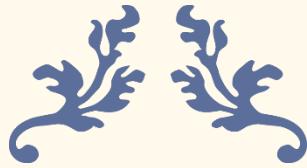


"a community-led career success platform for Testing professionals, providing peer-to-peer discussions & guidance, assessments & certification, interview preparation, upskill trainings and matching jobs - all at one place"



INTERVIEW PREP E-BOOK

stay ahead of the competition



parikshram.com

PARIKSHRAM TECHNOLOGIES
Bengaluru

Preface

Hi,

First, Thank You for showing interest. If you are reading this, you are already one-step ahead in your path to career success. “Job Switch” is one of the crucial steps in moving up the ladder – both professionally and financially.

But,

Interviews are a lot different than your actual day-to-day work. Interviewers assess you on multiple parameters like technical, managerial, team, attitude, communication, process, etc. Every organization, every project, every team might have different requirements when hiring. That’s when it becomes even more important to PREPARE well for your next interview.

Note: Interview Prep is **NOT a replacement to Upskill Trainings**.

That’s right. The target of interview preparation is clear, to get ready to face the next interview – to avoid nervousness, be confident, know the tips & tricks, and prepare for the commonly asked Qs. But it doesn’t in any way replace the need for upskill trainings.

Thank you to all the people connected with Parikshram (formerly Software Testing Studio) – whose inputs, queries, tips, experience, knowledge, & learnings over the last 3+ years helped in preparing this one-of-a-kind eBook. You are all awesome!

Since I started Software Testing Studio (now Parikshram), I have learned that for many people - having a support, constant motivation, answer to common queries & questions is a great booster to keep the momentum going. So, this book is NOT just about **technical Q&As but also about non-technical, HR, career & job search tips & tricks**. Hope it helps you in some or the other way!

DO's – Read each & every page starting to end.

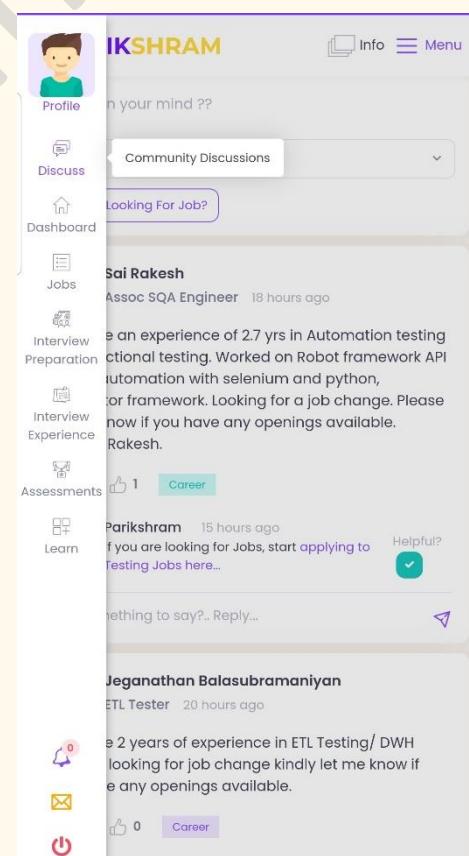
DON'T – Skip any page starting to end.

Simple! Let's get you started....



Discussions and
Guidance

Parikshram.com,
for all things
Testing!



The screenshot shows the Parikshram platform's user interface. On the left, a vertical sidebar lists navigation options: Profile, Discuss, Dashboard, Jobs (selected), Interview Preparation, Interview Experience, Assessments, and Learn. The main content area displays a job listing for "Manual Testing" in Noida. The job requirements include: Good Experience in Functional Testing, System Testing, Regression Testing, Load Testing and Performance Testing; Basic Understanding of relational database SQL Server; Understanding of Fiddler; Experience for WEB API Testing; and Ability to review business / software requirements and provide inputs. An "APPLY" button is visible at the bottom right of the job card.

QA-Testing Jobs, Directly recruiter inbox!

Interview process &
Qs asked in MNCs to
Start-ups!

The screenshot shows the Parikshram platform's user interface. On the left, a vertical sidebar lists navigation options: Profile, Discuss, Dashboard, Jobs (selected), Interview Preparation, Interview Experience (selected), Assessments, and Learn. The main content area displays a section titled "Share your experience". A job listing for "Automation testing" in Bangalore is shown, with details: 2 rounds, 3.5 years experience, Service Based, and a status of "Waiting". Below the job listing, there are several interview questions listed under "Interview Experience": "Tell me about yourself", "Explain the framework", "Write the code for data driven", "Full form of XSSF", "Write without Fileinputs...", and "Add more". At the bottom right, a profile card for "Aditya satapathy" is shown, with the title "Associate consultant".

The screenshot shows a user profile for 'IKSHRAM'. The main content area displays a course titled 'Agile & Scrum Essentials' with a yellow background image of a person running. Below the image, the course title is displayed along with '2 learners' and a price of '\$255'. A detailed description follows: 'all important concepts about Agile & Scrum, including multiple practice questions.' The course includes 3 chapters, 14 practice MCQ's, byte sized lessons, lifetime access, mobile & web friendly, and a certificate on completion.

Get certified,
MCQ assessments!

Byte-sized, Right-sized Learnings

The screenshot shows a user profile for 'IKSHRAM'. It lists two courses: 'Agile' and 'GIT'. The 'Agile' course has 805 times attempted and includes 8 chapters, 160 questions, 160 time (mins), and 75 passing %. The 'GIT' course has 120 times attempted and includes 3 chapters, 75 questions, 75 time (mins), and 75 passing %. Both courses feature an 'Explore' button and an 'Assess your knowledge' section.

Table of Contents

Software Testing (Theoretical)	6
Situational & Practical	43
Database & SQL Qs	99
JAVA Qs.....	109
Java Programs.....	129
SELENIUM, TESTNG & Cucumber	147
TEST AUTOMATION	196
API TESTING	242
Interview Experiences	261

Software Testing

(Theoretical)

What is Agile Testing?

A testing practice that follows the rules and principles of agile. Unlike Waterfall method, Agile testing can begin at the start of the project with continuous integration. Agile Testing is not sequential like Waterfall but continuous, i.e., testing happens in parallel with feature development as part of every sprint - goes hand in hand with development work and provides an ongoing feedback loop into the quality. It is a collaborative effort between testers, developers, product owners and even customers.

Why we use Agile Testing?

- Open & Honest Communication instead of documentation.
- “Just in time” requirements elaboration enables earlier initiation of development (without upfront documentation).
- Short sprints and constant feedback - easier to cope with the changes or course corrections.
- Transparency: Frequent status updates through regular meetings (issues, if any, can be identified in advance).
- Engagement: Collaborating with stakeholders - keeping them involved and engaged throughout projects.
- Self-managing and self-organizing team allow people to be creative & innovative.
- Environment of support and trust increases people’s overall motivation and morale.
- Continuous improvement via sprint reviews and retrospectives.
- Better Product: Visibility & continuous feedback ensures relevant & quality product.
- Business Value: Incrementally delivering value by prioritization.
- Feedback: Sprint reviews let stakeholders see working products & provide feedback.
- Delighted Customers

What is Scrum?

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum consists of defined roles, events, artifacts, and the rules that bind them together. To know Scrum - learn about the roles, artifacts and events and how they come together.

Note: Scrum is not a methodology.

What is Sprint?

Sprint is one timeboxed iteration of a continuous development cycle. Within a Sprint, planned amount of work has to be completed by the team and made ready for review. The term is mainly used in Agile Scrum. Example, a 2-weeks sprint where a scrum team completes 4 user stories, i.e., ready for review at sprint end. It wraps all other Scrum events like Daily Stand-up, Sprint Review and Sprint Retrospective.

#Tip | Act on your job search metrics

Yeah, metrics is everywhere. Keep a track of your job search metrics, i.e., Jobs applied, Calls received, Scheduled interviews, Interviews attended and offers received. These numbers show you the real problem to be worked on.

