SDLC Life Cycle

09 March 2022 12:32

- Sequential Development is the sequence followed with a good format for software testing it has 5 main sequences:
    1. Requirements
    2. Design
    3. Build (where we write code)
    4. Test and fix (testing part)
    5. Release
- SDLC has 2 types:
    1. Sequential
        a. Develops in a linear sequential flow
        b. No overlapping should happen (although in practice it may happen)
        c. It further has two types:
            i. Waterfall model – the development activities are completed one after the other. The sequence is "requirement – design - implementation – test – maintenance".
            ii. V model – Modified waterfall model. The test process and development model go side by side. It focus more on the testing part (waterfall had only one testing part). The 4 testing done here are "acceptance, system, integration, and unit".
    2. Incremental (or iterative)
        a. We develop our software in parts
        b. One part goes through one complete life cycle and then only the next part is taken forth.
- SCRUM
    - Scrum is a combination of sequential and Iterative model. It divides the software into short iteration called sprints.
    - We take backlogs in each session and in each iteration we develop increment in that system.

Software Testing

09 March 2022 12:33

- ST is a way to assess the quality of software and reduce risk of software failure.
- ST != test execution
- It is a process that has many activities.
- ST is divide into 2 types:
  - Dynamic – This means that the software is executed. Like opening a website, typing something into the app etc.
  - Static – Nothing is executed, only review is done here.
- ST is divided again divided based on performance:
  - Validation – User is happy with product
  - Verification – product is built In right way.
- Objectives of ST:
  - Work evaluation
  - Requirement fulfillment
  - Building confidence
  - Finding defects
  - Preventing defects (testing is done here)
  - Info to stakeholders
  - Risk assessment
  - Law compliance
  - Final objectives
- Testing and debugging has 3 stages:
  - Testing
  - Debugging
  - Confirmation testing

Test Process

09 March 2022 12:33

- There is no universal test process but any test process consists of 3 main steps:
    1. Plan (analysis – strategy - tools)
    2. Design (test cases – scripts – scenarios – environment)
    3. Execution (test report – defect tracking – defect analysis – final)
- A test process has 7 activities:
    1. Planning
    2. Monitoring & control
    3. Analysis (test scenarios/conditions are done here)
    4. Design (write test cases according to test scenarios)
    5. Implementation (organize test cases and test environment)
    6. Execution
    7. completion

Test Levels

09 March 2022 12:34

- Test levels are groups of test activities that are organized together.
- Each test level is an instance of the test process.
- There are 4 major test levels:
    1. Unit
        a. Also called component/module testing
        b. tests anything that is separately testable.
        c. The developer does this testing.
    2. Integration
        a. Unit/system test are integrated together. It has 2 types:
            i. Component integration – separating units are tested. Done by dev.
            ii. System integration - separate systems are tested. It is a big project so cannot be done by dev alone so it is done by testers.
    3. System
        a. focuses on behavior of the whole system/product. Also tests for non-function behaviors.
        b. Most important part of testing.
        c. The info that stakeholders need is provided by system testing.
        d. The test environment should test on similar environment that the user will use.
        e. Example if the product will we sed on win 7 then the test should be done in win 7.
        f. Find as many defects as possible.
    4. Acceptance
        a. Finds if the product is acceptable to final release. It has 2 types:
            i. Alpha – bring customer to the company and make them test the product in front of us.
            ii. Beta – customers test product at their own premises.

Testing Types

09 March 2022 12:37

1. Functional Testing
    a. Tests what the system does
    b. Usually answered with a yes/no
    c. Eg. Can the user log in using his user ID? Is the password hidden when enters it?
    d. Most dominant type of testing
    e. Can be combined with black box / white box testing
2. Non-functional testing
    a. Tests how well the system works
    b. Cannot be answered with a yes/no
    c. Usually measured as a range
    d. Eg. How long does the website takes to load? How many users can the server handle?
    e. Not that common
3. Black box testing
    a. Testing without knowing the internal structure of the system.
4. White box testing
    a. Testing with the knowledge of the internal structure of the system
    b. Requires knowledge of programming language
5. Dynamic Testing
    a. Testing that happens inside the product
    b. Eg. Testing the hardware manually, testing the software by coding...etc.
6. Static testing
    a. Testing that does not include execution.
    b. Only needs review.
    c. Only notes are taken
7. Confirmation testing (also called Retesting)
    a. Testing after debugging for confirmation that the bug is resolved.
    b. We do not trust the dev that he has completely solved the debug.
8. Regression testing
    a. Testing unchanged area that are not affected by previous changes.
    b. Eg. if the criterial for password changed in the registration page we will test the login page to see if it affects the passw ord change their too.
9. Smoke testing (also called build verification testing)
    a. Testing main areas of the product to ensure that the build is stable enough to proceed with testing.
    b. Tests the most basic functionality of the product
    c. Takes the least amount of time to perform
    d. If this fails we do not do anything else and return the entire build back to the developer.

