Lecture 1 –04/01/21

business

Organization

Client

BT

Testing

There are 2 approached where we can deliver Refined product to customer:

* Design fine/robust code🡪 DEV🡪This approach is quite difficult
* Remove/eliminate all the defects🡪Testing🡪 This approach is Practical and logical

Lecture 2 –05/01/21

**SQA (Software Quality Assurance)(CCCTR)**

SQA is a very important attribute. There are several important parameters

1. Customer Requirement: (on what purpose)
2. Customer Expectation:

Privacy: Password should be secure

Performance: Once u click on a button it should give result in few seconds)

Security

Part of Testing

1. Cost of Software:
2. Timely Deliverable:
3. Risk Management:

If something went wrong, it should be solved on time.

Q. Why SQA?

To measure and monitor the strength of dev process.

Set of activities ensuring quality of product.

Why are ur roles and responsibities in sytem as a tester?

1. Customer requirement Analysis
2. Test plan design
3. Test scenario identification
4. Test case design
5. Test case review
6. Traceability matrix (Requirement mapping)
7. Test case Execution
8. Defect logging and reporting
9. Analysis of defect and RCA
10. Identification of defect
11. Client interaction

Why did you join as a tester engineer/Why Quality/ Why u prefer/ What is significance of testing?

The world moves from customer requirement to customer satisfaction

In old times market as monopoly, I mean there is no such competition.

But in the current scenario market is absolute oligopoly I mean there is huge competition.

That means organization must deliver quality product to survive in the market.

It means to achieve Quality product; they must focus on testing.

Let’s take an example:

Nokia used to rule the world, during Symbian era. But if technology gets changed from Symbian to Android, even Nokia couldn’t survive.

The technology changes from 2g to 4g and moving toward 5g

We must adapt to the environment to survive. It means Quality is the only parameter for your existence.

Decision Making:

Nokia didn’t adapt to Android and failed.

Software Project: A software related issue is solved by s/w engineer through a

s/w engineering process.

Who is a good test engineer?

Who understands the SDLC completely and perfectly?

Who understands the point of view of customer?

Ability to communicate with technical and non-technical people.

How can u say that tester has more analytical knowledge than developer?

When client requirement comes developer develops the application according to customer requirement but tester then tests the application if it’s working according to customer requirement

Tester will have to think about the positive and negative scenarios basically we follow 3D approach.

**SDLC (Software Development Life Cycle)**

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software’s. The SDLC aims to **produce a high-quality software that meets or exceeds customer expectations**, **reaches completion within times and cost estimates**

SDLC🡪 LCD+LCT

Basically, it consists of 3 Generic phases:

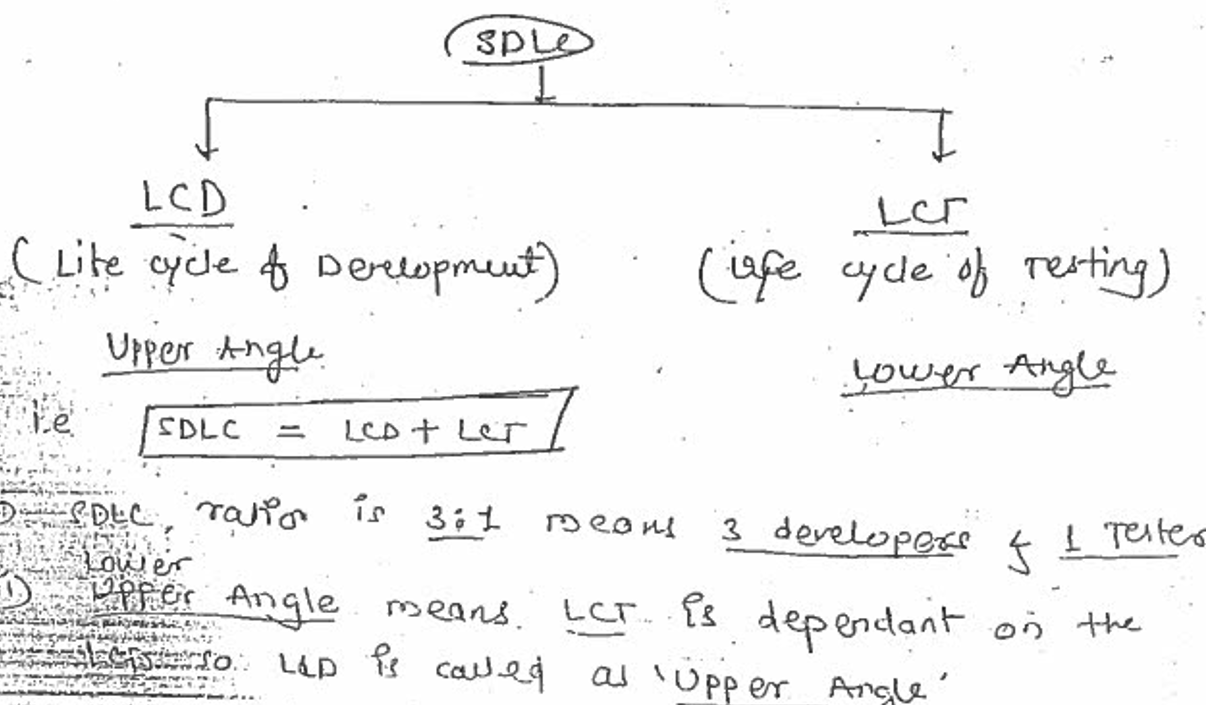
|  |  |  |
| --- | --- | --- |
| **What** | **How** | **Modification** |
| Customer Requirement | Design | Correction |
| Analysis | Coding | Adaptability |
|  | Testing | Reengineering  (changes done in prev product) |
|  |  | Enhancement  (Future Requirements) |