The stage where you're experiencing a block. So, you know at which stage you need to investigate what's going wrong. No calls? Might be resume or notice or skills. Not clearing interview? Need to prepare well. Clearing the tech rounds but then no offer? Work on your communication, CTC negotiation, behavioural Qs. Everybody faces challenges at different stages during the job search, find yours & start working on it.

Difference between smoke testing and sanity testing with example?

Software Build

What do you think it takes to build a software? Yeah! The code. But it isn't just one single code file, generally there are multiple source code files. Now these source code files need to be compiled & combined into a single executable file which can then be deployed by the release team. This process of combining multiple source code files into a single executable file is known as 'Software Build'. As you might have guessed, before being delivered to the client a software undergoes multiple changes (defect fixes), i.e., multiple 'Builds'!

Smoke Testing (General Health check-up)

What if developers are too reckless? There is a defect at the very first step – User is unable to login itself. Yeah! You will say what about Unit testing, but usually developers don't follow the rules every time 😊 Smoke testing is the preliminary testing to reveal simple failures severe enough to reject a prospective software release. In other terms you can call it as 'Confidence testing' or the 'Build verification testing'. Smoke tests cover the MOST CRITICAL functionalities. The purpose is to reject a 'critical defect' build before the Test team starts detailed functional tests. Before starting the day, a daily build and smoke test is among industry best practices.

Note: The term "smoke test" refers to powering on a device simply to make sure it does not start smoking (indicating a major problem). It originated in the testing of electronic hardware.

Sanity Testing (Specialized Health check-up)

First of all, irrespective of Testing, Sanity check is a basic concept – a simple check to see if the produced output is rational (that the product's creator was thinking rationally, applying sanity). Extending the concept to software, after every change in a build Sanity testing is performed to ascertain that the particular changes are working as expected post which detailed tests are performed. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

Note: Sanity in general refers to the soundness, rationality and healthiness of the human mind. A person is not considered sane anymore just if he/she is irrational.

What are Testing levels?

Software Testing Levels is nothing but different stages of software testing. How do you build anything? Generally, a system comprises of multiple integrated components. First create all the components >> Join all the components to build a system >> Once the system is up & running, deliver it to the client. For an effective delivery, it is important to test the system at each level of development, i.e., during development, individual components, integration, complete system, integrated systems, etc.

Unit Tests (Developer)

Done by developers as soon as they develop a certain code, if it's working or not. The smallest independent and testable part of the source code is referred to as a unit. Individual units of code (classes, functions, interfaces, procedures, etc. or a group of these) are checked by respective developers before handing over the build for deployment. As you might have guessed – Unit tests are basic checks just to verify that the developed code is at least working.

Component Testing (QA)

Individual component or module testing by the QA team, without integration with other components. This is to verify that standalone components are working as expected. Say an e-commerce site will have different components like Search engine, Filter setup, Payment gateway, User authentication, Cart management, Catalogs, Order tracking, etc.

Integration Testing (QA)

Testing of the interfaces between different components, so as to confirm that different components can work together in harmony. E.g. Order tracking module expects User details to be passed to it in a certain format >> What if the interface isn't working correctly? i.e. There is a difference between the expected and actual formats used for communication? Integration testing aims to uncover any defects in the way components interact with each other.

System Testing (QA)

Rigorous Testing of the integrated system as a whole, to ensure that the overall product meets the business requirements specified. Generally, this is the most important of all the Software Testing Levels as the end-user is not bothered about the individual components or the integration. He/She is just bothered about the system as a whole – if it works as expected. System testing is typically performed by a specialized testing team and in an environment that is very close to the production environment.

User Acceptance Tests (Business Users)

Whenever we purchase an electronic or electrical item, we always ask the seller to first give a demo so as to check if it's working fine. In case of Software – it is not used by the client, instead by the end-users, i.e. client's business reputation is at stake if it doesn't work. In this case how do you think client would accept your tested system without a walk-through? As the name suggests – Business Users walk-through the system during the UAT phase to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery / release (mainly from user's point of view)

For Commercial off the shelf (COTS) software's that are meant for the mass market testing needs (to be done by the potential users), there are two additional types of acceptance testing:

- **Alpha testing:** Done by potential users, at the developer's site. Potential users or members are invited to use the system and report defects.
- **Beta testing:** Also known as pre-release testing, Beta test versions of the software are distributed to a sample of actual end-users to give it a real-world test and gather qualitative feedback.

What is difference between Epic, User stories & Tasks?

- **Epic** - something so big it probably won't fit into a sprint and should be broken down into stories. T-shirt sizing is a common way to size epics. Epics are usually defined during initial product roadmap and decomposed into stories in the product backlog as more is learned.
- **Story** - something actionable and small enough to fit in a sprint. These are story pointed and defined using INVEST criteria. Stories should deliver a vertical slice of functionality to the customer that is valuable and complete by the end of an iteration.
- **Tasks** - decomposed parts of a story that get into HOW the story will be completed. Tasks can be hour estimated if desired. Tasks are usually defined by the people doing the work (developers, QA, etc), whereas stories and epics are generally created by the customer or the product owner representing the customer.

Pairs to remember

- **Preventive** – Tests are designed at an early stage, i.e. finding infinite ways to test and break your software before it goes to production. Identify or forecast the risks early & plan accordingly.
- **Responsive** – as in tests are designed after software development is completed. You react as & when the problem (defect) occurs.
- **Regression:** It's like checking for side-effects. The code is developed >> Build is deployed >> Sanity is performed >> Full testing is done >> Defects are logged, fixed & retested. What else? Yeah! Truth is stranger than fiction, and so is the Software. You never know a change in one function (defect fix or enhancement or change request) can impact multiple areas of the software. It's our duty as a Test team to ensure everything (apart from the change) impacted is working as expected.
- **Retesting:** What do you do in testing? Obviously find & log defects. After that? Yeah! Developer will fix the defect. As a Test team you need to verify that the defect fix is working fine, in other words you need to 'retest' the defect based on its steps to reproduce.

- **SDLC:** Software Development Life Cycle, SDLC for short, is a well-defined, structured sequence of stages in software engineering to develop the intended software product. Phases involve: Requirement gathering and analysis >> Design >> Implementation & Coding >> Testing >> Deployment >> Maintenance.
- **STLC:** How do you identify what's wrong? Software Testing Life Cycle, popularly known as STLC is a well-defined process with different structured sequence of phases to test the intended software. Phases involve: Requirement Analysis >> Test Planning >> Test Design >> Test Environment setup >> Test execution >> Test Closure >> Post-implementation support.
- **Smoke Testing:** It's like the general health check-up. Smoke testing is the preliminary testing to reveal simple failures severe enough to reject a prospective software release. In other terms you can call it as 'Confidence testing' or the 'Build verification testing'. Smoke tests cover the MOST CRITICAL functionalities.
- **Sanity Testing:** It's like specialized health check-up. After every change in a build Sanity testing is performed to ascertain that the particular changes are working as expected post which detailed tests are performed. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.
- **Software Testing:** To make it right, you first need to identify what's wrong. And when it's about finding the wrong in software, we call it "Software Testing"!
- **'Software Quality Assurance'** – a set of administrative and procedural activities (e.g. process implementation, training, auditing, etc.) implemented in software engineering processes so that requirements and goals for a software will be fulfilled.
- **Static** as in immobile – the code is not executed. Rather than the code, requirements, design and other documents are manually reviewed to find errors (such as code flaws or potentially malicious code) before the test cases are actually executed to verify the functionality.
- **Dynamic** as in active – the code is executed. How? By exercising the different functional or non-functional flows of the AUT in run-time environment from the User interface (e.g., webpage). The objective – to confirm that the AUT is working in accordance with the client requirements.
- **Test Design:** Simple put – writing the test cases based on client requirements. But it's not as simple as it sounds. A proper process has to be followed for Test Design to ensure that none of the client requirement is missed and we have an optimum test coverage before advancing to the Test execution phase.
- **Test Execution:** The next step after writing the test cases would be to execute those against the application-under-test. Just do it. Go on to execute the Test cases to find defects. If required, do some ad-hoc tests as well. The bottom line is – To find defects!

