Name	M. Zeeshan
ID#	14882
Section#	B
Departement	BS(SE)
paper	Software Engineering
Exam	Final
Date	22/06/2020
Semester	4th

Instructer:-

GHASSAN HUSNAIN

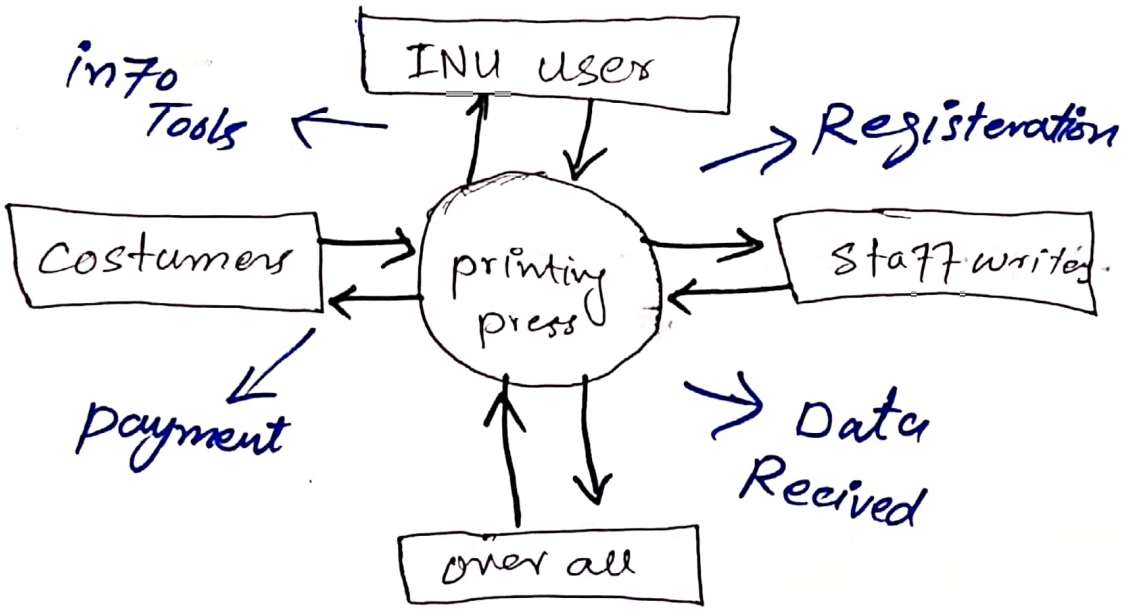
Question # 1:-

⇒ Case Description :-

↳ INU printing press, a part of a University undertakes printing jobs --?

Q 1.1: Draw a context diagram for INU printing press?

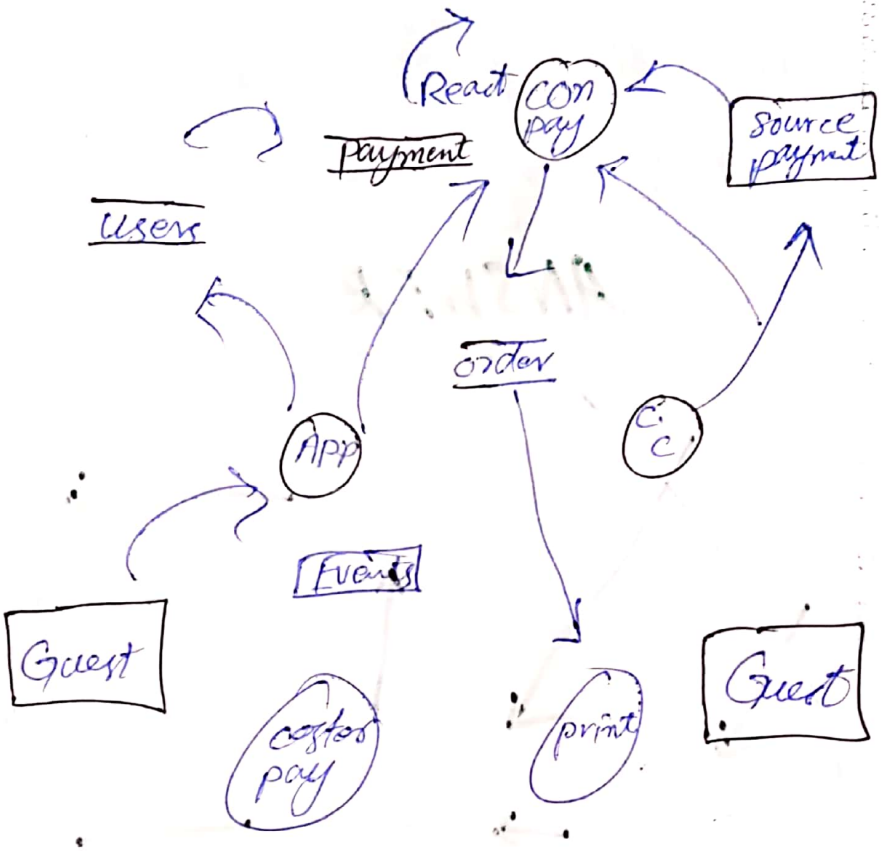
ANSWER



↳ context diagram for INU printing press.

