http://www.gcreddy.com/2014/07/software-development-life-cycle.html

SDLC : Software Development Life Cycle…set of phases that will continue in a cycle to complete the process.

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

SDLC model can be classified as two types :

1. Sequential models
   1. Waterfall model
   2. V-model
2. Incremental Models
   1. Prototype model
   2. Spiral model
   3. Agile model

A typical Software Development life cycle consists of the following stages:

1. Requirement Gathering /Planning – business manager / project manager or owner

2. Analysis – product owner / BA

3. Design – architect / ui designer

4. Implementation/Develop/coding – developers—frontEnd/backend/application developers and DBA team

5. Testing – test engineers and team

6. Maintenance --

People involved in projects : in diff company designations would be different.

Project cycle : DEV-🡪SIT🡪UAT🡪Deploy

Deployment : building bundled together with all the jars and moving into server (Application package into server machine: build Engineer/release Engineer/deployment Engineer)

Stake Folder : a person/group/Organisation who is effected or who those influence directly or indirectly

Use case diagrams: Uniform defines about actors and their responsibilities or functionality according to actors.

Activity Diagrams: like flow charts. Every steps are broken down into flow depending upon the execution continues.

Sequence diagrams: depicted in form of a sequence diagram…step by step sequence.

Class diagrams: defining variables and methods/properties and action for given behavior. Are very technical.

Wireframes: pictorial representation of the requirements, position of the elements. For mobile testing we get the requirements documents are called wireframes.

**Testing Fundamentals**

**Testing:** is the process to assure the quality of the product during which we find defects.

Before testing we make a test strategy, test plan, write test case ,implements test scripts and generate reports. This whole process is called as Testing.

**Why testing:** to deliver high quality product, to find the defects at the early stages. If testing was not done and the product is released into market and there are any defects found then it effects the business lot and costlier to fix the defects at the later stages. So finding the defects at the early stages is the best way which saves the cost and the outcome would be the high quality product.

**Verification and Validation:**

**Verification** is ensuring as we are building the product right as per the requirement, and **Validation** is did we built the right product.

Verification is done initially mostly by **developers**(and also testers) at early stages, and Validation is done at the later stages done by the **QA** people. Validation is doing beyond the verification from end users perspective.

Verification and Validation both are done after developing the product. Validating is like testing in performance based.

Ex. If we want to buy a printer we check the configurations insert the paper and check the printing, which is verification. Again we check how much papers are able to print, how much time it is taking how much power consumption is happening all these are come in Validation.

**Exaustive testing is not possible**: testing with all the test data..we try check with the all the possible test data. try to think of efficient test cases.

**Principles:**

1. testers should pursue defects not the people
2. We should not assume no defects exist
3. One defect may raise more defects
4. Test for valid expected and invalid unexpected
5. Testing to be done in very intellectual way

**Test Case:** a planned sequence of steps which we follow during the testing in order to find defects, which consists of test id, test description, test data, expected, actual, result--- and we write actual and result after testing.

**Other terminologies:**

**Test Suite:** A set of individual test cases that are executed as a test package, in a particular sequence.

**Test Cycle:** consists of series of test suite that keep executing for the given module, from the initial setup till the generation of reports and clenup.

**Test basis:** base documents /user story/picture which ever using for testing

**Test Data**: input data

**Test Execution**: the process of running a test: how we are executing: automating scripts or manual test cases

**Test Bed:** test Environment: contains hardware, instrumentation, simulators, software tools, and other supporting elements needs for testing. Making the test env. Ready.

**Testing Life Cycle:**

Test cases--🡪 Execution of testcases--🡪 results

RequirementReview-🡪Testplanning-🡪TestDesigning-🡪EnvironmentSetup-🡪TestExecution-🡪Reports

Diff Environments: Dev Sit UAT Prod

Dev QA Stage Prod/Live

Dev QA1 QA2 Prod

**Before Rising Defect, testers need to do few things:**

1. Debug the cause of the issue: sometimes the issue could be at deployment level: then we ask the deployment team to check and ask them to install the build and also try deploying the previous build to check the problem is with the current build or with every build.
2. Issue would be at DB level(dev,QA,Live environment—data issue): need to contact the DB team or whoever is maintain the database sometimes Sr.developer also take care of the DB.
3. Issue could be at API level: we check the developer tool(f12) and check the console, network for the network related issue or other.--> inspect the browser, check the server logs, application logs( by connecting to the Unix)
4. Issue at the developer level—Raise the defect.