- **Test Strategy:** Before commencement of any Testing project the management first devises a Test Strategy document covering the Methodology, Test Approach, Scope of testing, Types of testing to be covered, Environment details, Resource requirements, Test Cycles, RAID (Risk, Assumption, Issues & Dependency), etc.
- **Test Plan:** The Test Plan document lists your Test approach & all the details included in the Test Strategy – together with the timelines. What will be the timelines in case you follow agile methodology? What if it's Waterfall model? When will each test phase start & end? What will be the deliverables at each phase & corresponding reporting structure?
- **Scenarios:** 'What' is to be tested? As the name says Test scenarios cover the different situations / set-ups / states that need to be tested in order to sign-off the application. It is important to cover optimum scenarios to test an application effectively. Test scenarios are derived from the requirements or use-cases.
- **Cases:** 'How' to be tested. A set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as per the requirement. It details out the steps to be executed, expected and actual results as well. Test cases are derived from Test scenarios and one Test scenario may result in multiple test cases.
- **Verification:** In simple terms – 'Are we building the product right?' And how to ensure that? Simple – just verify the intermediary products (like documents) at each phase adhere to the guidelines or requirements imposed at the beginning of the project. Verification is more inclined towards development best-practices and conformance to requirements.
- **Validation:** In simple terms – 'Are we building the right product?' And how to ensure that? Software Testing. A verified product is of no use if it's not defect-free, or cannot be used by end-users. Validation ensures that the product is fit for use and satisfy the business needs. Validation is more inclined towards Testing and end-user perspective.
- **Alpha Testing:** Before moving to Beta phase, Alpha testing is performed at the developer's site by an independent team of testers (can be from client side). As it is performed at developer's side – it is done in a testing environment and not the actual real-time production environment.
- **Beta Testing:** You might have heard this popular term. No matter how carefully you test, there's nothing like the real-world test of having other people use the software. A Beta release means rolling out not-so-perfect copy of the software (in real-time environment) to a larger set of Users before you actually 'Go Live'. Beta testers (client users or set of actual end-users) try the software, use it consistently for a specific period of time and report back any issues, bugs or feedback about it.

- **Black-box** method treats the AUT as a black-box (no knowledge about its internal structure). Result – We are not bothered about how the internal structure of the application is maintained/changed until the outside functionality is working as expected (as per requirements).
- **White-box:** Since it's a White-box >> we can see what's in it, i.e. the internal structure, and use that knowledge to expand the coverage to test every possible flow at code-level. For example – Statement coverage, Branch coverage or Path coverage.
- **Severity:** The extent to which a defect affects the application-under-test. Different levels of severity – Critical >> Major >> Medium >> Minor >> Cosmetic.
- **Priority:** The importance of defect. The extent to which a defect affects the business. And why will you want to prioritize defects? Obviously to define the order in which defects should be fixed. Different levels – Critical >> High >> Medium >> Low.
- **Formal Review:** Since it's official – it has to be documented for effectiveness. It involves a moderator that drives the complete Inspection activity, and documenting the results based on a pre-defined checklist.
- **Informal Review:** First of all – it is not documented. It's like calling a peer-friend to review the code or test cases that I have written. Since it is casual, the effectiveness cannot be measured. It is often used for cost-effectiveness.
- **Functional:** The most-essential Test type, Functional testing focuses on testing the software against design document, use cases and requirements document – that it correctly performs all its required functions.
- **Non-functional:** Here, you are not testing the software for any functionality. Instead the focus is on non-functional aspects like performance, load, accessibility, localization, security, reliability, recovery, etc.
- **Internationalization:** What if your software is hard-coded to support just one language? Say English. How will you target German (or any other local-language) end-users into using your software? Nah! You don't have a chance. The way out – Internationalize your software, i.e. remove the hard-coding and make the design generic such that it can support any number of languages and other geography-based features like currency, date format, etc. Officially “Internationalization is the process of designing a software application so that it can potentially be adapted to various languages and regions without engineering changes.”
- **Localization:** Now that you have internationalized the software, you need to add region-specific details for your target geographies. Say you want to add text-translations for Portuguese, German, French but not Chinese. I.e. though it's internationalized, but your software is localized for Portugal, Germany and France but not for China. Officially “Localization is the process of adapting internationalized software for a specific region or language by adding locale-specific components and translating text.”

- **Manual Testing:** I hope this is self-explanatory. No automation tools like Selenium or HPE UFT are utilized to test the AUT. Instead, the designed test cases are run manually by the testers, i.e. manually checking different functional / non-functional flows.
- **Automation Testing:** Wouldn't it be great if computer can run the tests on an application all by itself? Yeah! Software-testing-Software J Automation testing refers to a test method wherein tools like Selenium, UFT, JMeter, LoadRunner, etc. are utilized to script (or record-&-play) the test cases which can then be run by the computer at any time.
- **Positive Testing:** Simply put – The system is validated against valid input data. The intention is to check if the system behaves as expected for a valid input (& not showing an error). Positive Testing proves that the application-under-test meets the requirements and specifications.
- **Negative Testing:** Just the reverse – The system is validated against invalid data. The intention is to analyze the stability of application-under-test when invalid input data is entered. The application shouldn't crash, rather in most cases should display a proper error message.

Defect Life-cycle

Different 'status' that a 'defect' undergoes during its 'life cycle'.

Note: Defect Life cycle varies from organization to organization and is governed by the software testing process the organization or project follows and/or the Defect tracking tool being used. Let's look at the industry-standard. Please note that the change in status responsibility is mentioned in the brackets.

New (Tester >> Lead): It's self-explanatory. Say you identify a defect in the Application-under-test (AUT). What's next? Yeah! Logging it in a defect management tool. By default, whenever you log a defect, its status is 'NEW'. Mind you, it doesn't mean it's a valid defect and will be fixed – the analysis is not yet done!

Assigned (Test / Development Lead >> Developer): As a Test Lead or a Development Lead whenever you notice a 'New' defect in the system the next step is to analyze the defect for its correctness. If valid, assign it to a particular developer ('Assigned to' field) and change the status to 'Assigned'!

a. Deferred (Team Lead, Business Team): Deferred as in delayed or postponed. After initial analysis it is observed that a particular defect will not be fixed as of now, i.e., in this release. It will be fixed in some future release. Therefore, the Lead will change its status to 'Deferred' and it will not be picked up as part of this release but some future release. And why do you think a defect will be deferred? There are multiple factors and there can be multiple reasons – very low priority, lack of time, not part of current requirement, technical dependencies, etc. Generally, it's a standard practice to get sign-off from the business team before marking a defect as 'Deferred'. After all, they know the best about the business impact.

b. Rejected (Test / Development Lead): After all, ‘To err is human’. Even a tester can make mistakes. What if you used a wrong test data? What if your application understanding is wrong? Didn’t understand the requirement? Defect is already logged by some other tester? The functionality is out-of-scope? It is not a requirement? Wrong observation? Yeah! What if you as a tester have raised an invalid defect? Obviously, it will be rejected. If after analysis it is observed that this is NOT a defect, then Lead will change the status to ‘Rejected’ mentioning the details in the comment section.

Open (Developer): Once a defect is assigned to a particular developer – he/she starts working on the defect fix and changes the status to ‘Open’. Working as in analyzing the defect, investigating its root cause and rectifying the code.

Fixed (Developer): Once the code changes are done & verified in the development environment – the assigned-to developer changes the status to ‘Fixed’. What’s next? The fixed code needs to be deployed in the Test environment post which it will be verified by the Test team.