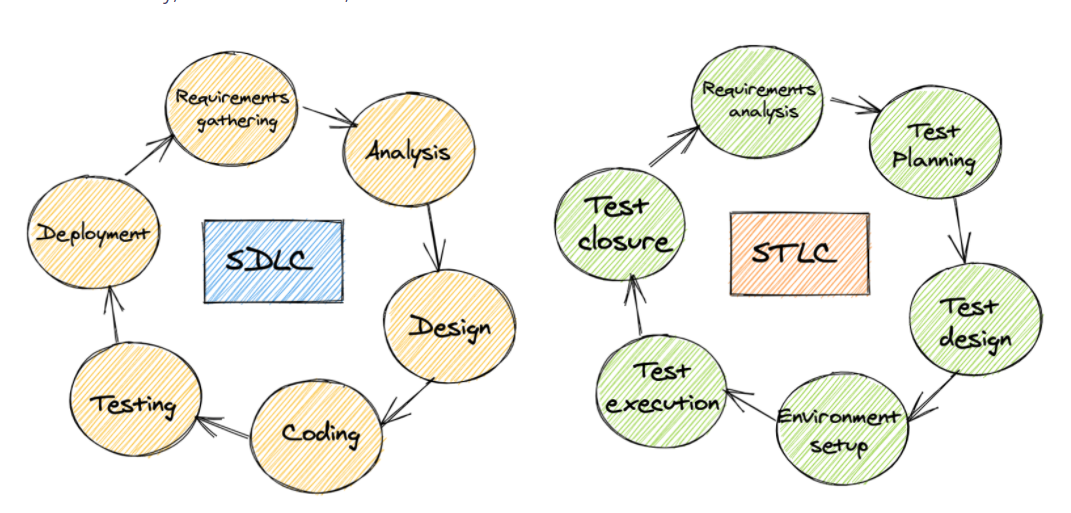
SDLC

|  |  |
| --- | --- |
| Stages of SDLC | Documents |
| Information Gathering/Customer Requirement | BRS (BA designs it) |
| Requirement Analysis | SRS/CRS(designed by Architect) |
| Design | HLD (designed by Architect) ,LLD |
| Coding | Programming |
| Testing | test Document |
| Maintenance | Support document |

Note: Testing happens on all the above stages.

Lecture 3 –06/01/21





Requirement

Order\_id: 408-3227689-4127586

Order\_id is a 17 digit numeric. 17+2 = 19

Amazon\_id,Product\_id and server\_id will be separated by ‘-’-.

order\_id

|  |  |  |
| --- | --- | --- |
| Amazon\_id | Product\_id | Server\_id |
| 408 | 3227689 | 4127586 |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| columns | Datatypes | Length | Constraint | Mandatory |
| Order\_id | String |  |  |  |
| Cell |  |  |  |  |
| Cust\_name |  |  |  |  |

**Analysis**:

Min Max

Note: As per my understanding the complete backend structure will be governed with 2 attributes. The most important information is to analyse or identify the nature of data in terms of static or dynamic.