Test Scenario

09 March 2022 12:39

- A test scenario is defined as any functionality that can be tested.
- Also called test possibility or test condition.
- Steps to create test scenario:
  - Study the required document. Some are:
    - BRS – Business requirement specification
    - SRS – Software requirement specification
    - SUT – System under test
  - Isolate every requirement and test them separately.
  - Abuse certain scenarios to get defects that may pass the common eyes.
  - Ensure the scenario taken over every user flow and business flow.
  - After the above is done create a RTM (Relation traceability matrix) to ensure every requirement is mapped to the test scenario.
  - RTM is necessary if the test scenario or case ends up with some changes. Sing an RTM in this case will help map the requirement changes.
  - Get your work reviewed by supervisor.

Test Case Writing

09 March 2022 14:46

- Most important document of testing
- A test case is a set of preconditions, inputs, actions, expected results and post conditions developed for a test condition.
- A test case consists of:
  - Test case title
    - The title of the test case
  - Preconditions
    - Specify the pre conditions required before executing
  - Test steps
    - Must be short and clear
  - Expected results
    - Result expected
  - Test Scenario
    - Also called test suite
    - A single test scenario can contain multiple test cases
  - Test environment
    - Hardware required. Eg Win 10, intel processor, 4GB ram...etc.
    - Software required. Eg OS required...etc.
    - Network required. Eg. 5G required, Wi-Fi speed above Mbps...etc.
  - Actual Result
    - Should not be filled until the test case has been executed
  - Status
    - New – The test case is not executed
    - Pass – Expected and actual result is same
    - Fail – Expected and actual result is different
    - Blocked/skipped – The test case cannot be executed. If a dependent test case failed the following test case gets blocked.
- A positive test case is a test case that follows all the laid out guidelines of the scenario.
- For a test scenario with 'n' guidelines the minimum number of good test cases required to cover all the positive and negative test cases for that scenario is 'n+1'.

Bug Reporting

09 March 2022 15:25

- A defect report is the documentation of the occurrence, nature and status of a defect.
- A defect in a work product is when it does not meet its requirements.
- Also known as deficiency, bug, fault
- A bug report is written in:
  - Title
    - Section -> Description (syntax)
  - Steps to reproduce
    - Must be very specific.
    - Should be in bullet points
  - Expected result
  - Actual result
    - Unlike in test scenario here it is filled as the steps are executed.
  - Test environment
    - Hardware
    - Software
    - Network
  - Screenshot/video
  - Must show the whole screen
  - Priority
    - Critical (eg. Login is not working, page crashes...etc.)
    - High (eg. Profile image is not there, page is slow...etc.)
    - Medium (eg. Landscape mode is not working...etc.
    - Low (eg. Spelling mistake, image misalignment...etc.)

Types Of Defect

09 March 2022 16:05

1. Functional
    a. A functionality in the application that is not working
    b. Eg. The password functionality is not working
2. UI defect
    a. Visual defect
    b. Eg. The icons are not displayed properly on the screen
3. Content
    a. A placeholder/inner HTML defect
4. Performance
    a. Eg. Site is slow, video is not playing
5. Suggestion
    a. Not a defect
    b. From a tester's perspective the thing can be done in a better way
    c. Eg. The background color would look better in pink

Test Reports

09 March 2022 16:12

- A report that summarizes everything done in testing.
- Has two types:
  - Test progress report
    - Produced at regular intervals
    - Produced during the testing phase and requires a decision to be made on the alternative approach to be carried out.
    - Also known as "test status report"
  - Test summary report
    - Produced at completion of milestones
    - Provides evaluation against a criteria