Ready for Test (Developer >> Tester): What’s the difference from ‘Fixed’? The Environment. Once the rectified code is deployed in the Test environment – it is available for the Test team to verify, i.e., retest. Post deployment, the developer changes the status to ‘Ready for Test’ and assign it back to the tester (who logged it). Tester verifies that the fixed code is working fine and the defect is no more reproducible.

Closed (Tester): If the rectified code is working as expected, i.e. the defect is no more reproducible – the tester changes the status to ‘Closed’. It means that the defect is fixed & retested successfully and is no longer reproducible. Happy Ending!

a. Reopen (Tester >> Developer): During retest you find that the defect is still present even after the fix. What to do now? Simple, just change the status to ‘Reopen’ and assign it back to the developer. The lifecycle starts again from status 3.

What is a Traceability Matrix? Or How do you ensure maximum Test coverage?

A traceability matrix is a document that co-relates any two baseline documents (section to section) that require a many-to-many relationship to check the completeness of the relationship, i.e. to ensure that everything within both documents is related in any way!

Purpose: Identification of anything that is not linked in two documents, which require further analysis.

Requirement Traceability Matrix

Applying the concept of Traceability matrix to requirements, i.e. one of the reference document will always be ‘Requirements Specification’. In simple terms, Requirement Traceability Matrix (commonly known as RTM) is a tool (mostly excel document) that traces requirement throughout the validation process, i.e., Requirement Analysis, Test Design, Test execution & Closure phase.

How?

Simple mapping – Requirement IDs >> Test Scenario IDs >> Test cases IDs >> Defect IDs (including final status)

Purpose: To trace that all requirements are covered in Test Design, Test execution & Closure.

- Client Confidence: First & foremost is to build the client’s confidence that the product is being developed & tested as per the requirements
- Ensure Test Coverage: All requirements have been covered in Test Design & Execution. The aim of any testing project should be 100% test coverage. Requirement Traceability Matrix helps in uncovering the gaps.
- Track Change Requests: Maintained throughout the lifecycle of the release, changes to the requirements are also recorded and tracked in the Requirement Traceability Matrix.
- Risk Management: ensuring every requirement is testable & eventually tested
- Maintainability: Any change in requirements can be tracked to corresponding required change in Test Design
- Product Quality: The Test coverage & corresponding defect status gives confidence to the client about the product’s quality
- Estimate Change Requests: Using Requirement Traceability Matrix, a Test Manager can easily gauge the impact and the amount of work required if any of the requirement is changed.
- Component’s Quality: Gauging which components in the system ‘were’ most flawed (most defects), another way to highlight the high-risk areas which needs thorough testing.

What Agile metrics do you capture? Any idea about Velocity?

Agile Velocity is an extremely simple method for measuring the rate at which scrum teams consistently deliver business value. In other words - How much product backlog effort a team can handle in one sprint? It’s the rate at which a team delivers stories from the product backlog, i.e., sum of estimates of delivered (i.e., accepted) features per iteration. It can be measured in story points, days, ideal days, or hours that the Scrum team delivers – all of which are considered acceptable.

A team delivers 20 and 30 story points in the first & second iteration respectively. The average velocity = Sum of Story points delivered / Number of iterations = $(20 + 30)/2 = 25$ Story points!

- Only the aggregate velocity of the team matters, and the phrase “individual velocity” is meaningless.
- Velocity is NOT productivity. Instead of a speedometer, it is more like a measurement of a team’s rhythm and should be used to monitor team health.

The goal is not maximized velocity, but rather optimal velocity over time, which takes into account many factors including the quality of the end product.

- Reporting, i.e., how much value is delivered (Story points/User stories)
- Resource and project planning (timelines)
- Adjusting project scope and business priorities, and forecasting release dates
- Identifying and removing obstacles that affect velocity

Do not try to over-complicate velocity – it really is a straight forward concept and a great deal of its value lies in its inherent simplicity.

How to write Test Conditions + Test Scenarios + Test cases

Testing terminologies can sometimes confuse even the most experienced of IT professionals. To a layman Test conditions, Test Scenarios, Test cases and Test suite might seem similar but there is a subtle difference between these terms which make a world of difference for a Software tester. Each of these implies a different level of detail and is used for a different purpose. Once a tester knows what each of these terms mean, they can figure out how to use them to describe the testing work that is done on a daily basis.

Test Condition [Functionalities]: What all conditions (functionalities) are you going to test? E.g., Home Page rendering | Different sections displayed – Header/Content/Footer | Main Menu Options & Navigation | Sub-menu Options & Navigation | All Navigation (All hyperlinks, some 50/100+) | Search functionality | Slider functionality | Location Detection | Any particular Business Logics | Language Translations | Device/Browser compatibility | Page Load performance.

Test Scenario [Business Flows within Functionalities]: It’s time to move from high-level to mid-level Test coverage. Test scenarios help in achieving a good test coverage by breaking down a functionality in different possible business flows, i.e., different ways to accomplish a task. It’s a short description about the business flow under test. Put yourself in the end-user shoes and figure out the real-world scenarios which can be performed on the application.

Test Cases [How to Test Business Flows]: Consists of a set of conditions or actions which are performed on the software application in order to verify the expected functionality of the feature. Here we describe the end-to-end logical flow of a specific requirement with test data, prerequisites and expected results. It consists of a Test case description, execution pre-condition, steps to follow, a set of input values, expected results and executed post condition.

Positive & Negative about Waterfall & Agile

Waterfall is disciplined by design and well documented. Suitable for projects where there are many interfaces and dependencies. It is easy to estimate costs, create timelines, and stick to deadlines. Since it is more plan-oriented, it is more secure and builds confidence of what will finally be delivered. Additionally, it has an easy learning curve to start working on a project ☺

The biggest catch - Client won't see a working product until late in the life cycle which means assurance of products being fit-for-purpose is demonstrated very late. Defining requirements in early stage is not easy and it's costly to make changes in later phases.

The Iterative design process of Agile gives it the flexibility to pivot when needed and reduces risk by constant user feedback during development. It is apt for projects where the end-goals are not clearly defined. Development/Testing is often more user-focused and reporting is effective via visual charts/Kanban/etc.

Agile is simple to understand in principle but hard to do well in practice. It requires high levels of collaboration and very regular communication between users. It's challenging on large projects or where co-location is not possible. Not that strong when dealing with lead times and major dependencies. It is hard to predict timelines or budgets. It is an inherent tendency to neglect documentation.

Agile and Waterfall are very different. Neither of it is inherently better than the other. You just need to understand which method is better suited to your project needs.

Client-Server vs Web based applications - Any idea what's the difference?

Client-Server: Application runs on the Client side (i.e., thick client) and access the remote server for information/service. Clients initiate communication sessions with servers which await incoming requests. Can be platform-dependent or independent based on the programming language used. E.g., Email Clients like Outlook or Chat clients. It requires installation on client's machine, i.e., runs as .exe.

Web-based: A platform-independent application that runs entirely on a Web browser (i.e., thin client). No installation needed. E.g., Gmail.

Note: A Web application is also a Client/Server application, i.e., client/server application is a kind of superclass. Just that 'Client' is always a 'Web Browser'.

#Tip | Always attend ALL the interviews

Should I skip this interview since I don't have this one particular skill-set? Or since I am not yet prepared? Or just out of fear of the unknown? Or for ANY other reason? NO!

Never under-estimate yourself. Always attend ALL interviews for which you are shortlisted. First, be happy that your profile was shortlisted among loads of applicants. Next, value every opportunity you get to attend interviews.

Why? Because it helps in gauging the market, interview trends, common questions and also builds confidence. Interviews are different than day-to-day work, and are a great source of learning. You might get rejected in few, but that's how it works 😊 If you work diligently, selection is just some interviews away. All the very best!