if data is static, min and max value is same

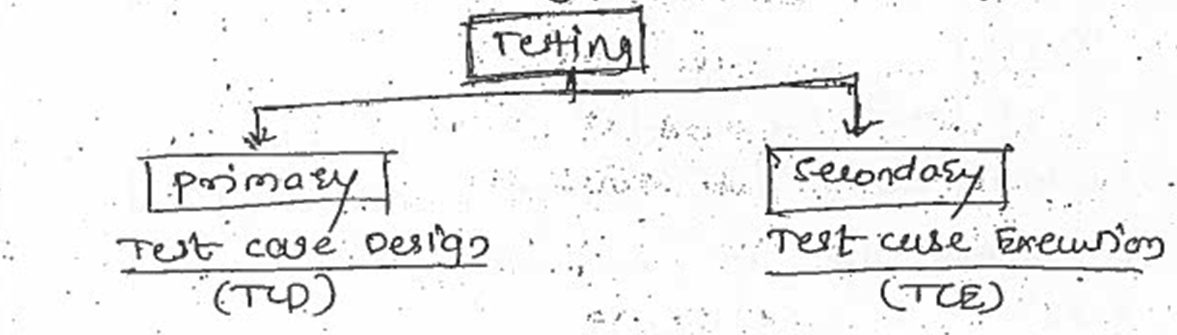
if data is dynamic, min and max value is not the same so it should be analysed by us.

Eg. cust\_name can be min=1 and max=50

Data types: int, string

Q. Testing comes before coding or after coding?

Testing starts side by side with coding



BRS

SRS

Design

Testing

Test case design

Test case execution

coding

Testing

Database

application

* Graphical representation of SDLC
* Graphical representation is best way of communication
* Images are better understandable
* Testing is done by many people in organization.
* The testing done by tester is 3D

**Fish Model**

Client Sow BA Designer Developer

Python SRS Design Coding

maintenance

Programming

BRS

Black Box Testing

Change Request

|  |  |  |
| --- | --- | --- |
| Review | Review Prototype | Whitebox testing |

|  |  |
| --- | --- |
| Static Testing | Dynamic Testing |
| Verification | Validation |
| 2 Dimensional | 3 Dimensional |

Fish model defines mapping between different stages of development and testing.

Black Box Testing/System and functional Testing

System and functional Testing:

Function will work if the system is working.

SOW: statement of work

It’s a business document that covers every nuance of the agreement between the client and an outsourcing company to boost cooperation and minimize the chance of conflict or confusion between organizations.

Diagram, timeline

Description automatically generated

TDD: Technical Design Document

SCD:

**What is the difference between Static Testing and Dynamic Testing?**

**Static (Verification)**

It is verification testing

We test the requirement document & design document prior to the s/w being developed.

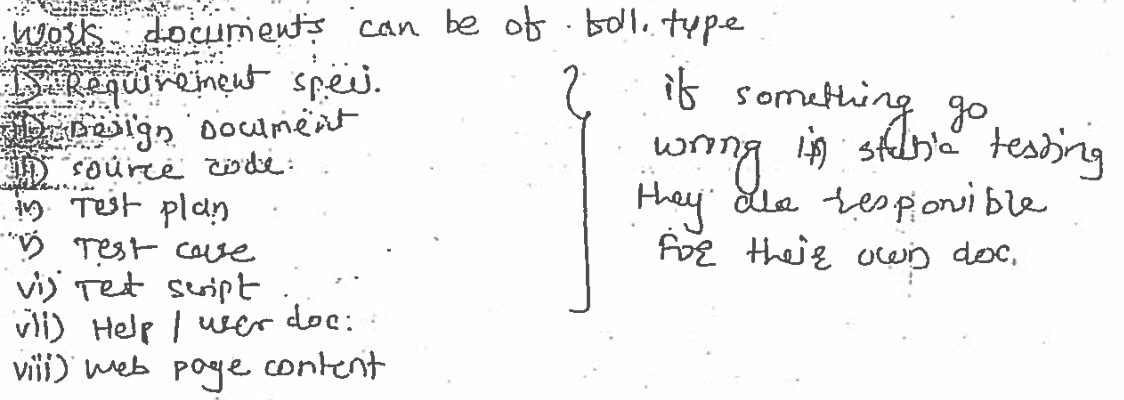
Testing without giving any input

This testing is done during analysis and review phase.  
It’s about prevention.

Takes less time than dynamic

Here no code is executed, manually the code and design documents are checked to find error.

Main objective is to improvise the s/w by finding errors in early stage of SDLC.



