders from other organizations or individuals also, but this is a rare practice. Orders once placed cannot be canceled. Any financial transactions of the various departments with the printing press are done through the University Office only. The University Office maintains the accounts of the press and the various departments of the University. The press acquires its inventory from outside suppliers. This is done by accepting the quotations from various suppliers and placing orders respectively. They maintain information of vendors / suppliers for the raw materials. A supplier can supply more than one item and an item can be supplied by more than one supplier. The supplier may supply the item in parts. Stock for a particular item can be acquired more than once in a year. The suppliers send two copies of invoice to the printing press, of which one is forwarded to the University Office. The press makes payment to the suppliers after obtaining approval from the University Office. The department places printing related order with the press and depending on the inventory, the press is entitled to give a response specifying whether the order will be fulfilled immediately or in the near future. Each department is assigned a budget at the beginning of the year, which becomes an important factor while getting any concession or discount from the press. Once the order is accepted, it has to go through various printing processes for its completion. However, these processes or the sequencing are not mandatory as a job or task may need different processing or handling. The various processes are:

- Composing
- Pasting
- Plate making
- Printing
- Binding

At the completion of the first copy, it is sent to the department for approval, which when received initiates the completion of the order. For the completion of the order, the press maintains a document known as costing sheet which specifies what all processes were applied and the inventory used for the completion of a particular order. This document is important in generating the bill.

3 delivery chalans are sent along with the items; one being kept in the press, second being given to the department and the third being sent to the University Office by the press. The invoice is not sent with the items, but it is sent at a later date due to complexity involved in preparing it. Two invoices are sent to the department; of which one is forwarded to the University Office by the department.

Summary of transactions as well as reports related to the printing press, are sent to the University Office on a regular basis. These reports or summary is useful to the management for analyzing their current budget and planning accordingly for future budget.

Answer: 1

Understanding the environment

The press under consideration is a very old press set up to cater to the specific needs of the university. Employees of this press have worked on the age old techniques of printing and are new to the concept of using computers for their purpose. The use of computers is basic related mostly to the Office automation tools for the purpose of record keeping and printing of hand calculated invoice. The job of completing the printing related orders is carried out on old and heavy machines.

Flaws with the existing system

1. ***Dependency*** – The existing system functions with the authorization of the manager only. The data is first authenticated by the manager for each activity and its process before sent to the next activity or process. From the SSAD view or the data and process oriented view, the flow of data is heavily dependent on the manager of the printing press for all communications with the other external entities and managing the work flow between the internal entities so as to complete the orders and other activities. This flaw in the system is taken care in the OOAD methodology, where the situation is analyzed as per the various users of the system and their role and responsibility.

2. ***Complexity and time involved in preparing the invoice*** – The information regarding papers and their costs as well as the various processes involved in the completion of any order, is complicated. Many parameters related to the type, ink, colours etc have to be taken care of while preparing the invoice to be sent to the departments. In the existing manual system, the user has to retrace all the various processes and their costs maintained in various registers to calculate the total amount for the completion of the order.

3. ***Large amount of paperwork*** – Maintenance of different registers related to various printing processes like order register, stock register, daily issue register, costing sheet details and bill register, is required in the existing system. Any query from the departments requires the user to refer several such registers.

Purpose of automation

1. Provide better customer services by reducing process complexity and time involved in preparing
   the bill.

2. Improves effectiveness by analyzing the data.
3. Generation of different reports to help in understanding and analyzing the allocation of various funds to different departments.
4. Creating history files for future purpose of data mining.

Implementation issues

1. Training of press personnel.
2. Installation
3. Parallel runs
4. Maintenance

Software Quality considerations

1. A minimum stand alone machine which should be enough to provide the automation of various processes from the SSAD point of view. A network of computers and the internet facility is ideal for the implementation of the software system developed using the OOAD methodology.
2. The press personnel have basic computer literacy, so the new envisaged software system should be easy to use and understand as well as provide error recovery and better help.
3. Investment for this system will be done by the university, so there is a need to incorporate minimum expenses.
4. Maintenance of the software system needs to be easy and simple.