What are the different ceremonies/events in Agile-Scrum?

Sprint (or iteration) is the heart of Scrum. It calls for below ceremonies that bring structure to each sprint:

Sprint Planning: Held at the start of each sprint to define the Sprint Backlog (importing stories from the Product/Release backlog), i.e., items that can be completed in the current sprint. As you might have guessed, the Product Owner drives Sprint Planning as in which stories are highest in priority.

Daily Scrum: Presided over by the Scrum Master, Daily Scrum is a 15-minute stand-up meeting to synchronize the work of team members, i.e., what's done on the prior day, what needs to be done today, and identify any impediments. It is also a means to track Sprint progress.

Sprint Review: Held at the end of each sprint to demonstrate the added functionality. The goal is to get feedback from the product owner and other stakeholders to ensure that the delivered increment met the business need and to revise the Product Backlog based on the feedback.

Sprint Retrospective: Held at the end of each sprint to reflect on the completed sprint and identify opportunities to improve in the next – what went well, what did not and what can be improved. It allows the team to focus on its overall performance and identify strategies for continuous improvement.

Follow Scrum in your Project? What are the known benefits of Scrum?

Agile Scrum provides development, process, product and business benefits - achieving objectives in a leaner and more business-focused way.

- Open & Honest Communication instead of documentation.
- “Just in time” requirements elaboration enables earlier initiation of development (without upfront documentation).
- Short sprints and constant feedback - easier to cope with the changes or course corrections.
- Transparency: Frequent status updates through regular meetings (issues, if any, can be identified in advance).
- Engagement: Collaborating with stakeholders - keeping them involved and engaged throughout projects.
- Self-managing and self-organizing team allow people to be creative & innovative.
- Environment of support and trust increases people's overall motivation and morale.
- Continuous improvement via sprint reviews and retrospectives.
- Better Product: Visibility & continuous feedback ensures relevant & quality product.
- Business Value: Incrementally delivering value by prioritization.
- Feedback: Sprint reviews let stakeholders see working products & provide feedback.
- Delighted Customers

#Tip: Attitude, Flexibility & Communication matters

Once you clear the technical rounds, it is more about the attitude, flexibility, communication and team work. Never show that you are rigid about taking some particular responsibilities like only-automation OR only-manual. Never criticize your old or current team or company. Showcase your helping nature, thinking prowess (innovations/enhancements), team-work and flexibility in taking on additional responsibilities. And yes, communication (spoken English and putting across your ideas) is very important during later rounds.

Keep handy your real experience where you contributed technically to the project, helped a team member, handled a tough project, enhancements you brought, challenges you faced, etc. Later rounds are more about your “personal story” about different testing situations.

What's the different between defect status Fixed & Ready for Test? What about New, Open and Assigned? Deferred defect and a Change request?

Fixed: The defect has been fixed by the developer in their dev-environment. But not yet deployed to QA environment.

Ready for Test: The fix has been deployed to QA environment >> and is ready for QA team to retest.

New: QA team has just logged the defect into bug-tracking system. It is still new and has not been reviewed or assigned to anyone.

Open: The defect has been reviewed as valid (often by the defect manager).

Assigned: The defect is assigned to a developer for analysis & fixing.

Deferred: In simple terms 'Postponed'. Why? There can be multiple reasons like low-severity, timelines crunch, out-of-scope, etc. The decision to postpone a fix has to be approved by BA/Product Owner.

Change Request: The defect logged demands a change in existing requirements. A CR can be taken up for implementation in the current release or can be deferred to a future release.

What are some risks you mention in a Test Plan?

What are some common risks every project may face/have faced?

- Timelines. The first & foremost. Any downtime or requirement changes may impact the timelines.
- Resourcing. Skilled employees & required hardware/software.
- Scope. Any scope change needs to be analyzed & assessed.
- Third-party Dependencies. Availability of third-party systems & teams.
- Environment. Environmental downtime.

What is the front-end technology used in your current project?

It is important to know about the technologies being used in your current project. Be it front-end, database, clouds, deployment, SVN, etc. Talking about the different front-end technologies,

- **HTML (1990):** Provides building blocks to all websites, particularly text and images.
- **JavaScript (1995):** Used to make webpages interactive and provide online programs, including video games.
- **Flash (1995/96):** Multimedia platform used for authoring of vector graphics, animation, games and rich internet applications.
- **VBScript (1996):** Used to give functionality and interaction to web pages.
- **Ajax (1999):** Allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes, without reloading the whole page.

- **JQuery (2006)**: Takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- **Node.js (2009)**: For developing server-side and networking applications.
- **AngularJS (2010)**: JavaScript-based open-source front-end web application framework that extends HTML DOM with additional attributes and makes it more responsive to user actions.
- **Responsive Web Design (2010)**: Approach to web design which makes web pages respond to the user's behavior and environment based on screen size, platform and orientation, i.e. render well on a variety of devices and window or screen sizes.
- **ReactJS (2013)**: Create large web applications which can change data, without reloading the page.

What all do you include in a Test Plan?

As the name suggest – it's a plan about your Testing. How have you organized your Testing efforts? A Test plan should be able to answer the very basic structure of your QA activities, i.e. What, When, Where, Why, Which, How, Who and What Else.

- **What**: Scope defines the features, functional or non-functional requirements that will be tested. Additionally, a Test Plan also explicitly mentions the features that are considered to be out-of-scope!
- **When**: The Timelines. It can be a cycle/sprint structure or the different test phases within waterfall along with the milestones.
- **Where**: Details about the Test Environment.
- **Why**: The objective of your planned Test Efforts. To build customer confidence? Catch maximum defects? Provide optimum quality? Sign-off the Product quality?
- **Which**: Which all Testing types are to be undertaken? And which all tools will be used? Etc.
- **How**: What is the Test methodology? Test approach...Etc.
- **Who**: What are the roles and responsibilities of different team members like Test Manager, QA Analyst, Automation Engineer, Configuration Manager, Developers, Installation Team, Business Analyst, etc.
- **What Else**: All other important things like defect management, deliverables, risks & assumptions and open items (if any).

The idea is simple, focus on the value it offers – boil Test planning down to only the essentials and cut all fat and fluff. A comprehensive Plan gives customers the confidence that an efficient Test process is adopted to ensure optimum Quality!

#Tip: Automation Knowledge | but no Experience?

Many candidates put ‘Automation knowledge’ in their resumes in hope of getting a call. But that has become yet another common phenomenon. To stand apart, build your GIT repository with your practice code samples. It may be your practiced Java programs, automation scripts OR framework learnings. Mention the GIT URL in your resume summary. It gives interviewer an idea that you not only have the knowledge – but have practiced it as well – and are keen to put that practice to actual project work. Instead of just practicing – practice & maintain your GIT account.

As a sample project, you can even automate the sanity and few regression cases of your existing current project. While automation, you will learn a lot about test automation. Additionally, you can even showcase it as hands-on automation experience since you practiced it on your own project.

What's the difference between Test Strategy and a Test Plan?

Yeah! One of the common interview questions. Yet we tend to confuse. Why? Simply because there is no standard universal template – every organization and project are following their own standards.

Some don't even prepare a separate strategy, just a Test plan. Whereas others prepare two separate documents with almost same content 😊 Most of us have different understanding of a Strategy and a Test plan, that's because they are synonyms 😊 .

- **Test Strategy:** Predominantly it should answer the 'How' aspect. How are you going to test? It varies from the type of application (say Desktop, Web, Mobile, etc.) to the type of testing (Functional or Non-functional) to the process followed (proactive or a reactive). Whether you will use emulators or an automation tool or stub/drivers. Whether QAs will be involved from the project initiation phase itself? Do you need a separate environment? Etc.
- **Test Plan:** It is more detailed than a strategy, covering all the Why – When – Where – Who aspects including any Risks-assumptions-issues-or-dependencies.

What latest technologies are you aware of?