Testing technique in static:

Informal review

Technical reviews

Walkthrough: led by author

Inspection: led by moderator

**Dynamic (Validation)**

We test the s/w being developed.

Testing after giving input & checking expected result.

This testing is done after full/part of code is developed.

It’s about cure.

Takes longer time than static  
Here code is executed, we check the functional behaviour of s/w system.

Memory/CPU usage and overall performance.

Main objective is to check if the product works as per customer requirements.

Testing technique in dynamic:

Unit testing

Integration testing

system testing

Lecture 4 –07/01/21

What is the difference between Verification and Validation? or

What is the difference between Static Testing and Dynamic Testing?

Now Que arises why is it known as Static or Dynamic:

In amazon the billing add and shipping address should be present

Requirement: billing add and shipping address should be same.

Requirement: billing add and shipping address should not be same.

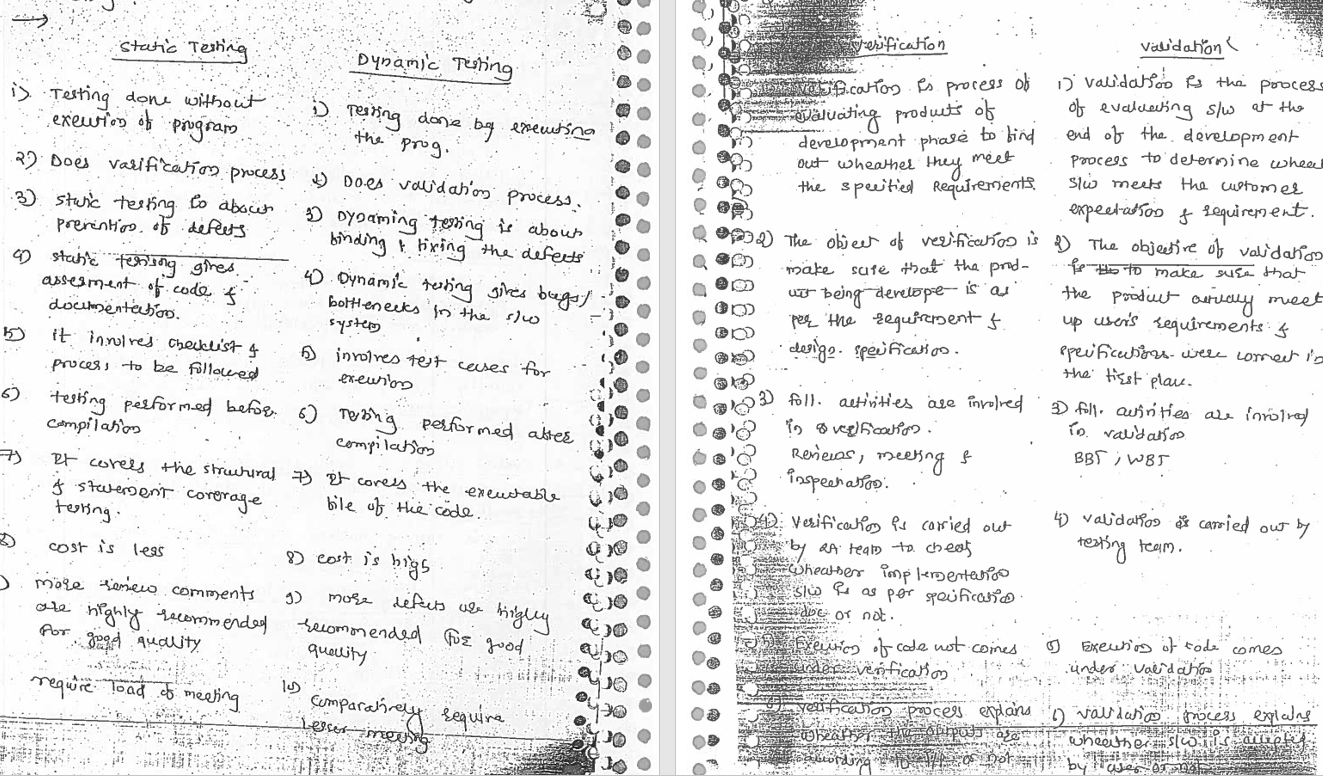
If we analyse and understand SDLC architecture, we will observe SRS, there will be review, design we will be reviewed, coding will be reviewed.

(so, this review is called as testing)

BA, Designer and developer are going to review their own code. BA will review the SRS report, Designer will review the DQ document and Developer will review the code. This is static approach of testing or white box testing.