**Levels in Software Testing:**

1. Unit Testing
2. Integration Testing
3. System Integration Testing
4. Acceptance Testing

Unit testing: testing the smallest individual unit of the code by writing another piece of code is called Unit Testing. This is also called whitebox testing because we know internal details of the code (code is visible).

Generally in order to write unit test cases, there are unit testing frameworks which are language specific. Ex. For Java—jUnit, testing for Dotnet—nunit.

TDD: Test Driven Development: test cases or test scripts will drive the development. While developing the application the developer writes the test scripts and try to check the results , and make some changes and again do testing, so based on test scripts code is modified and its driving the development, hence it is called as TDD. With this early stages the defects are identified and fixed. Unit testing is done in DEV Environment.

**Integration Testing:** individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. It is done in Dev Environment. Done by both developer and tester.

Big Bang Approach

Top-down

Botton-Up

Component testing

sit

**System Testing:** testing process where complete integrated system is tested in testing environment. This is done by the testers.

**SIT:** testing the system end to end with respect to UI and DB. Ex. User screen, integration between the connections, outcomes of the systems everything according to the requirements in whole system.

Sit is done after System testing.

SIT:FURPS: Functionality,Usability,Reliability,Performance,Supportability:

**User Acceptance Testing:** testing the application in end users perspective that everything is correct as per the requirements. This is done generally by PO/team who represents the end user.

**Type of Testing:**

1. Smoke testing: Testing the critical functionalities/ high priority things of the system before it is accepted for major testing. Checking priority test cases as per the requirement.(functional)
2. Checking the main functionalities of the system are called functionality testing.
3. Usability Testing: look and feel: size,frint size
4. Performance Testing: once the functionality is all good then we check the performance testing. Speed should not come down when the load is increasing, it you bare the load, this is tested using performance tools. Ex. Jmeter- PT is based on Load testing, Volume testing, Stress testing. (non-functional)
   1. Load: Simultaneous requested from multiple users
   2. Volume: with huge volumes of data
   3. Stress: finding the breaking point of the application. Ex. Present my application is supporting 500 users …I have the breaking point till 1000 users. Planning is done accordingly.
5. Security Testing: making sure authentication and authorization is working fine or not.
   1. Authentication: something like for login page, with the login credentials we are authenticate to the go further. Username/password or touch based finger prints, security questions, captcha, retina scanning, smart cards, opt passwords etc…
   2. Authorization: based on roles and permissions—like admin user diff options for the page, prime member facilities would be differ –Ex.amazon
   3. Cross-site scripting: it’s a way of sending malicious codes/ scripts into to hack the system. It’s vulnerability.
   4. Sequel injection: it’s also one kind of vulnerability
   5. DDOS: distributed denial of system:
6. Regression Testing: Testing the application with new enhancement making sure no features are giving any new defects. Testing the fixed defect is called Retesting.
7. User Acceptance Testing: this is done just before the system is going live. It’s done from end users perspective, and this is done by the person who represents the client or the team who represents the client. It’s also called as Beta Testing.
   1. **Alpha:** Testing done within their releasing the product within the company Ex.MS release new version of windows within their company to the employees first so that they can take feedback from team to correct any issues.
   2. **Beta:** Release the product to the end users as Beta Version to the end users and take the feedback and issues fixed and the final product will be released.

This UAT environment looks almost similar to live Env.

1. Production Validation Testing: sometimes there are some issues we won’t get in UAT those we get in production. These production issues are the most priority one to be fixed ASAP. Final opportunity to fix the bugs.

**Testing techniques: BlackBox Techniques:**

1. **Equivalence Portioning:** Ex. Registration date—checking valid and invalid input within the range…10 users for booking –min 1 person, valid 1to 10—so we check any 1,5,8any and we check invalid--<1 and >10—0,7,11
2. **Boundary Value Analysis:** checking at the boundary points –checking at 1 and 10 also…
3. **Decision Making graph:**
4. **Use Case testing:**
5. **State Base Testing:** Depending on the state
6. **Error Guessing:** based on error come with more test cases
7. **Exploratory Testing:** without the document, with minimal requirements we come up with our own idea with new techniques.