Considering all these issues for the existing system of printing press and the expected software system to be developed for it, we have focused our case study on two analysis and development methodologies ie Structured System Analysis & Design (SSAD) and Object Oriented Analysis & Design (OOAD). SSAD has a process / data flow oriented view whereas OOAD has object oriented view. Here, objects refer to the various entities involved in the functioning of the system.

The various diagrams related to the SSAD and OOAD approach are presented in the following sections so as to understand the approach of the methodology and implement one as suitable to customized software system needs, objectives and situations.

SSAD Methodology

The entities considered here are the external entities which interact with the system but are not a part of the system. The initial view of the system is known as the context diagram of the system which shows the system in its entirety and shows the data interactions between the external entities and the system. The next view is known as the top level diagram that expands the single system representation into detailed process level representation and the data flows between the processes.

Context Diagram

1. The context diagram depicts the entire system as a single process. It is also known as the 0 level DFD.
2. The context diagram shows
   a. All external entities like department, supplier and university office of the system.
   b. All data entering / leaving the system which is useful for collecting input / output documents.
   c. The interaction between Department and University ie related to the forwarding of the invoice is not shown here for technical considerations; but it is taken care by mentioning the acknowledgement for invoice by the university to the system.
   d.

Q.1.1: Draw a Context diagram for INU Printing Press?

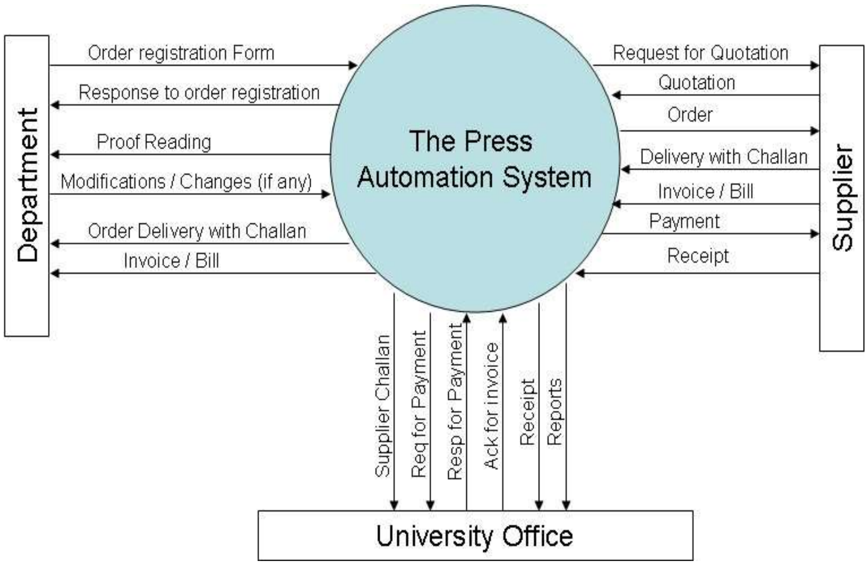


Figure 1. Context Diagrams

Existing First Level DFD

1. Keep the external entities unchanged.
2. Prepare the list of processes and data stores.
3. The logical DFD consists of 5 main processes like order registration, the actual printing process, inventory process, billing and authorization. The manager being a part of the system and also the user, is not shown as an external entity, but the role is taken care of by the use of the authorization process.
4. The costing sheet generated at the end of the press production process contains information about the order and the material ie the inventory required for processing the order.
5. A sample of the order placed by the department is first prepared for verification and sent to the department by the order registration process. This sample is given by the press production process, which being a physical entity is not shown on the DFD.