But at the end during black box testing tester is going to review the whole process.

We are going to catch the issues in the requirement, design document or the code



Script

Manual Business test case Automation Approach

Approach Logic

Front end

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Database | ETL | BI | Unix | Web Service | System Integration |

**Lecture 5 –08/01/21**

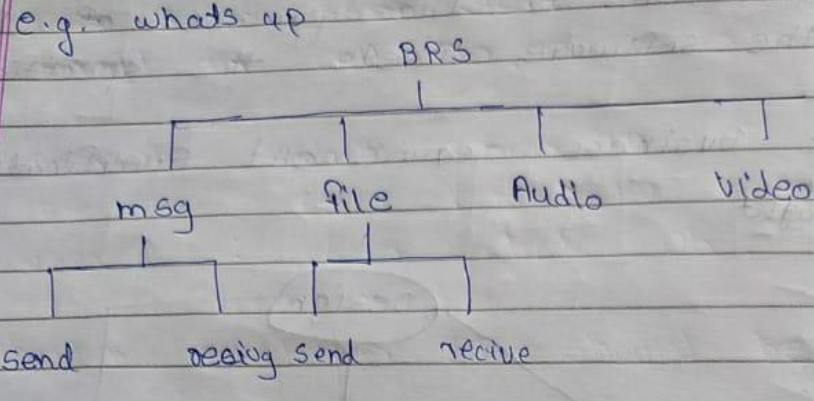
Q. If white box testing reviews the code then what is the importance black box testing?

Q. If white box testing reviews the code then why black box testing

* As per my understanding, Error comes for requirement, design as well as code design.
* WB testing checks the presence of defect in code but BB testing covers the presence of defect as well as the absence of defect.
* We have to focus on positive as well as negative validation.
* So, we can say that white box testing covers 2 Dimension Validation and we concentrate on 3D Testing.
* 3D means implementation of business logic.

Static and dynamic

* Static and dynamic are interdependent on each other
* Without static dynamic is not possible and without dynamic quality product is not achievable.
* Hence dynamic testing is one of the most significant elements in the SDLC process.
* Validation is he process of checking whether the specification captures customer needs.
* Verification comes under s/w quality assurance
* Validation comes under s/w quality control



Max 5mb and 250 characters

Sending file🡪 max 5mb ,<5, =5

Sending text🡪 max 250chars ,<250, =250

**Lecture 6 –09/01/21**

**SRS**- Software Requirement Specification

**CRS**- Customer Requirement Specification

**FRS**-Functional Requirement Specification

**BRS**-Business Requirement Specification

**BA-** Business Analyst

**Customer Requirement**

Cell no. Is mandatory

BA- Without filling cell no. end user can’t proceed further

BRS: The requirement of the customer to be developed as a software.

This document acts as a bridge between client and technical candidates.

A **Business Requirement Document (BRD)** focuses on the business perspective as it holds the details of the business solution for a project

Business requirements document also emphasizes on the needs and expectations of the customer. In simpler terms, BRD indicates what the business wants to achieve.  The BRD indicates all the project deliverable and the inputs and outputs associated with each process function.

* This document is designed by BA and is referred from BRS.
* This defines functional requirement to be developed and system requirement to be used.
* Like we need to develop a particular function by using system requirement.
* For online payment 🡪 we can use amazon pay/ gpay/ Phonepe

Software Requirement Specification

What is SRS?

Are u aware of SRS? What does it consist?

software requirements specification (**SRS**) is a **document** that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs.

It consists of:

Functional requirements: Business requirements / business conditions

Use case: SRS-Customer ID generation

Usecase: user must provide

UID: 4–8-character alphabets

Usecase 2:2 password must provide

UID: 4-6 numeric

Usecase 3:user select

Email max (40)

cell no. max (10)

after that click ok

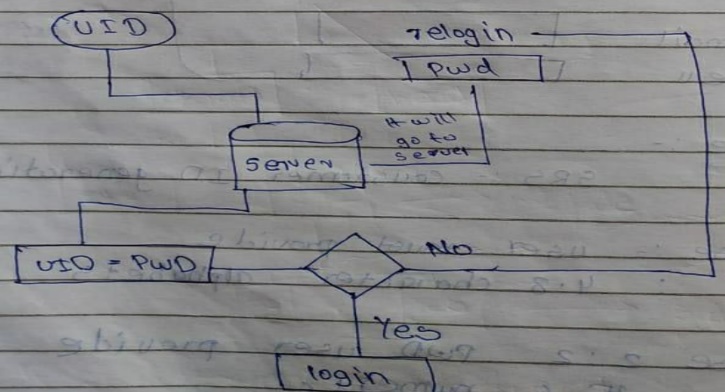
customer ID should be generated

Snapshot: How snapshot comes to picture? Where does the snapshots come from?