Interview Quiz

09 March 2022 16:41

- **What is the difference between SDLC and STLC?**
  - SDLC is the steps to develop a software. Whereas STLC is the steps to test a software
- **Which of the following test levels is done by testers?**
  - System testing
- **Which of the following is a correct sequence of steps that the defect goes through from discovery to resolution?**
  - New – assigned – open – fixed – retest – closed
- **Which of the following is true about test cases and test scenarios?**
  - Test scenario can be tested with more than one test case
- **Which of the following is a functional testing?**
  - Test the main workflows of the application to make sure they give the expected outputs
- **What is the difference between validation and verification?**
  - Verification can be performed using inspection, walkthrough, and technical reviews, while validation can be done using functional, system and smoke testing.
- **When should the software testing start?**
  - When the requirement are gathered
- **If we don't have clear written user requirements, which of the following can be an alternative option to test a software?**
  - If there is a prototype, a UI design, some user stories or wireframes, gather those requirement and try to understand them in order to write the test case.
- **Which of the following is NOT true about exploratory testing?**
  - It requires pre written detailed test cases before test execution.
- **Impact analysis is used to?**
  - Determine how much regression testing will be required
- **Reviewing the code is considered as?**
  - Static testing
- **According to the sequential software development models, which test cases are designed first?**
  - Black box testing
- **Which of the following best describes use-case testing?**
  - A technique that helps to identify test cases that exercise the whole system on a transaction by transaction basis from start to finish.
- **Why do we use boundary value analysis?**
  - Because the behavior at the boundaries of equivalence partitions is more likely to be incorrect than behavior within the partitions.
- **What is the main goal of the requirement traceability matrix?**
  - To validate that all requirement are checked via test cases
- **Tool driven evaluation of the code or other work products is called?**
  - Static analysis
- **Which of the following is true about test plans?**
  - There can be more than one test plan for each testing level or type.
- **Which of the following is the main difference between test progress reports and test summary reports?**
  - Progress report is prepared during test activity, while test summary report is prepared at the end of test activity.
- **Which of the following is not one of the mistakes the testers tend to do?**
  - Writing poor requirement documents
- **Which of the following is one of the recommended behaviors when dealing with a developer who rejected your reported defect?**
  - Return to the projects work/backlog
- **Which of the following statements is correct?**
  - Mistake leads to a bug that triggers failure
- **Which of the following options contains the MOST important components of a defect report?**
  - Steps to reproduce – expected result – actual result – screenshot
- **Which of the following is the correct order of the steps of risk-based testing?**
  - Identify – Analyse – prioritize – mitigate risks
- **What is the difference between alpha testing and beta testing?**
  - Alpha is done in organizing site that beta is done from own location
- **Which of the following is NOT a benefit of test independence?**
  - Independent testers are more focused on testing as they work isolated.
- **Which of the following is a potential drawback of test independence?**
  - Independent testers may lack imp. Info.
- **What is the purpose of test techniques?**
  - Help identifying test conditions, test cases, data
- **When random/monkey testing is used?**
  - To see if the system will hold under adverse effects
- **What is the difference between negative testing and positive testing?**
  - Negative testing is when you put invalid input and receive error, while positive testing is when you put valid input and expect some action to be completed
- **When should we use decision table testing?**
  - When system implements complex business rules
- **When do we perform testing activities in the waterfall model?**
  - After the system is developed
- **When is testing performed in the V-model?**
  - In every phase simultaneously
- **What is the alternative name of the V-model?**
  - Verification and validation model
- **Which of the following is NOT a best practice when writing test cases?**
  - Write test steps in technical way
- **A test case can be added to multiple test suites.**
  - True
- **Test environment supports test execution with?**
  - Hardware, software, network

Black Box Techniques

10 March 2022 09:13

1. Equivalence Partitioning (EP)
    a. The system is divided into equal partitions
    b. Each partition has its set of test cases and expected results
    c. Example if we have to test a software that grades students. Then the partitions will be as follows:
        i. 80 – 100 gets A
        ii. 60 – 79 – gets B
        iii. 0 – 60 gets C
        iv. Additional we also make 2 more partitions at <0 and > 100 to test the system.
2. Boundary value analysis (BVA)
    a. Extension of equivalence partition
    b. Better than equivalence partition
    c. It tests the end cases of the system
    d. It has 2 types
        i. 2 value analysis (most companies use this)
        ii. 3 value analysis (but this is more efficient)
    e. Example if a system tracks how the student studies between 5pm – 7pm
        i. We test at 5pm
        ii. We test at 7pm
        iii. We do 3 value analysis at (4:59pm – 5pm – 5:01pm) & (6:59pm – 7pm – 7:01pm)
        iv. We do 2 value analysis at (4:59pm – 5pm) & (6:59pm – 7pm)

- Notes
    - Each value belongs to only one partition
    - Equivalence partition can be applied to both input or output values
    - Equivalence partitions have two types:
        - Valid – system is designed to handle it
        - Invalid – system cannot handle it
    - In equivalence partitioning coverage = number of partitions covered by test cases / total number of boundaries
    - Testing two values from the same partition does not increase the coverage.