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

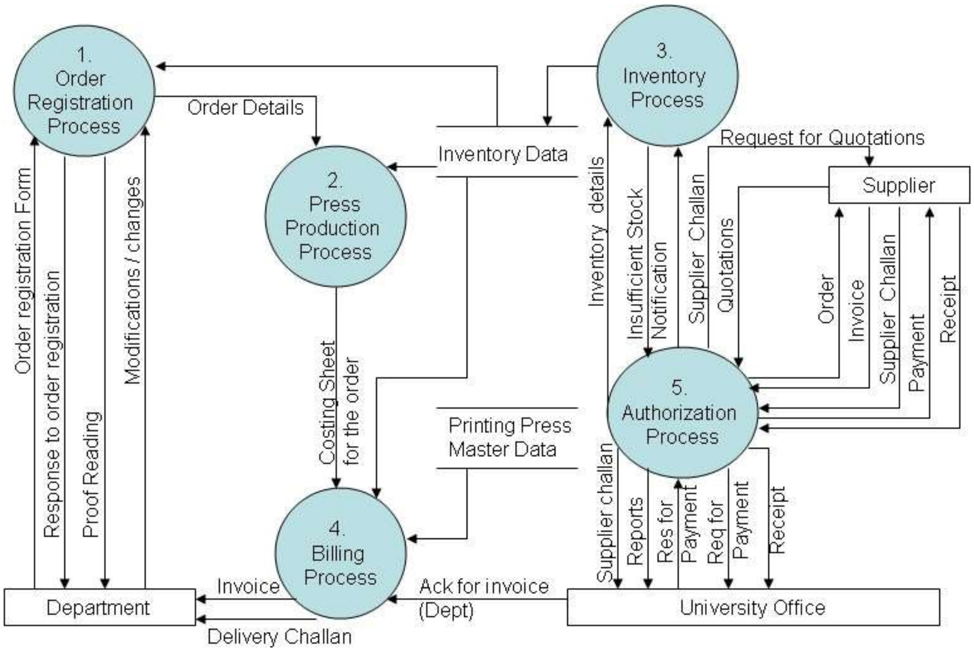


Figure 2. Existing First Level DFD

Proposed System : First Level DFD

1. The First Level DFD represents the system in detail with its processes that together constitute the entire system. Each individual process is numbered from 1 to n where n ideally should be between 5 to 9.
2. The printing press is a part of the University and is basically catering to the various departments of the University. Hence handling the other entity like individuals and organizations is out of scope of this system.
3. The new automated system can be improved by doing away with the authorization process as it can be taken care by assigning a different user role.
4. The corresponding data flows change accordingly and the execution of order can be done at a faster pace.

5. The logical DFD talks of the previously mentioned external entities along with the 2 important internal data stores ie inventory data and the printing press master data. The press master data store is related to the equipment and material that is required for the completion of any printing job for eg blocks, inks, letters etc.

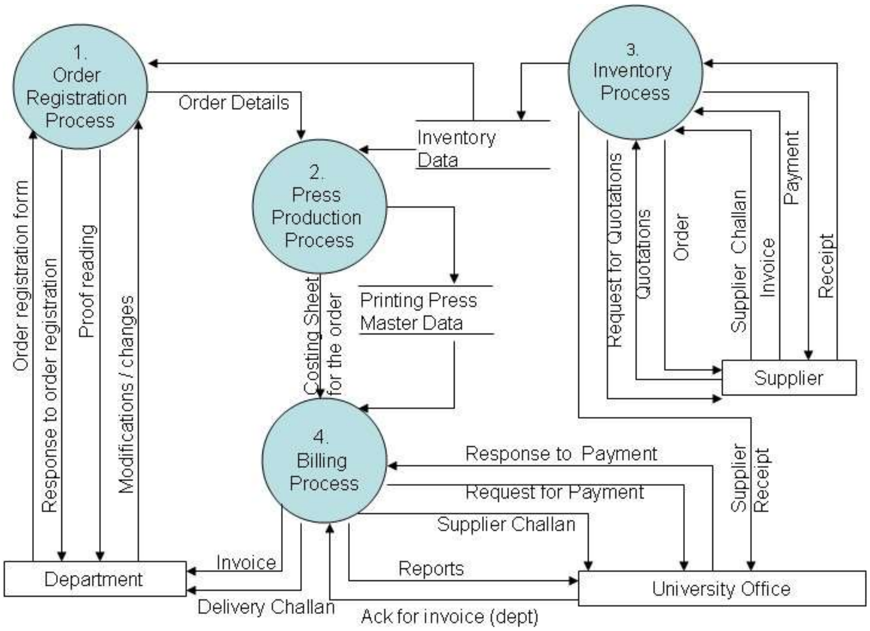


Figure 3. First Level DFD

First Level DFD (Physical)

1. The above DFDs were logical in nature where interaction between processes was shown considering that a process sending information to another process does so only after processing the information received from the external entity.
2. The given diagram is based on the actual situation where information once received by a process is stored in the data store. Any process now requiring this information will take it from the data store.