Technology is ever-changing. And recently it is changing at a rapid pace. It is really important that you are at least aware of the recent & up-coming technologies.

- **Virtual Reality:** an interactive computer-generated experience taking place within a simulated environment.
- **Block chain:** a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.
- **Robotic Process Automation:** Business process automation technology based on the notion of software robots or artificial intelligence (AI) workers.
- **Artificial Intelligence:** Intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and other animals.
- **Big Data:** Data sets that are too large or complex for traditional data-processing application software to adequately deal with.
- **Machine Learning:** Application of AI that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Note: Cloud Computing is a normal now 😊

#Tip | Be aware about the Tech trends

Always be aware about the latest technology trends. Test Automation is everywhere. Techies are talking about AI, ML, CI-CD, No-code automation, Big Data, Cloud, DevOps, VR, Block chain, etc. Know what it is all about. Just a brief understanding.

Even if you have not worked hands-on on few tools, at least know about them. What is the tool, where & how to use it? You will see that it helps in interviews as well, when you know about the latest technology & tools. It shows your awareness & eagerness to learn about new things happening in the technology space.

#Tip: Career Gap

Career gap is totally fine. Its normal. MANY people take a break – whatever be the reason. That doesn't mean you should under-estimate yourself.

Maybe you just need to brush up your tech concepts and start hands-on exercises. Showcase your hands-on experience in technology in your resume. If you have a genuine reason and are up-to-date, career break shouldn't be much of a problem.

Pricing Models

Money isn't everything but everything needs money.

When a customer hires a software company, they sign a billing contract. The pricing model used depends on the project.

- **Fixed-Price:** Finalized pricing based on an estimate of work that needs to be done. It ensures that a project is done and delivered within a specific timeframe and budget.
When the requirements are clear and customer has a pre-defined budget.
- **Time-and-Materials (T&M):** The customer is billed regularly (iteration-wise) on the basis of time & resources spent depending on the finalized hourly rate.
When the project is quite flexible and requirements change frequently.
- **Milestone Pricing:** The customer is billed when a specific scope of work is completed over a certain period of time, achieving a predefined milestone. I.e., approved payment for each milestone.

When the parties have a long-term and trusting relationship.

What are the biggest bottlenecks to Quality?

- The first & foremost – negligence, i.e., negligence of the higher management towards quality.
- Lack of training: not many orgs focus on training the employees on 'Testing as a process'. It's like dive-in and you'll learn.
- People choosing 'Testing' just because they think it doesn't need programming skills.
- Not adopting the shift-left testing approach.
- Illogical OR unnecessary usage of test automation. And vice-versa, i.e., not adopting automation where it should be.
- Focus on requirements-test design-execution-defects. Missing the bigger picture of complete product/solution being offered.
- Miscommunication OR Lack of collaboration between ALL.

QA Process Manual

"QA process document" – Do we have anything of this sort anywhere? Every company (or project) evolves over time to setup a QA process. Some of the pointers to take care can be,

- SDLC/STLC: what will be the delivery lifecycle? Which methodology to be followed? What all phases will be there & respective teams.
- Test environment planning and management.
- Team composition: It can be an agile team with scrum master-product owner-Dev-QAs OR a traditional setup of Project manager + BAs + QA team.

- Release management: What's your release cycle? At which phase are the testers involved?
- Stakeholder management: What are the client-facing roles and deliverables?
- Test planning & management: As a QA team, what is the process to be followed. Defining your STLC along with phase-wise activities + deliverables + roles might help here.

Go-No Go Meeting

By Definition: Before each public release all stakeholders meet to determine if the release criteria are met and we are good to deploy in production environment.

- **Participants:** make sure there is a representation (SPOCs) from all teams involved in the project - Dev, QA, release, DB, infra, client, etc. SPOCs as in who are the decision-makers.
- **Evaluation:** The most important aspect of a Go/No-Go meeting is to identify if there are any blockers for the particular release. I.e. blocking defects not fixed, not all requirements are demoed, QA test execution is not yet completed, release team & required infra is not yet ready, etc. It can be anything that can have an impact on production deployment and subsequent end-user usage.
- **The Exit criteria** OR checklist can vary on a project-to-project basis but few items to include can be requirements coverage, QA completion (including Unit test & UAT), Open defects, Risks (if any), infra readiness, team's availability, post-deployment support agreement, Roll-back plan, etc.
- **Documentation:** A Go/No-Go meeting should be documented in the form of evaluation criteria/checklist discussed and their corresponding RAG status (with comments) leading to the final Go/No-Go decision.
- **Communication:** The final outcome of the meeting should be communicated to ALL the stakeholders involved via Email.

What all important things do you include in your Daily Status Reports?

- **Highlights:** Important pointers to be conveyed about today's status.
- **Progress:** What progress has been made against the plan, i.e. it can be execution count, stories tested, value added, etc.
- **Defects List:** All defects raised along with respective priority, status, etc.
- **RAIDs:** What are the highlights, i.e. any risks, assumptions, issues or dependencies considered.
- **Downtime:** Did you have any downtime due to environment unavailability, OR a blocker?

What are stubs and drivers in Integration testing?

Stubs and Drivers are nothing but 'Jugaad' (placeholders/substitute) when you don't have a dependent component ready OR is not available.

Both have the same purpose to act as a placeholder for some component. The only difference being on which side that component is.

- **Stub**, i.e., the tail-side. If the next component is not available, you use a stub instead.
- **Driver**, i.e., head. If the immediate source component is not available, you use a driver as the ‘calling’ program.

Both stubs and drivers are dummy modules that simulate the behavior of missing component – for test purposes. Why wait to test parent component A if child component B is not available, just use a stub – and vice-versa.

What's Fallback Testing?

As the name suggests, fall-back is the “contingency plan for reverting to the old system or to an alternative emergency solution in the event of a failure of the new system during installation”.

You deploy a solution in production environment and boom! It stops working. What do you do? Simple – revert to the old system (fallback). But that’s not so easy. Scripts should be written to support the rollback of all changes done while installing the new build. And Fallback testing ensures that the old system works fine after rollback, that all new changes have been reverted successfully.

Why is it important? In case of rollback, you don’t want any residual changes impacting the functioning of old existing system.

Defect severity and priority difference with examples. Can you please give me real time examples or good examples?

- **Priority**: The ‘urgency’ to fix the defect.

How urgent is it to fix the issue from business perspective? Unable to login? Top priority. Misspelt brand name? Top priority since you don’t want to tarnish the brand image. Error in payment processing? Top priority. ‘Help’ URLs not accessible? Low priority since it can be fixed later, it is used very rarely.

- **Severity**: The ‘brutality’ of the defect.

What is the impact of issue from functionality perspective? Unable to login? Critical severity. Misspelt brand name? Low severity since it doesn’t impact any functionality. Error in payment processing? Critical severity.

- High-Priority-High-Severity: Login issues, payment issues, etc.
- Low-Priority-Low-Severity: One of the random labels in some page is misspelt.
- High-Priority-Low-Severity: Brand name misspelt. Data is saved but with an error message.
- Low-Priority-High-Severity: The application crashes at boundary values, say very long username/passwords. Which are rare.

It is important to clearly understand & update defect's severity & priority, from both business and functionality point of views.

What's 'Shift-right Testing'?

A method of continuously testing software while it is in immediate pre-release and post-release phases (testing in production). It helps software teams uncover new, unexpected scenarios that may not have been detected within the development environment. The goal of shift-right testing is to ensure correct behavior, performance and availability over the production use of an application.

Practices such as,

- Release validation
- API Monitoring
- Crowd testing
- Destructive/chaos testing
- CX-based testing (e.g., correlating user behavior with test requirements)
- Production monitoring, extraction of test insights from production data, etc.
- Feature Toggles | A/B testing

are just a few examples of how teams can extend their Continuous Testing culture.