3. Decision table testing
    a. Different combination of conditions result in different outcomes
    b. Make table – list conditions – write test cases
    c. Number of table rules (test cases needed to test the system) = answer to the condition question ^ no. Of questions.
    d. To write the decision table for "system that validates a discount":
        i. Number of questions = 4
            1. Buy items more than 100 $?
            2. Has a gold subscription?
            3. Resident of Dakota?
            4. Older than 18 years?
        ii. Ways to answer the questions asked = 3:
            1. Yes
            2. No
            3. maybe
        iii. Therefore number of test rule = 3^4 = 81 test cases required to test the system.
    e. Hence it is important to be very smart and specific about the table rules as too many test cases will be hard to track.

4. State transition testing
    a. Used when system has states (ON/OFF) and transitions (path between ON and OFF).
    b. Look at course number 65 for the ATM example (9 min video).

4 Values

10 March 2022 09:14

1. Individual and interaction over processes and tools
    a. Good relation between tester – tester
    b. Good relation between tester – developer
2. Working software over comprehensive documentation
    a. Working software is more important than simply writing things and documenting it.
3. Customer collaboration over contract negotiation
    a. If we agreed with the client to add 10 functionalities and later the client says he needs 11 we must collaborate with him to make sure he is satisfied.
4. Respond to change over following the plan
    a. Don't be rigid and adapt according to the situation
    b. Be very flexible

12 Principles

10 March 2022 12:18

1. Customer satisfaction is highest priority
2. Deliver working software frequently at intervals of
    a. 1 week (smallest interval)
    b. 3 months or more (largest interval).
3. Working software is the primary measure of progress.
4. Welcome the change in requirements (even it is very late in development)
5. Continuous attention to technical excellence and good design
6. The process should promote sustainable development
    a. The whole team must have the same knowledge and pace in the project
    b. No one should move at a pace such that others in the team can't keep up
7. Keep it simple
    a. Keep only those things that provide value
8. Build projects around motivated individuals.
9. The best designs emerges from self-organizing teams.
10. Developers and business people must work together
    a. There should be daily feedback between them
11. If it can be done face-to-face it should be done face-to-face
12. At regular intervals, the team should reflect on how to become more effective and adjust their behavior accordingly.

Whole Team Approach

10 March 2022 12:36

- Involve everyone
- Better if small team (3 – 9)
- Team is co-located
- Quality is everyone's responsibility
- Tester's role
  - Collaborate with the business representative and help them create suitable acceptance test.
  - Work with developer to work on testing strategy
  - Transfer and extend knowledge to other team members.
- Power of three
  - It is the concept of involving tester, dev, and business representative in all discussions related to the feature.
- Daily stand up meeting
  - Meetings that support the whole team approach and which involves all members of the team.
- Early and frequent feedback (continuous integration)
  - When sequential development is used the customer does not see the product until the project is nearly completed. Therefore at that time it is too late for the dev team to make any other changes.
  - Therefore early feedback is imp as it focuses on functionality with highest business value and can manage risk.
  - It maintains transparency in the team.
- Collaborative user story creation
- Early feedback is not a technique that can be used here.
  - Consists of "As a _____", "I want to_____", "so that I can_____".
  - User story is written to capture the perspective of dev, tester and business representatives.
  - In sequential development, the user story creation is done through formal reviews after each stage.
  - How each user story is written may vary but the approach should be simple and sufficient.
- 3C Concept
  - Card
    - Physical form of user story. Identifies requirement, criticality, expected development, test duration. It must be accurate as it is used in product backlog.
  - Conversation
    - Explains how the software will be used. It can be verbal or documented. It begins during pre – release phase and continues when the story is scheduled.
  - Confirmation
    - The acceptance criteria in the confirmation is used to confirm the user story. Both positive and negative test must be used to cover the criteria. Various participants play a role of a tester here, eg. Dev, security team, specialists, quality checkers. Once a story is confirmed the acceptance criteria tested becomes satisfied.

