Day 1:

------

Software Testing Introduction

What is Testing?

SDLC life cycle

Software Testing Methodologies- White Box, Black Box, GreyBox Testing

Requirement Evaluation

Test link - https://forms.gle/GLBMxg98FBdS7Pib9

What is Testing?

Its a process of evaluating a system.

2 Types:

1. Manual Testing  -> Write and execute the test cases without any help of tool support.

2. Automation Testing -> Write and execute the test Scripts with the help of tool support.

Who does?

Developers will also do testing --> Unit testing

Managers will do -> Exploratory Testing

User will do --> User Acceptance Testing (UAT)

When to do tetsing?

- Build ready

- Once test case document is written and approved.

- Setup the environment (Software , hardware req for doing testing)

- Test Data

- In the initial stage of SDLC.

When to stop?

- When all the test cases are executed and there are no high severity and high priority defect.

- When Management decides to stop.

Diff between testing and de-bugging:

Testing ->  is the process to find the defect without correcting (Testers)

De-bugging -> is the process to find the defect and correct the code (Developers)

SDLC life cycle:

Software Development LifeCycle

=> Structured process , 6 phases.

=> To deliver a high quality product, with low cost and in the shortest possible duration of time.

1. Requirement Analysis and Planning

2. Design

3. Development

4. Testing

5. Deployment

6. Maintenance

1. Requirement Analysis and Planning

BA -> Business Analyst -> collects all the details from the client and prepare a document

BRS (Business Requirement Specification)

BA will conduct meeting with Team Leads and discuss about the req.

If the Tech leads are ok, then FRS (Functional Requirement Specification)

1)FRS -> It contains all the functionality details. eg: Login functionality, Sign up

2)SRS (Software Requirement Specification) : Eg: What needs to be present in Login functionality

i.e: field specification

2.Design:

=> Done by designers(UI/UX team)

=> During this phase 2 documents will be prepared:

1) HLD : High level design document

2) LLD : Low level design document

Development:

-> Development team writes the code and the once the build is ready they will give it to the testing team

Testing: -> We follow a lifecycle called : STLC (Software Testing Lifecycle)

Deployment :

DevOps team -> move the application from local system.

Maintenance Phase :

Support Team will take care.

Software Testing Methodologies-White Box, Black Box, GreyBox Testing

-------------------------------------------------------------------

Manual Testing :

1. White Box Testing

2. Black Box Testing

3. Grey Box Testing

1. White Box Testing:  -> Structural based testing

-> Done by developers

-> Unit Testing

2. Black Box Testing -> Specification Based Testing

-> Tester

-> System Testing

1. Functional Testing -> Based on the input given whether the app is working fine or not

2. Non- Functional Testing -> Testing the speed and performance of the app. Done by performance engg., using performance tool e.g: JMeter.

1. Performance Testing:

-> It is done by giving the workload.

-> 2 types:

Load Testing  -> Giving the load within the limit

Stress Testing -> giving the load beyond the limit.

Scalability Testing -> increase the no.of user and checking it's performance

Stability Testing -> Testing whether app is stable at all situation.

2. Usability Testing: GUI testing

-> Testing look and feel of the application i.e: User interface

3. Compatibility Testing:

-> Testing whether the app is working on all the devices, websites, OS etc.,

4. Accessability Testing:

-> Checking whether the app is acessible by everyone including differently abled.

Functional Testing:

1. Smoke Testing:

Testing the basic or critical functionality is working fine or not.

2. Re-testing:

Testing the failed test case again.

Once the developer fixed the bug, the tester has to test again.

3. Regression Testing:

Done after there is an enhancement made to the build.

Testing the addressed scope along with it's related functionality.

E.g: Adding emailid to login page.

4. Sanity Testing:

Done after there is an enhancement made to the build.

Testing only the enhanced part.

Sanity is a subset of regression -> Testing will happen when there is an enhancement.

5. Exploratory Testing:

Testing done by experienced person.

6. Adhoc Testing:

-> The person who do this have coding and testing knowledge.

-> They will randomly pick the functionality and do the testing.

-> It's like cross verification.

7. Monkey Testing:

-> They will randomly pick the functionality and do the testing.

-> They don't have any coding or testing knowledge.

3. Grey Box Testing -> Combination of black box and white box.

Requirement Evaluation:

-> It's the process of checking that requirement defined for testing, define the system that the customer wants.

Day 2:

------

SDLC phases-Requirements Phase,Analysis Phase,Design phase,Coding Phase,Testing phase,Delivery and Maintenance Phase

SDLC Models-Waterfall Model.,V Model,Agile Model,Spiral Model.