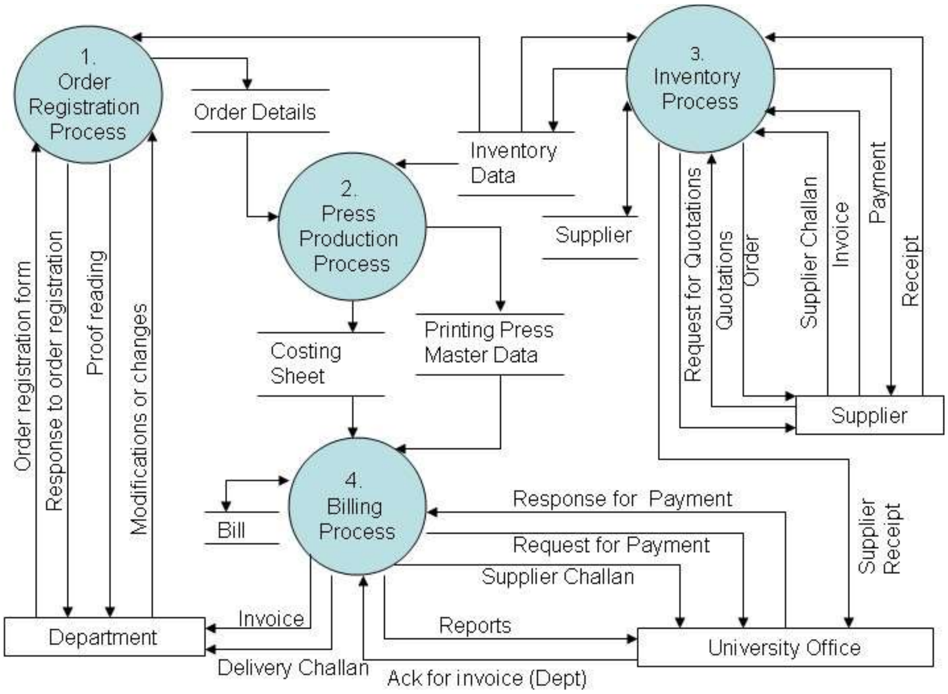


Figure 4. First Level DFD (Physical)

Second Level DFDs

The Second Level DFD represents the details of individual processes that have been defined in the First Level DFD. The sub processes are numbered from 1.1, 1.2, … and so on. Such detailed data flow at times defines from the logical working to the physical working of a system. Internal data stores additionally may be mentioned so as to complete the data input, storage and output.

In the printing press system, the following processes have been considered for further expanding as sub processes.