1. *Note that in the above points business representative is not the same as client.*

INVEST

10 March 2022 13:37

- INVEST is a way a tester can test the user story:
  - I – Independent
    - Requirement can exists outside user story and still be meaningful
  - N – Negotiable
    - The story should written in lay man's term so that dev and client can work on it.
  - V – Valuable
    - Should have value
  - E – Estimable
    - It should be possible to estimate how much time it will take to implement the requirements.
  - S – Simple
    - The user story should be small and simple because it is meant to be developed in a short time.
  - T – Testable
    - The user story should be verifiable against some criteria in order for it to be accepted.
- For an example look at material number 88.

Scrum

10 March 2022 13:47

- It is an agile management framework which contains constituents instruments and practices.
- Most famous methodology used in Agile
- See material 89 for example explanation.
- A Scrum consists of:
  - Product increment
    - Each sprint has a shippable product
  - Product backlog
    - The product owner manages a prioritized list of planned product items (called product backlog). This when moves from one sprint to other is called backlog refinement.
  - Sprint backlog
    - The highest priority items from the product backlog of a sprint is called sprint backlog.
    - Evolving the backlog from time to time is called backlog refinement.
  - Definition of done
    - To make sure of the proper release of the product, the scrum team discusses and defines appropriate criteria for sprints.
  - Timeboxing
    - Only those tasks or features that the team expects to finish within a sprint.
  - Transparency
    - Transparency is maintained by reporting daily scrums.
  - Scrum Master
    - Ensures that scrum rules and practices are followed without any violation.
  - Product Owner
    - Represents the customer and maintains product backlog. Is not the team lead.
  - Development Team
    - Does the development and test of the product and is self-organized + cross functional.
- Scrum as opposed to XP does not dictate software development techniques and does not provide guidance as to how the testing should be done.

1. *Sprints can also be called iterations*

Kanban

10 March 2022 14:56

- It is a management approach used in agile product.
- Not that popular.
- The main objective is to optimize the work flow within a value added chain.
- Consists of
  - Kanban Board
    - Each column in the board shows a station which contains list of activities.
  - Work in progress limit
    - The amount of parallel strictly tasks that is strictly limited.
    - Controlled by max num. Of tickets allowed for a board.
  - Lead time
    - Continuous flow of tasks by minimizing the avg time for one complete stream.
- Kanban has some similarities to Scrum:
  - Both visualize the active task
  - Both provide transparency of product
  - Both use backlogs for tasks not yet scheduled (in waiting)
- Kanban and scrum have some difference:
  - Sprints are optional in Kanban.
  - Kanban allows item-by-item release.
  - In scrum all tasks are synchronized within a sprint.

Basics

10 March 2022 12:22

- Mostly used by product owners and business analyst
- A Jira project template by default a Kanban template
- Jira is good for...
  - Reporting defects
  - Static testing
  - Reviewing things like:
    - Activities
    - requirements
    - Product backlog
    - SRS
    - BRD
    - User story
  - Writing acceptance criteria

- JIRA has 3 types of projects
  - Kanban
    - Visualizing flow of work
  - Scrum
    - Sprints, iteration, product backlog, user story, product owner....etc.
    - 70%+ of developers use this type.
  - Bug tracking
    - Tracks defect lifecycle

- In JIRA:
  - Issue
    - Anything we need to do (user story, epics, tasks, subtasks)
  - Defect
  - Fault
  - Bug

- Team management in Jira
  - In team-managed projects, you can have only one running sprint, while in company-managed projects, you can have multiple parallel sprints.

- JIRA Workspace
  - Where can we find the “Create component” option? - From the “Components” page
  - After creating a component, you can find it in the “Backlog” page. - False
  - The project consists of components which are divided into epics and each epic is divided into user stories
  - How can we create an Epic in Jira project? - From the “Create issue” button in the “Backlog” page
  - When creating an epic, we must specify its other linked epics - False
  - Which of the following is not a valid way to create a user story? - From the “Create user story” button in the “Components” page.
  - The tester is responsible for writing the acceptance criteria for user stories.
  - A user story can be part of more than one component.
  - Story points mainly refer to the estimated effort for the user story
  - The estimating story point is done by the tester, developer, product owner.
  - If a user story has 1d and 2h logged time, this means it has 10 hours.
  - To start a sprint click on "start sprint" in the "backlog" page
  - The sprint backlog can be found in "active sprints" section.
  - If we have incomplete sprint issues in a completed sprint, they will be moved to backlog.
  - A burn-down chart is a chart that shows the amount of work remaining throughout the iteration
  - Velocity chart shows all the committed and completed story points in all sprints
  - How can we create a new bug - Create an issue and choose its type as “Bug”
  - Why is it better to have a separate hardening or stabilization sprint? - In order not to distract developers by the found bugs and leave them focused on their work.