Test Case Design Techniques-

Levels of Testing

Test Link: https://forms.gle/zHXJujwGbiRPBLaM7

SDLC Models:

Waterfall Model:

-> Linear Sequential Model

-> Without completing the previous phase, next phase cannot be started.

Where?

-> Requirements are stable

-> Small project.

Drawbacks:

-> If the requirement is changed in between we have to start from the first.

Advantages:

-> Each phase is clearly defined.

-> Each phase is completed one at a time.

-> Cost and time is calculated before the start.

Disadvantage:

-> Client interaction will not be available throughout.

-> Adapting changes to the requirement is not possible.

-> Intergration testing is done at the end.

V-Model:  Verification and Validation

Verification -> Static Testing --> We don't execute anything just check

-> Are you building the product right?

Validation -> Dynamic Testing --> We will execute the steps.

-> Are you building the right product?

Where to use?

-> Development and testing happens parallely.

-> Requirements can change but not too frequently.

-> High sucess rate compared to waterfall model.

Spiral Model:

It is a combination of waterfall model and iterative model.

Each phase starts with customer communication and ends with customer evaluation.

Software is developed in a series of incremental releases.

Phase 1: Communication

Phase 2: Planning - project estimation and risk analysis

Phase 3: Modeling -> Design

Phase 4: Construct -> code and test

Phase 5: Deployment -> Delivery , support and feedback.

First Spiral -> Concept development

2nd Spiral -> System development

3rd Spiral -> System Enhancement

4th Spiral -> System maintenance.

When?

-> Large project

-> When release are required to be frequent.

-> When risk and cost evaluation is important

-> When chnages may require at any time.

Agile Model:

-> Incremental and Iterative model

-> When requirements are not stable we can go for agile.

150 modules -> 15 each \* 10 cycle ->

Test Case Design Techniques / Black Box testing technique:

-> We cannot test all possile ways, so this techniques will help us to obtain required test data to be tested.

-> Reduce the size of input and obtain best inputs.

E.g: Input field allow only 41 characters --> Not possible to check 41 times, so we can use 2 case, 1st case within the limit(1-41) and 2nd case beyond the limit(>41).

We have 2 major categories:

1. Static Technique -> Line by line review of test case document.

1. Review  ->  Manually without any tool

1. Informal Review  -> Domain expert

2. Walkthrough -> peer review

3. Technical Review -> Technical Lead

4. Inspection -> Formal review

2. Static Analysis -> Done using tool

2. Dynamic Technique: Execute the steps

1. Equivalence Class Partitioning

2. Boundary Value Analysis

3. Decision Table Testing

4. State Transition Tetsing

5. Use Case Testing

Equivalence Class Partitioning:(ECP)

-------------------------------

E.g: FirstName field(mandatory) -> 40 chars

rage: 1 - 40  --> we can partition the range of values into equal class

1 - 10   11 - 20   21 - 30   31 - 40  -> 4 class (Valid input condition)  --> positive testing

8 -> asdfgyhd

15 -> asdfghuyrhegtds

26 ->

37 ->

Below 1   Above 40   --> 2 class (Invalid input)  --> negative testing

E.g: password -

Range: 1 - 10 chars

1. 1 - 5    6 - 10   --> 2 class (positive data)

2. Below 1   Above 10  --> 2 class (-ve data)

1. 1 to 2   3 to 4    5 to 6   7 to 8    9 to 10 --> 5 class

Boundary Value Analysis: (BVA)

-----------------------

E.g: 1 - 43

A = 1    B=43

A-1,A,A+1,B-1,B,B+1     --> 0 1 2 42 43 44

0 and 44 --> invalid i/p

1 2 42 43 --> valid i/p

Decision Table tetsing:

------------------------

Formula : 2^n  ---> n = no.of test condtions (i/p fields)

E.g: To vote for election we have 2 conditions

1. He/She should be an Indian citizen 2. He/She should be above the age of 18

2^n -: 2^2 --> 4 posible inputs

Inputs                       O/P

Citizen   age > 18

True   True   True

False      True                   False

True   False                  False

False      False   False

State Transition Testing:

------------------------

According the user access level the state of the application should change accordingly.

E.g: Basic user, Permium user, Admin user --> all these three users should be able to access application at their access level only.

Use Case Testing:

-----------------

Interaction between system user and the function supported by the system.

Levels of Testing

------------------

1. Unit Testing: Done by Developer

-> Tetsing the induviual part is called unit testing.

-> This comes under White Box Testing.

-> Code testing will happen.

2. Integration Testing:  Integrate --> combine or consolidate

-> Combining the logically related modules together and performing the testing.