Q 1.2:- Draw a level 1 Data Flow Diagram (DFD) for the above case study?

ANSWER

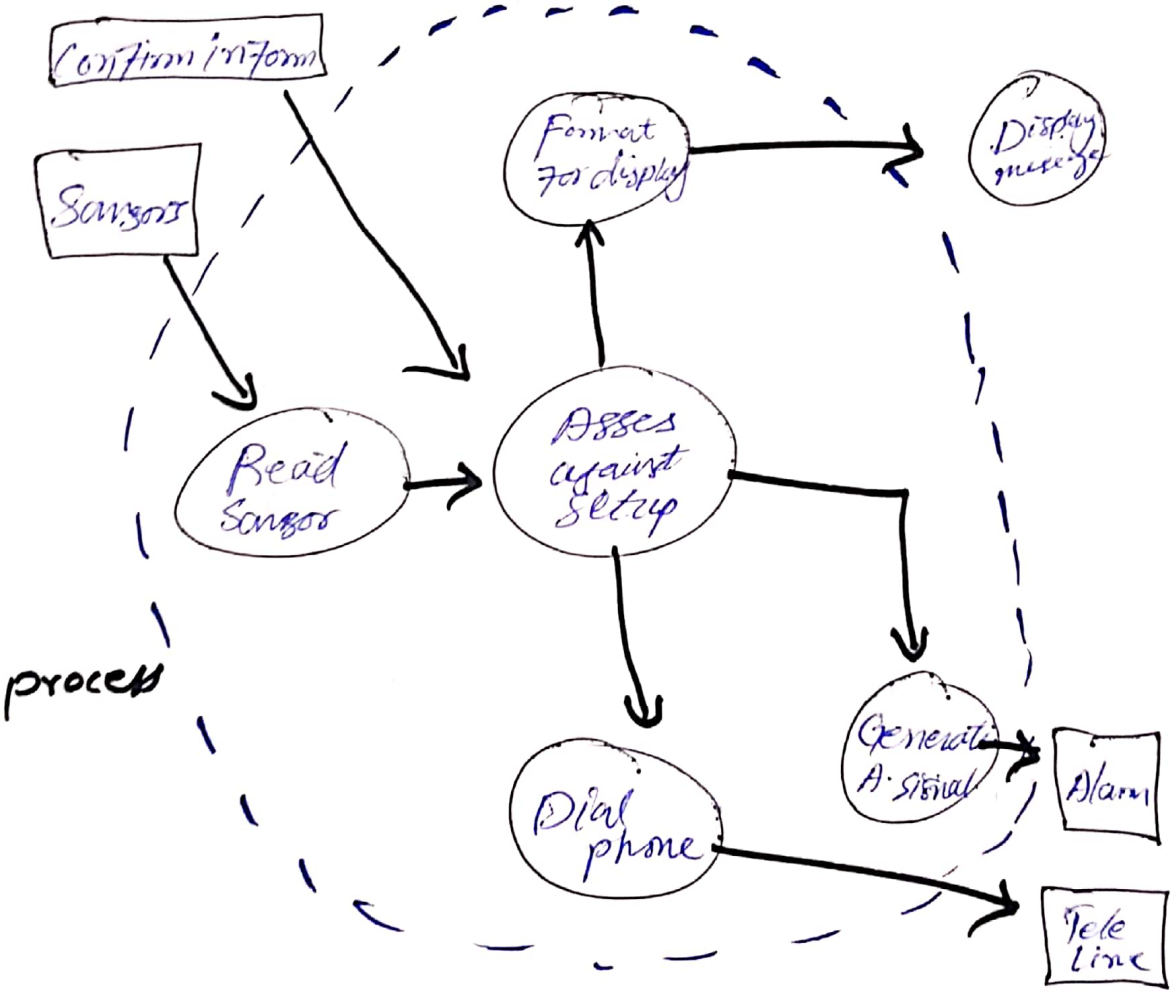


↳ Above is the level 1 Data flow Diagram. that Allowed to combined several data flows from lower level diagram at a higher level under one data flow to reduce the Risk.