Html code or idoc (Accenture product) and designed by BA.

Functional flow diagram: The flow of data in system.

Whenever 2 or modules communicate the communication flow is called functional flow diagram



Module1 Module2

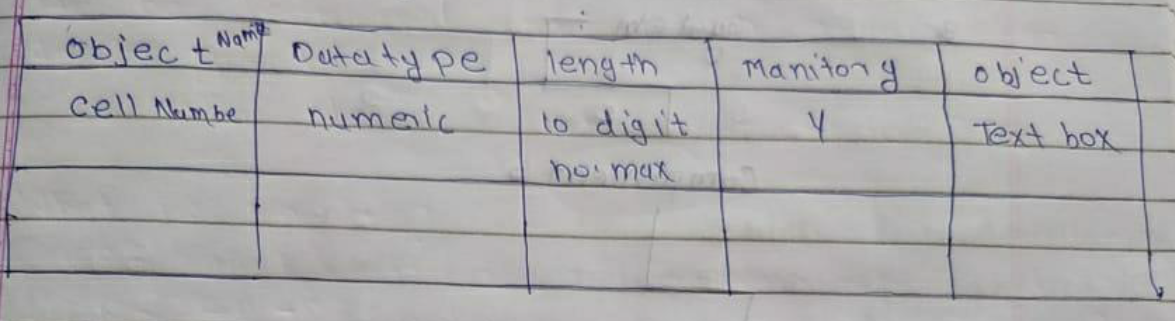
Request

Response

System/module details ecommerce, logistics, Payment

Environment Details:Hardware/software

**Data Model Design:**



Apart from SRS what are the other related documents required?

HLD: High level Design document

LLD: Low level Design Document

Message Sequence/Queue Service

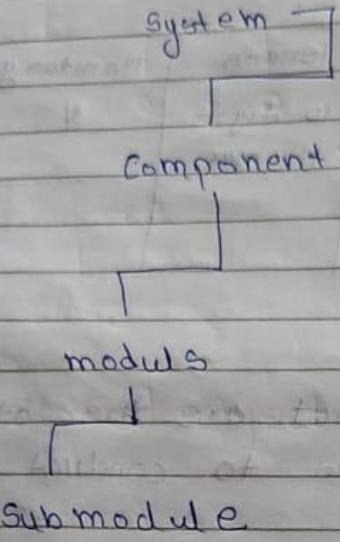
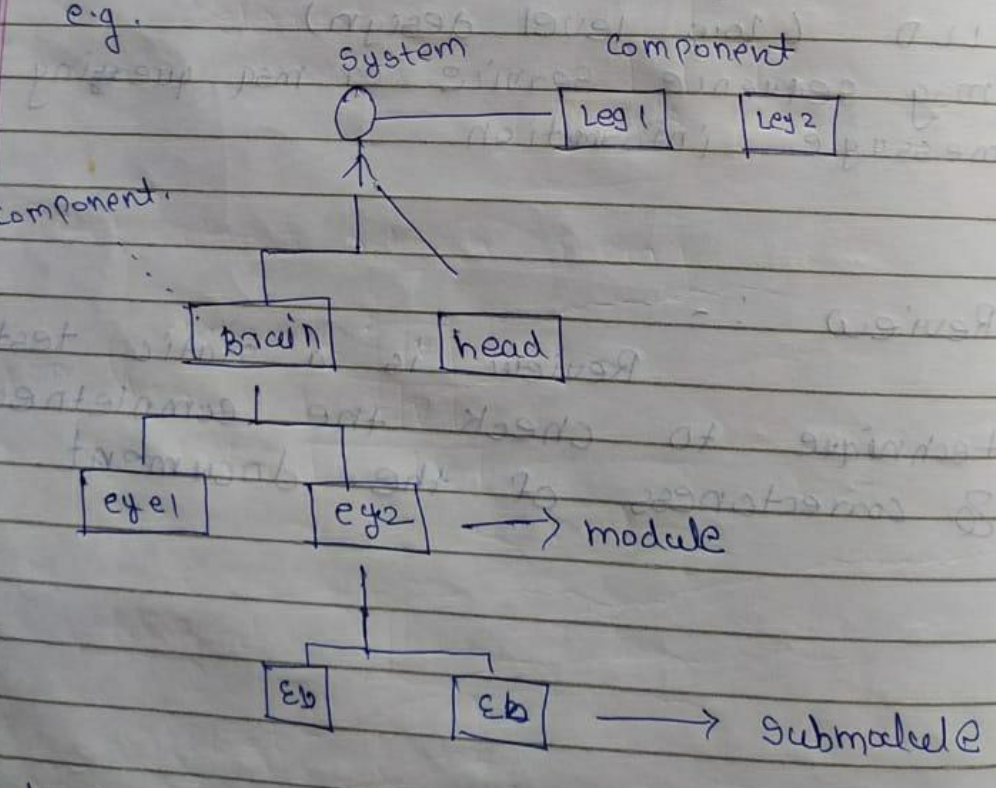
Review: static testing technique to check the completeness and correctness of code/document.