-> To check the data flow/ connectivity.

-> To check whether our app is moving from one page to another/one app to another.

-> Done either by developers or tester

-> This comes under grey box testing.

1. Top-Down Approach --> If any of the low level module is not available then we use dummy module called stub.

2. Bottom-Up Approach --> If any of the high level modul is not available then we use dummy module

called driver.

3. System Testing: End - to - end testing

-> Done by tester

-> This comes under Black Box testing.

4. Acceptance Testing:

1. Internal Acceptance  --> Alpha tetsing (At developer site)

2. External Acceptance  --> UAT (Client side) (Beta testing)

Day-3:

-----

Software Testing Life Cycle

QA & QC & Testing

Bug Life Cycle

Test Link: https://forms.gle/HBSxVR54BQ8jekHZ7

Software Testing Life Cycle  - STLC

Sequence of different activities performed only by testers.

-> Requirement Analysis

-> Test Planning

-> Test Case Development

-> Test Environment setup

-> Test Execution

-> Test Closure.

Requirement Analysis:

---------------------

If the testing team has any doubts, they conduct meeting with BA, Dev team clarify the doubts and prepare the 2 documents:

1. Automation Feasibility Report --> Prepared by Test Lead

It will have some set of questions and

-> If the score is above 70 -> go for automation

-> If the score is below 50 -> go for manual

-> If the score is between 50 and 70 -> submit proof of concept.

2. Requirement Traceability Matrix(RTM): Entire team will be involved.

They map the business requirement with test cases.

Each business requirement will have it's corresponding id. Every test case will have it's id. We have to mark this in excel. So by referring to the document, we will get to know the status of the test case execution.

JIRA - Project Management Tool.

E.g: BD01(Login functionality)  --> 4 Test case TC\_001,TC\_002,TC\_003,TC\_004

Test Planning:

--------------

Only the Lead or Manager will involve and they will prepare 2 documents:

1. Strategy document: This will be same for all the projects inside an organization.

2. Test Plan:

What Tools?

Which Technology?

Do we have proper resource ? Do we need to train?

Resource planning, roles and responsibility.

Test Case Development: Test will be involved.

----------------------

-> In case of automation we will write the test scripts.

-> In case of manual we will write the test cases.

2 Documents:

1. Test Scenario: What needs to be tested?

Functionality that needs to be tested.

E.g: Login Functionality, SignUp Functionality, Add\_To\_Cart Functionality, Checkout Functionality, Payment Functionality etc.,

TS\_AM\_001  Login Functionality

TS\_AM\_002  SignUp Functionality

2. Test Case Document: How to be tested?

This will have the steps to execute.

TC\_AM\_001

Environment Setup:

-> Setup Test Bed  ->  Configuring the Hardware and software needed for testing.

-> Build Verification Testing(Smoke Testing): Testing the Critical Functionality.

Test Execution:

-> Execution of test cases

-> Reporting the test result

-> Logging defects for failed test case.

-> Verification and re-testing of the defect.

-> Closure of defect.

Test Closure:

If there is no high priority bug, test will be closed.

-> Test Closure report / Test Summary Report

-> How many test case written

-> How many passed, how many failed, how many defects were reported etc.,

QA & QC & Testing

Quality Assurance:

Ensuring a company is providing a top notch quality product.

Ensuring the product will fulfill the quality

Part of SDLC

Whole team is responsible for this.

Preventive approach.

Quality Control:

-> Part of STLC

-> QC is a subset of QA.

-> Finding the bugs and getting it fixed.

-> Reactive approach

Testing:

-> Find the bugs.

-> Testing is a subset of QC

-> Only Tester will involve.

Bug Life Cycle:

Error found by the tester is called Defect. The same error accepted by the developer is called Bug.

1) New => When the bug is identified.

2) Assigned => When it's assigned to a developer.

3) Open => When the developer starts to work on the reported bug

4) Pending retest => If the testers are occupied with other work then it will be pending.

5) Retest => Tester will retest the changed code.

6) Reopened => If the bug is still present even after developers change.

7) Verified => Verify whether all the reported bugs are fixed.

8) Closed => Once all the bugs are fixed, we will close it.

Developers can ignore/reject a defect with following reasons:

1. Duplicate

2. Not a bug

3. Deferred - Postponing the bug to next release based on the priority if there is no time.

Bug leakage: Product is released, but the defect is found by the end user, missed by tester.

Bug Release: Releasing the product with the known low priority defect.

Day 4:

------

Recap of testing techniques

Agile methodology

Test Link: https://forms.gle/mR88tWbsyc6xHFA39

Agile methodology

-----------------

Agile is an incremental and iterative.