(3)

Q1.3: Draw a level 2(DFD) for the order Registration process, press production process, inventory process and Billing process?

ANSWER



↳ level 2(DFD) diagram recived information via central panel and proces on printing press.

Question#2

Q2.1:- Explain why testing can only detect the presence of error, not their absence?

ANSWER

⇒ Testing can only the presence of error, not their absence because the main goal of the testing is to observe the behaviour of the particular software and to checks whether its sequirement Expectation or not.

Testing is a part of broader process of software verification. 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 demonastrate the fruits other than specified in every circumstance.

it is always possible that a test: haive overlooked could discover further problem with the system.

Assum that exhastive testing of a program where every possible valid input is checked, is impossible true for all But arrivial programs. Test cases either do not reveal a fault in the program or reveal a program fault.

The goal of software testing is to observe the software behaviour to meet its sequirement expectation.

in software engineering might be harder client's expectation may be vague or unclear.

Q 2.2:- Define the following terms?

1 - unit Testing.
2 - system Testing.
3 - Black Box Testing.
4 - white Box Testing.

ANSWER

→ 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 ... Tester).

⇒ Unit Testing is a level of software testing where individual units/components of a software are tested. the purpose of is to validate that each unit of the software performed.

Testing is the process to ensure that an Application is as per the customers requirement.

During the course, the tester may find Errors and Bugs which they can report to developers, so they can detect the Error but cannot ensure the absence of Error.

⇒ Software Tester:-

- ✓ Find Bugs as easly as possible and make sure they get fixed.
- ✓ To understand the Application well
- ✓ study the Functionality in detail to find where the bugs are likely to occur.
- ✓ Create test cases in such a way that testing is done to uncover the hidden bugs and also ensure that the Software is usable and reliable.

2 System Testing:-

→objective:-

→ To verify that the system components perform control functions.

→ To perform inter-system test.

→ To demonstrate that the system performs both functionality and operationally as specified.

→ To perform appropriate types of Test Relating to transaction flow, installation, Releabality, Regressionete.

→When:-

↳ After Integration Testing.

→input:-

→ Detailed Requirements of External Application design.

→ Master test plan.

→ system test plan.

→ **output:-**

↳ system test Report.

→ **Who:-**

↳ Developement team and users.

→ **Methods:-**

↳ problem/configuration Management.

→ **Tools:-**

↳ Depends.

→ **Education:-**

↳ Testing methodology.

→ system testing is a level of Testing that viladates the complete and fully integrated software products. the purpose of a system Test is to evaluate the end-to-end system Specification. only the software is only one Element of a larger computer-based system.

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.

→ cover all combined parts of a system.

→ Test are data driven (Tests are based on putting some data to checked the system).

→ it uncover
- incorrect or missing functions.
- interface Error.
- Error in data structures or External database access :
- performance error.
- initialization and Termination Errors.

⇒ **Types of Black Box Testing:-**

- **Functional Testing:-**

↳ Black Box type testing geard to functional

Functional requirements of an Application.

- Done By Testers.

Some others types also have Black Box Testing. eg

- system testing.
- End-to-End Testing.
- mutation testing etc.

→ Black Box Testing is a Software testing method in which the internal structure/design/ implimintation of the items being tested is not known to tester. these tests can be Functional or non-Functional though usually Functional.

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's ensure.

- All independent paths within a module have been Exercised at leasts once.
- Excersied all logical decession on their true and False sides.
- Excute all loops at their Boundries and within their operational Bounds.

- Excersise internal data structures to ensure their validity.

→ White Box testing > Transport Box Testing code - Based Testing or Structural testing is a Software testing method in which the internal structure/design/ implimintation of the item being tested is known as Tester.

Question# 3

Q3.1:- Briefly Describe the three main types of software maintenance, why is it sometimes difficult to Distinguish Between them?

Answer

→ **Software Maintenance:-**

↳ modifying a program after it has been put into use.

→ The terms is mostly used for changing custom software. Generic software products are said to evolve to create new version.

→ Maintenance does not normally involve major changes to the system's architec

→ Three main types of maintenance:-

1)→ Fault Repairs:-

↳ coding errors are usually Relatively cheap to correct, design errors are more Expensive as they may involve rewriting several program components. Requirements errors are the most expensive to Repair Because of the expensive system redesign which be necessary.

2) Environment Adaptation:-

↳ This type of maintenance is Required when some aspects of the system's environment

such as the hardware, the platform operating system, or other support software changes the. Application system must be modified to adapt it to cope with these environment changes.