**Lecture 7 –10/01/21**

**High level Design document**

* It is also called as external design/ software architecture.
* It defines Hierarchy of all possible functionalities to be developed as a module/component/system.
* It is designed by Project Architect/solution designer

High-level architecture diagram depicting the components, interfaces, and networks that need to be further specified or developed. The document may also depict or otherwise refer to work flow and/or data flows between component systems.

**System**

* Application
* Component
* Module
* Submodule
* Element
* Attribute

**Low level Design Document**

* It is also known as Internal design/ detailed design
* It defines the static logic of every sub module.
* This is done by developer.
* Class diagram, object diagram, ER diagram [Entity relationship diagram]

Class has multiple objects

* It’s used to design internals of the individual modules identified during HLD i.e. data structures and algorithms of the modules are designed and documented.
* The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code.
* LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

**Class Object attributes**

Electronics🡪Mobile🡪 Samsung🡪16GB

Laptop🡪 Nokia 🡪 8 GB

**Review Prototype**:

It is a sample application without functionality

**Lecture 8 –12/01/21**

**Coding**

After completion of design and their review, developer writes the program by using various language such as java, python, etc to physically construct on s/w.

**White Box Testing/Code Review**

* It is also called glass box testing/clear box testing.
* During this test the developer checks the correctness and completeness of program in the terms of values and logic.
* Internal structure Is known

**Black Box Testing**

* We should know the logic behind the program.
* After completion of code design and their review (WBT), we validate the functionality of the application by using BB testing technique.
* It is a build level testing technique. During this test we validate the functionality of the application w.r.t customer requirement.
* What I want to say here is that we focus on 3D mechanism to perform BB testing.
* I mean to say during this testing we concentrate on functionality of the system wrt business logic.

We have to check the Internal functionality wrt external interface. We need to test front end engineering and backend engineering and business logic

**Maintenance:**

Change request/Request for change (RFC)/:

* Post deployment if any issues occur and any modification is needed CR is raised.
* During test execution or Post deployment if any modification it is called as a change request.

Disaster Recovery:

1D 2D 3D

Lecture 9 –13/01/21

Organization Structure:

Development Team

Environment Team

BA Build

Testing

SIT UAT

**Organisational Structure**

MD

CFO

AM: Account Manager

DM: Delivery Manager

SPM: Senior project Manager

PM: Project Manager

Project Lead

Team Lead

Test Engg

**Testing Flow:**

Project

Logical data

BRS SRS Design Coding Unit Testing SIT UAT Regression testing

HLD LLD

Pre-production server Pre-release Test Production

Physical data

PP

**Unit Testing:** done by developer

**SIT/CIT: System/Customer integration testing**

**UAT/CAT: User/Customer acceptance test**

**Regression Testing:**

**Preproduction server:** Physical data is tested

Creates environment like production environment

**Pre-release/Confidence/Post-mortem Testing:**

**Production server:**

SIT member doesn’t do UAT testing. Since we want some other member to test using a different approach.

Lecture 11 –15/01/21

**Environment:**

**DEv -http://amazon/dev/8080**

**SIT- http://amazon/sit/8081**

**UAT- http://amazon/uat/8082**

**Preproduction –** [**http://amazon/live copy/8082**](http://amazon/live%20copy/8082)

Replica of production

**Production – http://www.amazon.com/8082**

**Cloud Environment-**

**Red Box Testing: Protocol Testing**

A black box tester who deals with telecom network.

**Gray Box Testing: Hybrid Testing**

A black box tester who has knowledge of development (Programming Language).