1. Note
   a. *Check module 94 to see how to create a project*
   b. *Check module 95 to see how to create components*
   c. *Check module 96 to see how to create epics*
   d. *Check module 97 to see how to create versions/builds*
   e. *Check module 98 to see how to create use story (see 101 for story points)*
   f. *Check module 99 - 100 to see how to create acceptance criteria*
   g. *Check module102 to see how to create sprints*
   h. *Check module103 to see how to check burn down/velocity chart*
   i. *Check module104 to see how to check bug reports, hardening iterations*

API & Webservice

10 March 2022 09:25

- API = application programming interface
- API is when you (interface) tell a computer (program) to run an application.
- The 3 major parts:
  - Request
  - Program
  - Response
- Web = internet | Service = API
- Web services = API that goes through internet
  - All webservices are APIs
  - All APIs are not webservices (some APIs are not on internet)
- XML, JSON are used to format data.
- REST, SOAP are used to transfer this data

HTTP

10 March 2022 16:31

- Hypertext transfer protocol
- Has 4 parts
  - Start line (compulsory)
    - Request
      - Needs a version (1.1 or 2)
      - Needs a method (GET/POST/DELETE)
      - Needs location and parameters
    - Response
      - Needs a version (1.1 or 2)
      - Needs status code
        - 2xx – good
        - 4xx – problem with source
        - 5xx – problem with API
  - Headers
    - Request
      - Needs a host (ex. www.google.com)
      - Needs a token (for security)
    - Response
      - Cookies
      - HTML
  - Blank line
    - Separates header from body
  - Body
    - Request
      - GET – nothing in the body
      - POST – data provided, XML, JSON in body
    - Response
      - Get back the requested
      - HTML webpage

XML & JSON

10 March 2022 16:40

- XML = extensible markup language
- A format used to send APIs
- Created by W3C (who created HTML)
- In XML tags don't mean anything
- Unlike XML, HTML is not extensible (can't extend to mean something else)
- Browsers understand XML

- JSON = JavaScript object notation
- Part of JavaScript that holds data
- Smaller code than XML

Types Of API's

10 March 2022 16:46

1. SOAP
    a. SOAP = simple object access protocol
    b. Uses WSDL (web service description language)
    c. SOAP is not stateless
        i. This means that if the program stops then there will be a major issue
    d. Consists of 4 major parts:
        i. Start line (POST - WSDL - HTTP version)
        ii. Header (Content type)
        iii. Bank Line
        iv. Body (XML formed using WSDL)

2. REST
    a. Easier that SOAP, more popular than SOAP
    b. REST = representational state transfer
    c. Since it is representational, the actual record is not sent. Only a representation is sent.
    d. REST is stateless
        i. This means that if the program stops REST will wait unless the program resumes.
    e. Uses JSON
    f. API order
        i. POST – create a JSON order
        ii. PUT – erase order and create another one
        iii. PATCH – keep the order and add another one
        iv. GET – receive order details
        v. DELETE – remove the order

Basics

10 March 2022 09:26

- Postman is a tool that you can use to test your own APIs
- One of the most common API testing tool
- Postman is a browser
- Module 121 – 125 shows how to use Postman
- Module 126 - 127 has two sample projects.

