Stage 1 – Testings

Stage 2 – Python Programming

Stage 3 – Automation – Robot Framework (Web automation/windows automation), Database

Stage 4 – Customized Framework

Stage 5 – CI/CD – Git & Jenkins

* Assessment for each stages.
  + Assessment – Feb 16
  + Assessment – Feb 21
  + Assessment – Feb 23/24
  + Assessment – Feb 28
  + Assessment – Mar 2
  + Assessment – Mar 4
* Final Caption Project

1. Software Testing
   1. It is the process of executing of programs with the intent of finding the errors as early as possible.
   2. Helps verifyting and validating the software as per the requirement
   3. Aims at
      1. Finding defects
      2. Demonstrating the lack of quality
      3. Gap in requirement and product
      4. Faith in the end product
2. Role played in the company

Trainee 🡪 System engineer 🡪 Senior System engineer 🡪 Technology Analyst/Business Analyst 🡪 Team leader/Technology lead/Test lead 🡪 Project Manager 🡪 Senior Project Manager 🡪 Delivery Manger

1. How product is developed in the market?
2. Software development model – Waterfall model, V-Model, Agile Scrum model
3. SDLC – Software Development Life Cycle
   1. Requirement Analysis
      1. Gathering & Analysis
      2. Documentation
      3. System input and System functions
   2. Design
      1. Design Specfication
      2. Technology
      3. Broken into small modules
      4. Funtional planning
   3. Development Phase
      1. Design Specfication is transformed as working model
      2. Develop component and intergrated
      3. Unit Testing – developers
   4. Testing
      1. Small component will be tested first & then major components
      2. Test Execution based on plan
      3. Report the defects to the team and fixes
      4. Testing – Sign off
   5. Maintenance
      1. Troubleshooting
      2. Support – help desk & training
      3. Latest updates software
4. Waterfall model

One should move to next phone after completing the preceding phase only

Advantages:-

* Simple
* Provides a structured way of doing things
* Models works when requirement is crystal clear.

Disadvantages:-

* Communication gap can result in disaster
* Poor resource allocation
* Need crystal clear requirement

1. V-Model
2. Agile Scrum model – iterative model

Vision is to develop application similar to teams for internal usage.

Product backlogs – login, signup, start meeting page, voice call, video call option, cancel meeting, add to calendar option, mail feature.

(assume we have – 100 product backlogs) – need to complete 1 year – 52 weeks

Scrum master – role is to understand the product backlogs, explain to everyone in the team, allocate backlogs to develop, controlling the sprint, sprint backlogs

100 product backlogs – 52 weeks

Total Sprints – 26 Sprints (considering each sprint as 2 weeks)

Sprint grooming – where review happens now a days. – before each sprint

Sprint 1 (8 sprint backlogs – decided before entering the sprint) – 2 weeks 🡪 some backlogs should be developed and tested (manual testing).

Sprint 2 (10 sprint backlogs) - 2 weeks 🡪 some backlogs should be developed and tested (progession testing) (manual testing), automation for sprint 1 will happen

verifying the previous sprint feature is also working fine or not (regression testing for all sprint 1).

Sprint 3 (10 sprint backlogs) - 2 weeks 🡪 some backlogs should be developed and tested (progession testing) (manual testing), automation for sprint 2 will happen

verifying the previous sprint feature is also working fine or not (regression testing for all sprint 2 & for sprint 1 check – automation will be triggered ).

....

Sprint 26 (4 Sprint backlogs) – 2 weeks 🡪 some backlogs should be developed and tested (progession testing) (manual testing), automation for sprint 25 will happen

verifying the previous sprint feature is also working fine or not (regression testing for 25th sprint manually and automation will be triggered for 24 sprint ).

1. Importance of testing:
   1. Loss of money & testing
   2. Customer Satification
   3. Stay in business
   4. Injury or leads to death
   5. Reliablity of the software
2. Necessity of testing
   1. Discover & prevent the defects
   2. Quality control assurance
   3. Gain customer confidence
   4. Early testing & defect removal is less expensive

Facts:

Ariane 5 explosion – (16 bit digits instead of 64 bit digits – june 4, 1996)

Microsoft’s anti-Unix site crashes – the sites were running open source unix and apache servers. So they tried to host their own server (IIS server)

1. Error – Failure – Defect

Error (mistake) – action perform by human (human mistakes)

Failure – inablity of the system to perform functionality according to its requirment.