**Entry Criteria:**

Make sure entry criteria of Test case Unit is exit criteria of Requirement Analysis

Make sure entry criteria of SIT is exit criteria of Execution

Make sure entry criteria of Execution is exit criteria of UAT test case

**Exit Criteria:**

Exit criteria of design is entry criteria of coding

Exit criteria of coding is entry criteria of Unit test

Exit criteria of SIT is entry criteria of UAT.

Requirement Analysis🡪Test case SIT🡪Execution🡪UAT test case

Testing done by: QA, Testing

Defects: 80 30 10

**Defect Leakage** (**New Defect**)

Let us Assume are doing test on SIT and we have identified 80 defects and UAT identified 30

Now the question arises We need to analyse if these defects are new or existing, If these defects are new then we consider it as defect leakage of SIT.

What is **Gap Analysis**? (**Existing Defect**)

If 10 defects are new that means they are defect leakage of SIT and 20 are existing defects.

To understand which team is lagging to understand the requirement. Coz that results in existing defects.

**Existing Defect: If the defect already came in previous stage and how does it repeat later on in next stage?**

If data is not saved properly after doing modifications after 1st defect

* Same object/data can be used in multiple components/modules in that case existing defect can occur
* Or due to wrong check in check out analysis by developer

**Check in:(ctrl c)**

**Check out:**

**Review during Analysis(SRS)](CCART)**

After completion of SRS design the BA people are responsible to perform review on SRS and in review, they are responsible to check the correctness of the document.

They will check:

1. Are they complete?

Cust can filter past data till 10 years

Where systemdate-4 years

1. Are the meeting client Requirement

Invoice had 4 requirements

All should define

Invoice--- Printable, in PDF

1. Are they achievable?

W.r.t technology that should be implementable

1. Are they reasonable?

w.r.t budget and cost it should be feasible

1. Are they testable?

is u capable to test it? 🡪 descope

**Review during Design(CCUFH)**

1. Are they complete?
2. Are the meeting right Requirement
3. Are they Understandable?

If design is not understandable it will be rejected

1. Are they followable?

If logic is difficult it should be simplified by architect

1. Do they handle errors?

Does the design have error handling capacity?

Standby system: if one architecture/module fails the stand by by works

**Review during Coding: white box testing (Unit testing) (CCUFH)**

* After completion of code design the developer checks the completeness and correction of code, they are going to conduct the unit test i.e they use 2D approach.
* The unit test is also called as micro testing/Program/structural

If unit testing certifies everything still here is chance of error that is solved by 3D logic

2D logic:

Price=26

Quantity=2

Total= Price\* Quantity

Print(total)

o/p=52

In 3D logic we can understand that Quantity can’t be negative

Price=26

Quantity=-2

Total= Price\* Quantity

Print(total)

o/p=-52

1. Are they complete?
2. Are the meeting client Requirement
3. Are they Understandable?

If design is not understandable it will be rejected

1. Are they followable?

If logic is difficult it should be simplified by architect

1. Do they handle errors?

Does the design have error handling capacity?

Standby system: if one architecture/module fails the stand by by works

**Process Management:**

How to implement SDLC

If the process is correct the software will work properly

It defines how we would implement SDLC

Types of process Management:

Spiral/ waterfall/ V model/ Agile/PET

**DRE(Defect Removal Effeciency)(Test productivity/test efficiency/ defect efficiency)**

It is a s/w testing metric to measure the percentage of defects detected internally vs externally

Internal defects🡪 all defects from requirement till end of testing

Externally defects🡪 defect during UAT(found by client)

**DRE: Defect removal Efficiency/DD: Defect Deficiency/ Test Productivity/Test Efficiency**

*Defect Removal Efficiency relates to the ability to remove defects introduced to a system by a project during the project life cycle.*

*At its simplest DRE can be expressed as a percentage where*

*DRE = (total defects found during the project/total defects introduced by the project)*

A=defect found by SIT

B= defect found by UAT

80/(80+0)

80/110={if fraction = 0.8 to 0.9 🡪 it’s called good efficiency of SIT/good testing}

a/(a+b) =1🡪 Its called zero bug density/super efficiency

is it possible to achieve zero bug density?

of course, it’s possible but difficult to achieve

A=defect found by UAT

B= defect found by production

80/(80+70)

80/150=0.5🡪🡪 it’s called bad efficiency of UAT

{if fraction = 0.8 to 0.9 🡪 it’s called good efficiency of UAT/good testing}

a/(a+b)

What are your roles and responsibilities in testing.