Big projects will be divided to smaller chunks/units/iteration that is called as sprint.

Iterative Development: Software is developed and delivered to the customer and based on the feedback, again developed in cycles or release or sprints.

E.g: 150 modules  --> 15 modules for each sprint. --> 15 \* 10 = 150 -> 1 sprint (2weeks)

Example: Release 1 software is developed in 5 sprints and delivered to the customer, now the customer is having change the development team plan for 2nd release which can be completed in some sprints.

Incremental Development: Software is developed in parts or increments.

E.g: In each increment, a portion of the complete requirement will be delivered.

Advantages of Agile:

-> If the req's are not stable or if its keep on changing then we can go for agile.

-> Customer interaction will be available through out the project.

-> It is used for big project.

To implement the agile methodology we do have many frameworks:

a) Scrum b) Kanban c) Extreme Programming

Framework: It is a set of rules or protocols to implement anything.

Scrum: It's a framework which is used to implement an agile methodology.

Agile core values:

1. Individuals and interactions over processes and tools;

2. working software over comprehensive documentation;

3. customer collaboration over contract negotiation and

4. responding to change over following a plan.

Scrum:

------

Sprint --> Its a continuous time-boxed life cycle.

The duration of sprint: It should not go beyond 4 weeks.

User stories -> In agile the requirements called as user stories.

E.g: In gmail, in the login functionality, each field UN, PWD, Login button --> these are called as user story.

Collection of user stories --> called as Epic

Ex: UN, PWD, Login button  --> the entire login functionality is called as Epic.

Product Owner --> who collects the user stories from the client

Scrum Roles:

1. Scrum Team

2. Product Owner

3. Scrum Master

1.Scrum Team

-> It's a self organized team (There is no bossy culture)

-> It's a cross functional team (All members (designer, dev, tester, support etc.,) all work together.

- Team size is 6 + or - 3

2. Product Owner:

-> who collects the user stories from the client

-> Who acts as a bridge b/w customer and the team

-> Once the user story is collected , PO will create Product backlog.

-> Who prioritize/re-prioritize the user story from backlog.

-> Who will decide whether the sprint should gets release or not.

3. Scrum Master:(3'P)

a)Process coach

b)Protector

c)Problem solver.

Scrum Meetings:

1.Sprint grooming meeting

2.Sprint planning meeting

3.Daily scrum meeting

4.Sprint review meeting

5.Sprint retrospective meeting

1. Sprint grooming meeting:

Grooming is an open discussion b/w scrum team and PO.

The user stories are discussed to help the team to gain better understanding about the functionality.

Have to give story/user points (measure to indicate the work to be done(hrs,days)

1 user point - 4hr

Username - user story

des --> 12 hrs ---> 3 story points

dev --> 12 hrs ---> 3 story points

tes --> 16 hrs ---> 4 story points.

2. Sprint Planning meeting:

PO,SM,Team --> plan for the sprint

Like what are all the user stories they have to complete for the particular sprint, sprint duration, roles and responsibility for each team member.

3. Daily scrum(stand-up) meeting:

15mins

1. How much work has been completed

2. What's plan for the day.

3. Mention technical issues.

No discussion only update.

4. Sprint Review Meeting:

-> Before the release of the sprint, team will give demo to the PO, Client, SM.

-> If they are ok, PO decides whether the sprint should gets released or not.

-> PO may give some modifications --> will gets added in the backlog

5. Sprint Retrospective meeting:

-> They will discuss the lessons learnt(positive or negative) from the sprint

-> This will help us for upcoming sprint.

Sprint Backlog: Sprint planning is done, but the user story work is yet to be started.

Spill Over: If any user story is not yet completed in the particular sprint, we can able to move on to the next upcoming sprint.

Day 5

Excel – Test case

Day - 6:

--------

Project management tool

Hands-on with Jira & Scrum:

Creating epic, user stories, Sprint planning,Backlog ,Seviority ,Priority and Release Version.

JIRA: Project Management Tool

It is used for bug tracking, issue tracking, planning and maintaining the project.

This helps you manage your project effectively and smoothly.

We can manage:

Planning

Tracking

Releasing

Reporting

What is Epic?

Large use story that cannot be delivered with single iteration.

What is User story?

Smallest unit of work.

Eg:

As a [person], I [want to], [so that I can achieve this]

As a user, I want to be able to add items to my shopping cart.

As a customer, I want to perform login functionality with valid credentials.

-------------------------------------------------------------------------------

1. Create account in GitHub -> https://github.com/signup?source=login

2. Download Git in your system -> https://git-scm.com/download/win

Execute and install the app to your system