3) Functionality Addition :-

↳ These type of maintenence is necessory when the system Requirements change in Response to organizational or Busines change. the scale of the change required to the software is often much greater than for the other types of maintenance.

(18)

⇒ Difficult to Differentiate :-

↳ Distinguish Between these types of maintenance, when the system adapt to new environment, then add functionality to take advantage of new environmental features, software fault are often exposed Because users use the system in unanticipated ways, these types of ~~xx~~ maintenance for fault Repair. "Adaptive" maintenance Sometimes means adapting to new environment.

"perfictive maintenance" Sometimes mean perfecting the software by implimenting.

Q 3.2:- What are the principal factors that affect the costs of system seengineering? also briefly Explain the Reengineering process with the help of diagram?

ANSWER

→ **System re-engineering:-**

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

→ Applicable 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.

→ Advantage of re-engineering :-

→ Reduced Risk :-

↳ There is a high Risk in new software developement, there may be developement problems, staffing problems and specification Problems.

→ Reduced Cost :-

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

→ Re-engineering cost factors :-

↳ The quality of the software to be reengineered.

→ The tools supports available for Re-engineering.

→ 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.

→ Refactoring is the process of making improvement to a program to slow down degradation through change.

→ Re-engineering process activities :-

↳ Source code Translation.

- convert code to a new language

→ Reverse engineering.

- Analyse the program to understand it.

→ program structure improvement.

- Restructure automatically for understand.

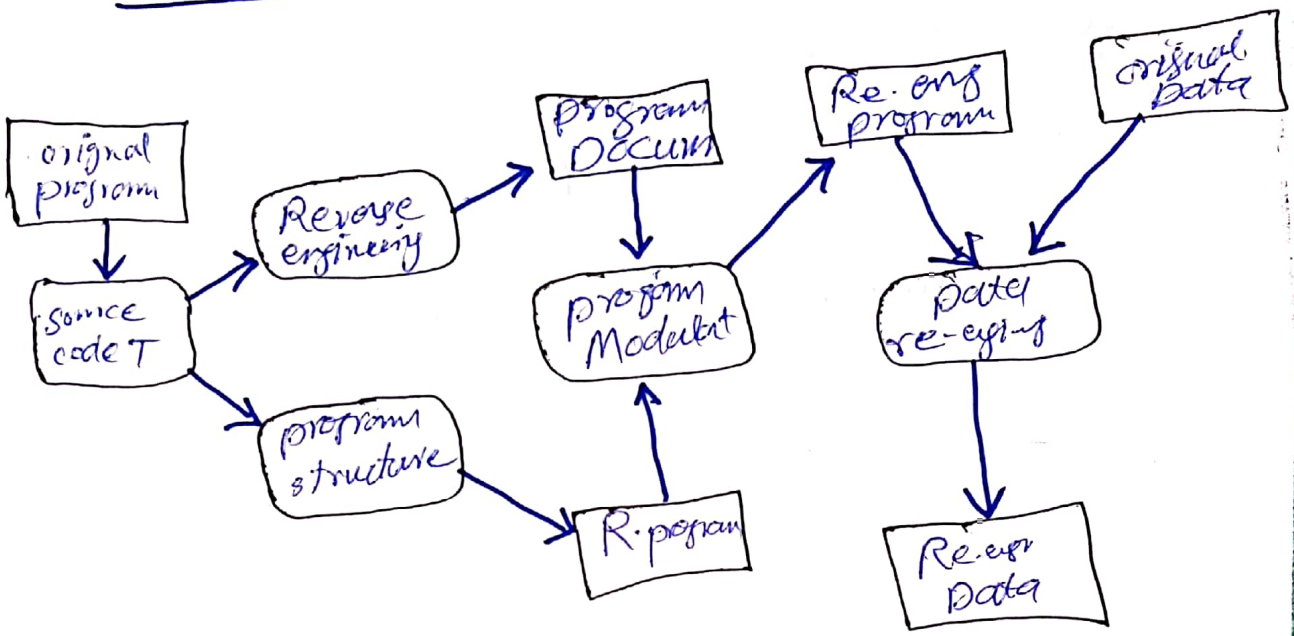
→ program modularisation.

- Re organise the program structure.

→ Data Re-engineering.

- clean-up and restructure system data.

→ Re-engineering diagram :-



→ inventory Analyse :-

↳ inventory can be nothing more then a spreadsheet model containig information

→ Document reconstructing :-

↳ Document must be updated.
→ it may be not necessary to purify doc.

→ Reverse engineering :-

↳ process of design Recovery.
→ Reverse engineering tools extracts data.

→ code Recanstructing :-

↳ To accomplish code Reconstructing
→ the source code is analysed.

THANK YOU SIR