Defect – mismatch of the actual and expected result.

Examples: adding two numbers

{

a= 50

b=20

c = a – b

print (c)

}

Output – 70 ( but output is 30)

1. Seven testing principle
   1. Testing shows presence of defects - Testing reduces proability of undiscovered defects remaining in the software
   2. Exhaustive testing is impossible – Testing everything (all combinations of input and precondition ) is not feasible
   3. Early testing – focused defined objective.
   4. Defect Clusting – small number of modules contains most of the defects discovered during pre-release or show most of the failure
   5. Pesticide paradox
   6. Testing Context – testing depends on the context. Act differenlty on different contexts
   7. Absence of error fallacy – If your software or system is unusable then it does not matter how many defects are found and fixed – it is still unusable.
2. Fundamental test process - Software Testing Life Cycle
   1. Test Planning
      1. Testing objective for the requirement
      2. Test Plan – Documentation
   2. Test Design
      1. Test case Designing
      2. Test Indentifying the test data
      3. Environment setup - Tools required
   3. Test Implementation
      1. Test Execution
      2. Test Result
      3. Log the bugs found
      4. Re-test
   4. Test Closure
      1. Environment Cleanup
      2. Process & reviewed in the meeting
      3. Lesson learnt – for upcoming cycles
      4. Archive
3. STLC Deliverables – Entry & Exit Criteria
4. Test Planning
   1. Requirement Traceablity Matrix (RTM)
   2. Test planning/Stratergy document
      1. Automation feasability
      2. Effort estimation
5. Test Design
   1. Test Scenario connected with proper requirement
   2. Test Case
   3. Test Data & Script
6. Test Execution
   1. Test result
   2. Bug report – Bug/Defect - priority & Severity
   3. Submitting the final executable report
7. Test Closure
   1. Test Summary Report
   2. Test Metrics
      1. Test coverage
      2. Defect priority & Severity
   3. Lesson learnt – for upcoming cycles
   4. Sign off to release the product

Requirement <-Scenario <-Test Cases

1. Responsibilty of QA engineer:
2. Analyse
3. Planning
4. Scenario
5. Test case
6. Test script & Data
7. Test execution – pass or fail
8. Bug/Defect tracking –
   1. Finding Bug/Defect & tracking it
   2. Retest after fixing
9. Status report
10. Testing levels
    1. Unit Testing
    2. Component Testing
    3. Integration Testing

Integration Testing Approach

* + 1. Big bang approch

Wait for the all part to get intergrate and then testing how well the parts are interacting.

When some module is not integrated then we use below concept to test

* + 1. Top down approch

Login ( create a dummy model)

Transfer

Withdrawal

FD

* + 1. Bottom up approch
       1. Login
       2. Transfer
       3. Withdrawal
       4. FD (create dummy model)

* 1. System Testing
     1. Concentrating the behavior of the entire system
     2. Functional and non-functional testing
  2. Acceptance Testing – end user/customer
     1. Operations Acceptance testing – performance, stress, volume.
     2. Compliance acceptance testing – safe at work, rules & regulation
     3. Alpha/Beta testing – refer slide

BDD/ATDD – majorly used for achieving the acceptance testing (popular way of acceptance testing)

1. Testing Types
   1. Functional Testing

Black box testing

* + 1. Requirement specific testing
    2. Business process based testing
  1. Non-Functional Testing
     1. Performance – reponse the application
     2. Stress – bug fixing at high load.
     3. load testing – loading the many user
  2. Structural Testing
     1. White box testing – can be done at development
  3. Testing for changes
     1. Confirmation Testing or retesting
        1. Making sure the defect are fixed or not
        2. If it is working fine, then we cannot conlude the sytem is fine.
     2. Regeression Testing
        1. Test carried out to check the modification done has not brought any new defects.
        2. To be executed whenever code changes.
        3. Suitable level for implemention of automation.
  4. Maintenance Testing
* Done on the application already in production and want to change the old system.
* Enhance the application to improve the response.
* Migration of application from one database to another database
* Migration of application from one platform to another
* All levels of testing like component, intergration, system, acceptance testing will happens here.