Types

10 March 2022 09:16

- Performance testing is an umbrella term, including any kind of testing focused on performance of the system under different volumes of load.
- Performance testing is defined by 3 things:
    i. Time behavior – ability of the system to respond to user inputs within a specified time and under specified conditions
    ii. Resource utilization – how much resource heavy the system/product is. Eg. RAM consumptions
    iii. Capacity – the capacity the system can handle. Eg. How many users can the server hold at any given time

- Types of performance testing:
    1. Load testing
        i. Focuses on the ability of the system to handle anticipated and realistic loads.
    2. Stress testing
        i. Same as load testing
        ii. But we stress the system much more
        iii. Load testing tests anticipated load and stress testing goes way beyond that amount.
        iv. Eg. Load testing is done for 1,00,000 users on a server, stress testing will be done by testing 10,00,00,000 users.
    3. Scalability testing
        i. Testing to see if the system can meet future requirements efficiently.
    4. Spike testing
        i. Like stress testing
        ii. The load happens for a very short span of time, when the load spikes momentarily.
    5. Endurance testing
        i. How long the system can handle a particular load on it.
    6. Concurrency testing
        i. When many users do the same thing at the same time
        ii. Eg. Every single Instagram user is changing their PF at the same time.
    7. Capacity testing
        i. How many transactions the system can the system handle with no problem.

Concept Of Load

10 March 2022 17:49

- Load generation
  - 4 ways to apply load:
    i. User interface
       1. Most basic
       2. An adequate approach if only small number users
    ii. Crowds
       1. Needs large number of testers that will act as real users
       2. We can freelance them
       3. Can be from different geo location
       4. Very costly
    iii. APIs
       1. Less sensitive to changes in UI
       2. Eg. Using tools like J-Profile
       3. Done when there is frequent changes in the API
    iv. Communication protocol
       1. Very large number of users in a repeatable and reliable manner.
- Load Profiles
  - Creating load profiles – a load profile specifies the activity which a system being tested may experience in production.
  - It consists of pre-defined instances/operation profiles that mimic the actual user behavior.
  - A correct load profile needs:
    - Performance test objectives
    - Operational profile which accurately represents individual usage patterns
    - The quantity and time distribution with which operation profiles are executes. It depends on:
      1. Ramp-up – steady increase of load
      2. Ramp-down – steady decreasing load
      3. Steps – instantaneous load changes
      4. Predefined distribution – mimicking daily/seasonal business cycles
  - For load profile examples see module number 131.

Statement Coverage

10 March 2022 09:16

- White box is mostly used by developers
- Also called as architectural/structural testing

- Statement coverage covers all the statements in side our code
- Each line of code is called a statement
- Statement coverage = no of statement tested / total number of statements
- E.g.
  - If your code has 5 if statement (I.e. 10 lines of code)
  - And our test passes only 2 of the if statement (4 lines of code)
  - Then statement coverage = 4/10 = 40% statement coverage.
- Not that efficient type.

Decision Coverage

11 March 2022 10:14

- Also called branch coverage
- We think about the decision part of the code
- Decision coverage = decision outcomes covered / total number of decision outcomes.
- E.g.
  - If there are 3 "if clauses" and 2 "else clauses"
  - And our test case passes through 3 of the if cond. And 1 of the else cond.
  - The DC = (3+1 / (3+2) = 4/5 = 80%
- Decision coverages better and more efficient than statement coverage
- It is more commonly used
- If DC is 100% then it is guaranteed that Statement coverage is also 100%
- SC is a sub set of DC.

Condition Coverage

11 March 2022 10:24

- Does the same thing as SC and DC, but in a more through way
- E.g.
  - If a code has 2 "if" and 3 "else"
  - Then in condition coverage the test cases should be tested in such a way the each test case of "if" statement "passes and fails" the condition and each test case of "else" statement "passes and fails" the condition.
- We don't care about the % of coverage.
- Finds more bugs than SC or DC.
- Used when an "if/else" statement has one or more "or, and, not" condition within.

Path Coverage

11 March 2022 10:25

- Test all possible paths in the code
- If a code has 2 "if statement" and a "switch" statement with 7 cases. Then the test cases should be such that all 2 cases of "if" and 7 cases of "switch" are being tested.
- We don't care bout % coverage.
- All condition statement must be tested.
- Better than DC as it coverers every path possible.
- Used when an "if/else" statement has one or more "or, and, not" condition within.
- Although if there are way too many paths (eg a for loop with n = 10,000) than it is better to go with DC because than the pat h coverage becomes a type of exhaustive testing.

MCDC

11 March 2022 10:26

- Modified Condition-Decision Coverage
- Mix of DC and CC
- Look at module 177 for example.

Loop Testing

11 March 2022 10:25

- Used when the code has a loop.
- If there is a nester looping, we take test cases such that we test the outer loop first and then the inner loop
- Look at module 178 for example.