Second Level DFD for the Order Registration Process

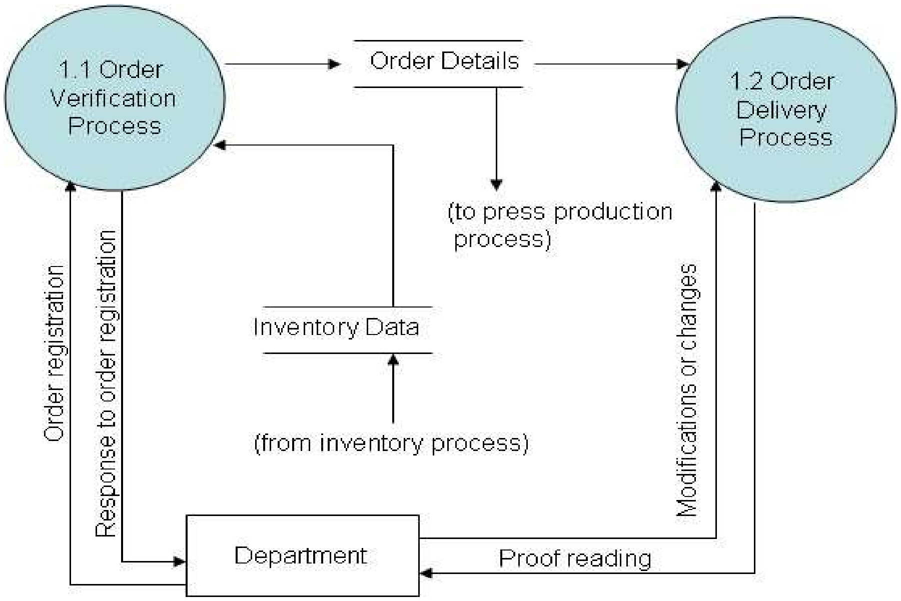


Figure 5. Second Level DFD for the Order Registration Process

Second Level DFD for the Press Production Process

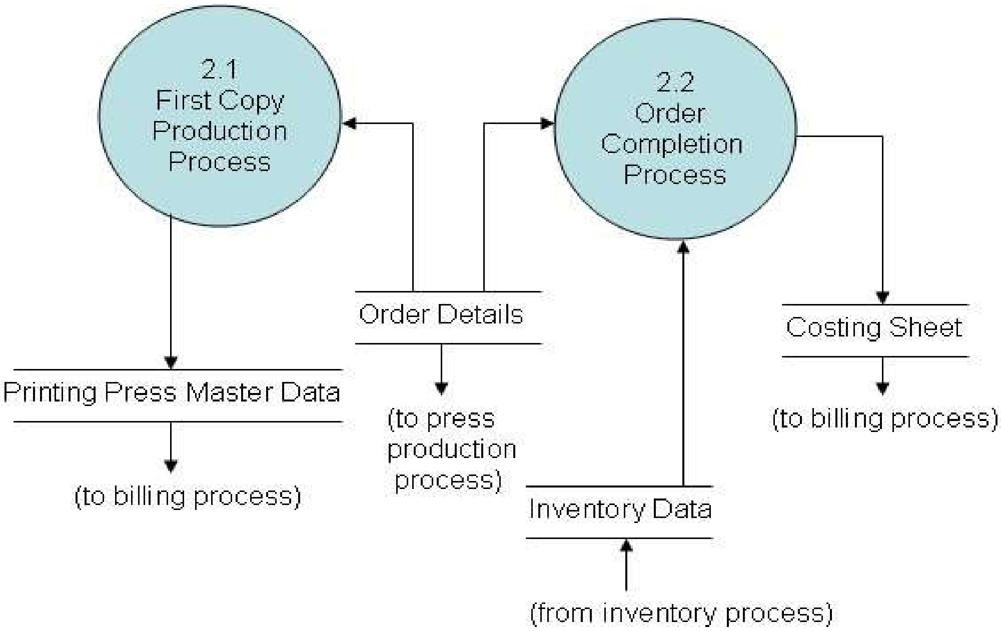

2.1 First Copy Production Process
2.2 Order Completion Process
Order Details
Costing Sheet
Printing Press Master Data
(to press production process)
(to billing process)
(to billing process)
Inventory Data
(from inventory process)