**White Box techniques:**

1. **Statement Coverage:**
2. **Decision Coverage:**
3. **Memory Leak:**
4. **Debugging:**

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

Quality : Quality does not means same for everyone.

State Holders : Client, Producers and customers…

Something which expands beyong its limit is called Scalability.

Defect : any unexpected behavior

Defects are also referred as : Issue, bug, problem, fault and error.

Ex.Missing requirements , extra requirements if functions are not working…environment problems, documention errors etc.

Diff between FAULT and FAILURE ?

Fault is when a requirement or feature is working incorrectly, where as Failure is when whole AUT goes down, it usually accompanied by restoration activity.(restarting database, restarting the system…recovery operations)

Fault is any defect.

AUT : any kind of Application Under Test

SDLC : V-Model : Extremely useful for testers.. another development life cycle

Validation is the actual testing.

Left handed : verification testing and right handed is the validation testing

Verification …When we are checking if we built the right product.

Validation ….. did we build the product right.

Verification is static testing and validation is Dynamic testing.

Unit Testing : Done mostly by developer, done in development environment, manually or automatedly…

Unit code of code is developed and ready to test-verification is already done…and the output ready to be integrated based on the design

100% Test coverage :

Integaration Testing : input for integration testing is the “units of code are tested and ready to be integrated.”

Rules for Integration testing : two strategies : Bottom-Up and Top-Down

These are the directions to follow the integration process..

Integration testing is white box testing as we can see the code and also called as the GREY box testing.

Ex.from module A to B we see the coding , and module B to C we see the code ,which is white box testing. Once the A, B, C modules are integrated ..we test it with some input and get check for the correct output, in this scenario we don’t need to see any code which is again Black box testing. But at the end if we get the incorrect output then we need to see the code which again is White box testing.Hence it is the shades of mixing white box and black box which is sometimes called as Grey Box testing.

STUBS : two module Calling function and called funtions

A is 100% developed and B is 0% developed ..in order to test integration we create a dummy module as a proxy which has limited functionality and which gives temperory solution. the dummy one is called STUB.

DRIVERS : IF A is 0% developed and B is 100% developed ,then we create a dummy for called function which is called as DRIVER.

Calling function is STUB and called function is DRIVER. The output of the Integration testing is the Entire system put together to test.

SYSTEM TESTING : is a Black Box Testing….done by QA team in testing environment. functional and non functional testing can be done in system testing.

System testing is done in stages..

Questions :

What is software,project, domain,technology and testing,defect

SDLC :Initiate-define-design-build-test-deplyment

Main topics : youtube video

Quality :

What is Testing : **Testing** is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, **testing** is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Waterfall SDLC :

Funtional non functional requirements need

Test Strategy

Agile test strategy, Agile SDLC(iterative approach to sdlc)

Test Scope

Test types

Components testing based on internal architecture of the system.

Static Testing : testing the specifications, not actual system.by checking documents we can avoid defect multiplications. testing quality work products,meaning checking documents not final product ,

Developers use code reviews without code executions, just reviewing code, line by line they can find some errors and they can fix them.

1233780375 / item no. 190322030691 / no. 510 797 3333 / usps tracking number : 92748999920645553013952941 / 9th sep 1:46pm / mailbox

To automate a Rest API:

Toolswhich are used for are soupui, postmanWe need for rest api automation:

Automation can be done based on lang: like for java—httpClient,RestAuusred

For java scripy—chakram.

To use HttpClient to automate first need to creave the maven project and add dependencies….

Dependencies:

1. http framework using apache—to call theapi and to get theresponse
2. ip utils jar: to convert response into proper string because it comes as a entity –apache iputils
3. in order to implement oauth configuaration setup- OAuth signpost and oauth common http –
4. data driven testing –apache poi
5. testing for assertions

add all the dependencies to the maven project – create the class under maven project and create a test class for assertions

testing the twitter api as it has the oauth

1. json org

we use the postman tool first to get the response…first time we need to install the app and launch –postman plugin

using twitter API –for that we need the ligin account(normally we get the credentioals\_--we are using public api’s now

every company has their own api docs(documents)

plugins are browser besed : postman –chrome, restclient-firefox, we can use soupui also for response

What is payload?

How do you set a payload when it comes to request?