Heuristics in Testing

Heuristics are simply a "Rule of Thumb", not based on fact but based on experience and sound judgment, its comparable to an educated guess, or following your intuition.

Heuristics can improve your approach to test design and execution and the underlying test techniques. For example, as a rule of thumb you should test boundary values.

- CRUD: a very useful and intuitive Heuristic.
- Personas: Different users use the software in different ways.
- Nouns & Verbs: Identify the Nouns and Verbs in system >> Randomly choose a combination >> find a closely related test case.
- Goldilocks heuristic: E.g., Data entry fields testing with too big data, too small and with entries that are more "typical" for that field.
- And many more...

As a tester, you often use your own heuristics - whether you are aware of it or not. This is partly because the use and development of heuristics are intuitive and innate. Focus on user permissions? Or trying all invalid data? Checking the happy flows first? Trying all buttons available on a page? Any pattern in your test activities is your personal heuristic.

What's Chaos testing?

"Regardless how encompassing your test suite is, once your code is running on enough machines and reaches enough complexity - errors are going to happen".

If this is true – then why not deliberately introduce chaos to ensure systems and processes can deal with the failure?

Candidate: Large distributed systems using cloud computing with a variety of services and processes designed to scale up & down.

The benefit? Design and build highly fault tolerant systems to withstand massive outages with minimal downtime.

How? Using tools. One of the tools is 'Chaos Monkey' - invented in 2011 by Netflix to test the resilience of its IT infrastructure. It works by intentionally disabling computers in Netflix's production network to test how remaining systems respond to the outage.

How does a client-server application work?

- **Client:** someone using a service. In the web world, browser can be seen as a client requesting info.
- **Server:** medium that serves something. A remote computer that provides info or access to services.

But how do they communicate? Yes, via a transfer protocol [say http request-response].

- Once you enter the URL - browser requests the DNS Server for the address of the Web server.
- DNS Server responds with the IP address of the WEB Server.
- Browser sends over an HTTP/HTTPS request to WEB Server's IP (provided by DNS server).
- Server sends over the necessary files of the website.
- Browser then renders the files and the website is displayed. This rendering is done with the help of DOM (Document Object Model) interpreter, CSS interpreter and JS Engine collectively known as the JIT or (Just in Time) Compilers.

There are both pros and cons for the client-server architecture, yet it is used widely.

#Tip | Compounded interest

Guess everybody is aware about the power of compounded interest. Same applies to 'Compounded Efforts' as well.

Small little learning & practice on a daily basis will surely result in a long-term huge benefit. Read a blog, watch technical video, practice programming, search about a tool, anything. Make it a habit. It might take time – but it will surely reap the benefits.

#Tip | Confident voice

Don't underestimate the importance of a telephonic interview. Don't forget to be in a quiet area during your interview - avoid places which are windy or noisy. Remember - your tone of voice is all you have, to convey enthusiasm and confidence in your abilities.

Wear a smile, it shows in your voice 

What are the two main types of databases?

Knowledge of database is important, both from development as well as testing perspective. Recently (e.g., social media) there is a load of unstructured data which cannot be accommodated in traditional Relational-DBMS efficiently. Hence, we get two main types of database,

- **SQL [relational]:** Table-based DB with predefined schema including tables-rows-columns. Use os structured query language (SQL) for defining and manipulating data. E.g., MySQL | Oracle | PostgreSQL | Microsoft SQL Server | etc.
- **NoSQL [not-only-SQL, non-relational]:** dynamic schemas for unstructured data where data can be stored in many ways: column-oriented, document-oriented, graph-based or organized as a Key-Value store. E.g., MongoDB (document) | Cassandra (wide-column) | Redis (key-value) | Neptune (graph) | etc.

What is the difference between Cache and Cookies?

Cookies and cache are two ways to store data on client's machine. They serve different purposes.

- **Cookie:** Used to store information to track different characteristics related to user. Cookie is small piece of information stored on your computer by a website you visit. Every time the user loads the website, the browser sends the cookie back to the server to notify the website of the user's previous activity. Ex - how frequently the user visits, what are the times of visits, what banners have been clicked on, what button clicked, user preferences, items in shopping cart, etc.
- **Cache:** used to make the loading of web pages faster. A web cache (or HTTP cache) is an information technology for the temporary storage (caching) of web documents, such as HTML pages and images, to reduce bandwidth usage, server load, and perceived lag.

What is push notification?

A message that pops up on a mobile device. App publishers can send them at any time; users don't have to be in the app or using their devices to receive them.

What is the difference between Payment gateway and in app purchase?

- **Payment gateway:** a middleman used to transfer transaction data from the merchant to the bank. This software serves as an interface between the merchant's website and an acquiring bank.

- **In-app purchase:** a purchase you make within the mobile app without redirection to the third-party payment page. Instead of you swiping or dipping or tapping your card or device to a physical terminal the app platform, mainly Apple and Google, have your card credentials on file and simply pass them along to their payment gateway for proper routing.

What Are Some Typical Mobile App Errors & Defects?

- **App crashes:** The most popular & annoying defect! Best way to capture is to make alpha, beta release and constantly monitor the crash reports via google analytics or some other analytics.
- **Usability:** The most popular parameter of an App success! Wrong colors combination (background, text or other objects), necessity to type much, flickering objects on the screens, App load time, Screen orientation, text selection, scrolling, navigation, etc. can cause great trouble.
- **Memory & Data Consumption:** OMG! This App takes a chunk of my limited storage and eats up the RAM when functioning. Why is my smartphone so slow when running this App? What! Half of my data plan is consumed by this one nonsense App.
- **Connection Speed:** “This App needs a high bandwidth signal strength to operate”, result – Uninstalled. Make sure that app should be tested on different carriers for ensuring the performance of application as per the expected goal.
- **Data Transmission:** “They haven’t tested it properly. Every time I try sending multiple videos the App hangs”. App errors during data transmission can force Users to uninstall it within hours of downloading it.
- **Screen Size:** Every time I use this App I have to scroll right & left, Top & down. As application can be used on different devices like mobile, tablets, etc., hence it is to be checked, how much a user needs to scroll the page horizontally / vertically for complete view.
- **Battery Usage:** I use this App for half an hour and it drains out the battery. With Smartphones still struggling to tackle the Battery usage problem, this is a hidden important factor for any App’s consideration.
- **Security:** Mobile devices face an array of threats that take advantage of numerous vulnerabilities. Wireless transmissions are not always encrypted. Information such as e-mails sent by a mobile device is usually not encrypted while in transit. In addition, many applications do not encrypt the data they transmit and receive over the network, making it easy for the data to be intercepted.

- **Touch Screen:** Touch screen is one of the elements that should be given due consideration during mobile application testing, especially in Gaming Apps.
- **Interruptions:** The most obvious factor in Mobility world! The mobile app hangs or cannot restore itself when battery strength low or after an incoming SMS, Calls, MMS, etc.
- **Error Messages:** “Don’t know what is the problem, but it just doesn’t function”! Error messages should be well configured for every situation and should be clear & to the point.

Defect Metrics | Severity, Density, Removal, Leakage, Cost and Age

Total Defect Metrics

The first & foremost defect metrics to look at is – “Total defects”. The relative assessment of Total defects vs. Module size & complexity can tell you a lot of things about the effectiveness of testing efforts.