Figure 6. Second Level DFD for the Press Production Process

Second Level DFD for the Inventory Process

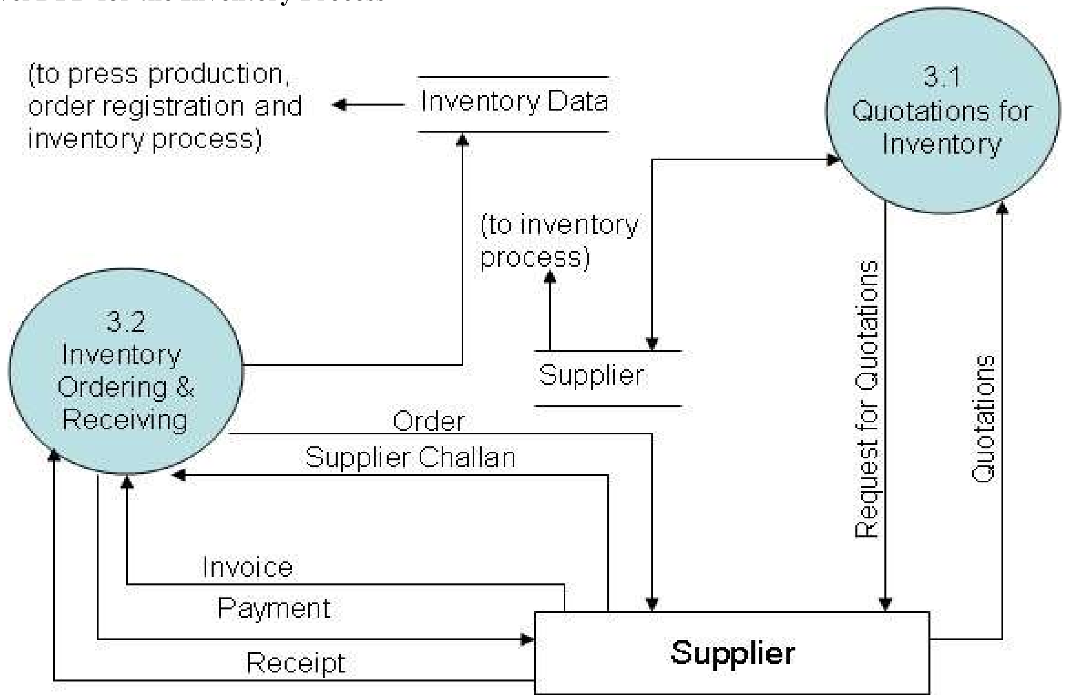


Figure 7. Second Level DFD for the Inventory Process

Second Level DFD for the Billing Process

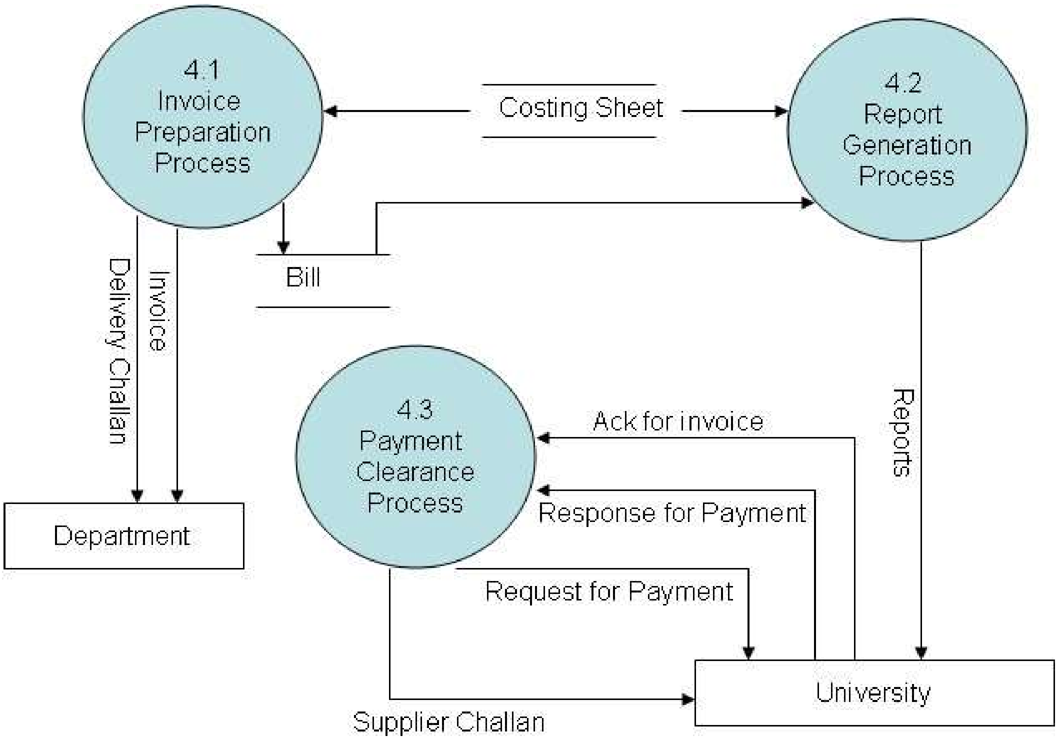


Figure 8. Second Level DFD for the Billing Process

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

Answer 2.1

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**. ... It is always possible that a **test** have overlooked could discover further problem with the system.

Q.2.2: Define the following terms:

1. Unit Testing
2. System Testing
3. Black Box Testing
4. White Box Testing

- ## Unit Testing

  UNIT TESTING is a level of software **testing** where individual units/ components of a software are tested. The purpose is to validate that each **unit** of the software performs as designed. A **unit** is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

- **System Testing**

  SYSTEM TESTING is a level of **testing** that validates the complete and fully integrated software product. The purpose of a **system test** is to evaluate the end-to-end **system** specifications. ... **System Testing** is actually a series of different **tests** whose sole purpose is to exercise the full computer-based **system**.