Test plan 🡪 anlysis and design 🡪 implementation 🡪 exit criteria

1. Test case design:- working on orange hrm application
   1. Login module / homepage
   2. Forgot your password module
2. Testing techniques
   1. Static testing – undergone without running the software code
      1. Walkthrough
      2. Inpection
      3. Review
   2. Dynamic testing – undergone with running software code
      1. Black box tesing
      2. White box testing
      3. Grey box testing
      4. Experience based testing
3. Review & Test Process:-

Informal review – no proper protocal to run it

Formal review:-

Roles

Moderator – leads the review.

Scribe – someone logs the defect raised in the meeting.

Author – unclear documentation / validating the defects.

Reviewers – review the material for defects. (your role)

Manager – make sure the review objective is met.

1. Black box testing/ Requirement specification based testing
   1. Equivalent partitioning testing

OTP – 4 digit

|invalid|valid |invalid

| <4 | =4 | >4

101 – invalid

1203 – valid

12345 – invalid

* 1. Boundary value analysis

OTP – 4 digit

Invalid|valid|invalid

999| 0000/1111 to 9999 | 10000

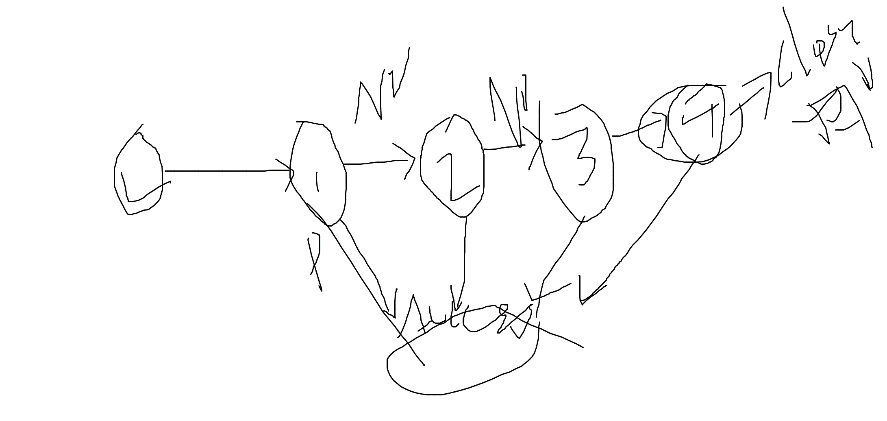
Testing the valid records – 0000/1111 and 9999

Invalid – 999 and 10000

* 1. Decision making

|  |  |  |
| --- | --- | --- |
| From | To | Search (enabled or disabled) |
| T | T | T |
|  |  |  |
| Any one for testing | | |
| F | T | F |
| T | F | F |
| F | F | F |

* 1. State transition



1. Expirence based testing - no format will be followed
   1. Exploratory testing – based on expirence testing will be conducted
   2. Ad-hoc testing – Situtation based testing
2. Structure based testing – by the developer. (we use some testing framework)

Selenium – Only Web automation

* 1. Selenium IDE
     1. No Programming knowledge required.
     2. Record and playback feature
     3. Plugin available – chrome, edge, firefox
     4. Use for simple scripting & exploratory testing
  2. Selenium RC
  3. Selenium WebDriver
  4. Selenium Grid

UpperCamelCase – MyFirstProject

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

Git – Source Code Management – Open Source

Architecture of Git:

Local folder (local machine) 🡪 Local repository (in your local system) 🡪 Remote repository (github, bit bucket, AWS Code commit)

Installation:

Setup Git in your machine –

<https://github.com/git-for-windows/git/releases/download/v2.35.1.windows.2/Git-2.35.1.2-64-bit.exe>

Working with Git:-

Working on Local repo:-

* + - 1. Create a local repository – **git init**
      2. Files planning to move 🡪 **git add .**
      3. Update the local repository 🡪 **git commit** –m “day1 notes”

Working on Remote repo:

* + - 1. To register the github url using variable “origin”🡪 **git remote add origin** [**https://github.com/balaji-githubstore/NCSNotes.git**](https://github.com/balaji-githubstore/NCSNotes.git)
      2. To update remote repo 🡪 **git push -u origin master**





Notes:

1 GB – 1024 MB

1 MB – 1024 KB

1 KB – 1024 B

1 B – 8 bits

1 bit – 0 or 1

//hello