- **Black Box Testing**
  Black Box Testing is a software **testing** method in which the internal structure/ design/ implementation of the item being **tested** is not known to the **teste.**

- **White Box Testing**
  White Box Testing is a software **testing** method in which the internal structure/ design/ implementation of the item being **tested** is known to the **tester**.

Question 3:

Q.3.1: Briefly describe the three main types of software maintenance. Why is it sometimes difficult to distinguish between them?

Answer:

Types of 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.
- ✧ Maintenance to repair software faults
  - ✧ Changing a system to correct deficiencies in the way meets its requirements.
- ✧ 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.
- ✧ Maintenance to add to or modify the system's functionality

  Modifying the system to satisfy new requirements

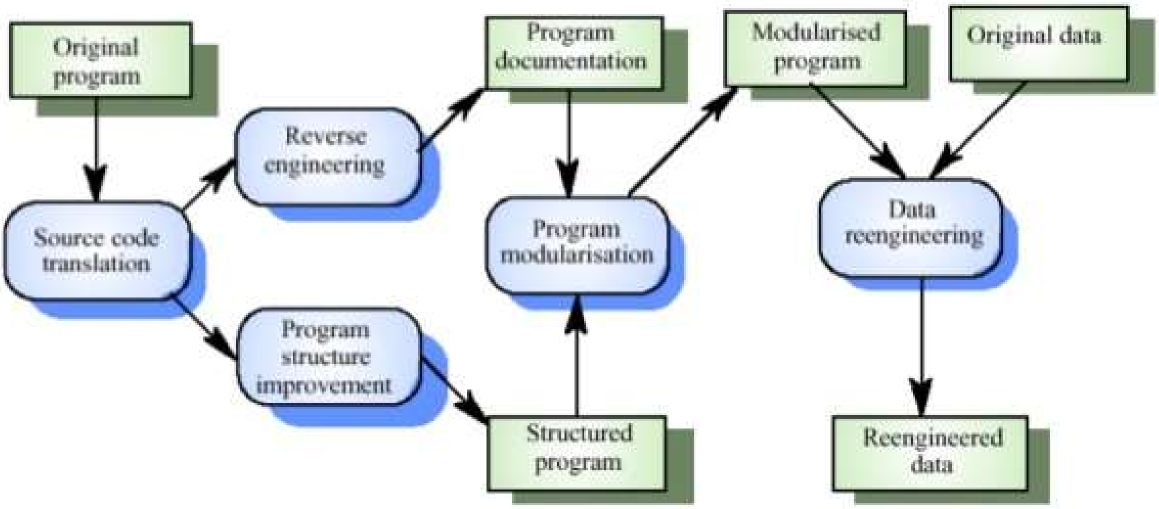
1. Bug Fixing - this is repairing faults found in the software after it has been launched. The bugs are there possibly because testing was not as thorough as it should have been or clients have exposed bugs by using the software in unexpected ways. Coding errors, design errors, and requirement errors are the least, middle, and most expensive to correct, respectively.
2. Modifying software to work in a new environment - when the hardware or platform that the system was built to run on changes, then the software must change as well in order to be compatible and avoid being obsolete.
3. Implementing new or changed requirements - software must be updated or changed so that it conforms with any new requirements.

- ✓ It is sometimes difficult to distinguish between the different types of maintenance because they are often given different names and also because faults that arise within a system can maybe have overlapping maintenance requirements.

Q.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?

Answer: 3.2

The re-engineering process



Cost of Re-engineering

✧ 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 a problem with old systems based on technology that is no longer widely used.

The cost of re-engineering depend on 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 costs are:

i. The quality of the software to reengineered: The lower of the quality of the software and its associated documentation(if any), the higher reengineering costs. ii. The tool support available for reengineering: The use of CASE tools to automate most of the program changes is normally cost effective to re-engineer a software.

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

iv. The availability of expert staff: If the staff responsible for maintaining the system can’t be involved in the re-engineering process, this will increase the costs. System re-engineers will have to spend a great deal of time understanding the system.

*************THE END***********************