Defects Distribution by Priority & Severity

Total defects give a good idea about the test efforts, but it doesn't reflect on the 'quality' of testing or the build. A more appropriate metrics in this case would be "Defects Distribution by Priority & Severity". Priority and Severity distribution serves as an input to both Business as well as Project team, which helps them gauge the actual "Quality" of the Test team's efforts. Additionally, critical & high defects can be presented as a percentage,

$$(\# \text{ Critical or High Defects} / \text{Total Defects}) * 100$$

Defect Density Metric | Test Design Efficiency

Most of the testing terminologies can be understood by their straight-forward English meaning. Density as in Thickness or Concentration. On a similar line we can also identify the percent concentration of defects within a software build. How?

$$\text{Defect Density} = (\text{No. of Defects identified} / \text{size}) * 100$$

Here “Size” can be considered as the number of requirements or test cases. Hence the Defect Density is calculated as number of defects identified per requirement or test case. Example:

- Total test cases: 1200
- Passed: 900
- Failed: 300 (or Total defects = 300)
- Defect Density = $(300 / 1200) * 100 = 25\%$

Now what does this mean? Simple, 25% of your test cases failed during execution OR 25% of your test design was able to catch relevant defects.

Defect Removal Efficiency (DRE) | Positive outlook

Why do we do Testing? Yeah! To identify defects and improve the Software quality by removing (fixing) them. One of the most important defect metrics, Defect removal efficiency is a measure of Test team's competence to remove (identify) maximum defects before a software is moved to the subsequent stage. Here defects that matter are the ones caught by either the test team or by other users in the next phase.

*Defect Removal Efficiency = #Defects found during testing / (#Defects found during testing + #Defects found during next phase) * 100*

DRE is the percentage of defects that have been removed during an activity. Say the Test team identified 100 defects during system testing. After fixing, retest & regression the build is promoted for User acceptance tests. The UAT team identifies 25 more defects in the build which could have been identified during system testing. What does it mean? The system testing team could not find these 25 defects and they were NOT removed from the system. Hence,

*Defect Removal Efficiency = [100 / (100 + 25)] * 100 = 80%*

Means that the System testing team was able to remove (identify) only 80% of the defects, rest 20% were identified during the subsequent phase. As you might have guessed, DRE can be (or rather should be) computed for each SDLC phase and plotted on a bar graph to show the relative defect removal efficiencies for each phase.

Defect Leakage metrics | Negative outlook

Defect Leakage, as in what percentage of defects actually leaked from the current testing phase to the subsequent phase. As you might have guessed, Defect leakage (defects passed to next phase) is just the reverse of Defect removal (defects identified in current phase) efficiency. Defect leakage is another defect metrics but a deadly one 😊 i.e., defect leakage percentage should be minimal in order to prove test team's worth!

*Defect Leakage = #Defects found during next phase / (#Defects found during testing + #Defects found during next phase) * 100*

Say the Test team identified 100 defects during system testing. After fixing, retest & regression the build is promoted for User acceptance tests. The UAT team identifies 25 more defects in the build which could have been identified during system testing. What does it mean? The system testing team could not find these 25 defects and they were LEAKED to the next phase. Hence,

*Defect Leakage = [25 / (100 + 25)] * 100 = 20%*

Means that 20% of the defects were leaked by the Testing team to the next phase, or in other words test team wasn't able to identify these 20% of the defects. Similar to DRE, Defect Leakage can be (or rather should be) computed for each SDLC phase and plotted on a bar graph to show the relative defect leakage for each phase.

Cost of finding a Defect

How to measure test effectiveness with respect to the money spent? Yeah! The cost of finding a defect!

Cost of finding a defect = Total effort (money) spent on testing / total defects found

Note: Total effort can be calculated by considering total resources, the duration and the billing rate.

Defect Age

Like everything in this universe, defects too have a life-cycle – from birth (new) till death (closed). Defect age is a measure of its oldness, i.e., how much time has elapsed since it was identified and NOT closed.

Defect Age (in Time) = Current Date (or Closed date) – Defect detection date

Note: Defect age can be calculated in hours or days. It is a measure to gauge the responsiveness of the development/testing team. Lesser the age better the responsiveness.

Weighted Defect Density

We already discussed the Defects Distribution by Priority & Severity. Now how to find the defect density with respect to severity or priority? i.e., what was the average defect severity per test case (or requirement) created.

*Weighted Defect Density = [(5*High + 3*Medium + 1*Low Defects) / Size] * 100*

Here “Size” can be considered as the number of requirements or test cases. Weights 5, 3 and 1 are assigned based on the defect severity of High, Medium and Low.

What all to include in a Defect report?

The defect report is the most effective manner of communicating, tracking, and explaining defects to managers and development staff. Contents,

- Defect ID and Status
- Summary
- Description
- Severity & Priority
- Build/Platform/Version
- Steps to reproduce
- Actual results
- Expected results
- Related Requirement ID
- Supporting documents

Important pointers,

- Take note of any configuration, user role, or setup changes that have been made to the QA system - if yes, mention it.
- Sometimes it requires evidence; can be error message text, actual error log, or screenshot/video.

In this way, anybody can quickly get a solid idea of what the defect is and how does it impacts the application.

What Test Design Techniques have you used?

As the name suggests – techniques we as testers use everyday to derive effective test cases from the requirements. The most common ones,

Equivalence Partitioning

Grouping the inputs with the same attributes to partitions. Pick one condition out of each partition, which covers all possible scenarios, to execute test cases. This helps reduce the number of test cases. E.g., testing an input field – all alphabets, all digits, all special chars, combination, blank, null, etc. might become your partitions.

Boundary Value Analysis

Include values at the boundaries. If the input is within the boundary value, it is considered ‘Positive testing.’ If the input is outside - ‘Negative testing.’ The goal is to select test cases to execute boundary values. Baseline: Boundaries are an area in which testing is more likely to yield defects. E.g., testing an input field – blank, full length, null, exceeding length, etc.

Experience-based Exploratory Testing

Domain experts perform testing just by exploring the functionalities of the application without having the knowledge of the requirements. Testers can explore and learn the system while using these techniques. High severity bugs are found very quickly. E.g., based on your experience, you might want to cover different buckets of test cases – positive input [happy flow], alternate flows, negative inputs [error messages], no input/null/etc., browser combinations, etc.

What is the basic difference between Verification and Validation?

- Verification [static]: evaluating the artifacts as well as the software development process in order to ensure that the product being developed will comply with the standards.
"Are we building the Product right?"
- Validation [dynamic]: validating that the developed software product conforms to the specified business requirements.
"Are we building the right Product?"

What is STLC?

Software Testing Life Cycle refers to all the activities performed during testing of a software product.

1. **Requirement Analysis:** requirements are analysed and validated and the scope of testing is defined.
2. **Test planning:** Test plan & strategy is defined, estimation of test effort along with automation strategy and tool selection is done.
3. **Test Design:** Test cases are designed; test data is prepared and automation scripts are implemented.
4. **Test environment setup:** test environment closely simulating the real-world environment is prepared.
5. **Test execution:** The test cases are executed, bugs are reported and retested once resolved.
6. **Test closure and reporting:** A test closure report is prepared to have the final test results summary, learning, and test metrics.

Note: This STLC was more applicable in Waterfall model where each phase output was input to the next phase. With Agile, there is no hard-line segregation of these phases. But still it is good to know about STLC from interview perspective.

#Tip: Creative portfolio

It is tough to shortlist just from resume. Why? Because many people use the keywords in it just for shortlisting – without any real hands-on experience. So, what can genuine experienced people do about it?

Create your portfolio and not just resume. Build a solid LinkedIn profile, network with seniors in the industry, start blogging, maintain a git repository, attend webinars/meetups – whichever way you can showcase your testing prowess apart from the resume. Yes, it does help!

#Tip: Describe Your Current Job Responsibilities

Relevance is key here. Don't tell them everything you did, because not all of it will be significant to this new company. Curate a list of tasks that are the most relevant, and speak about those. This is why preparation is so important!

Prepare important pointers about your current job role like "My current responsibilities include test planning with stakeholders, daily communication & reporting, test design & execution, requirement discussions to ensure test coverage, team management & goal setting, tracking progress, and test automation strategy-design-and-execution."

What is the difference between System testing, Integration testing and E2E Testing?

These are different levels at which testing can be performed, with a bottom-up approach. Start with individual system testing, post that check the integration between different systems. Lastly, check for the end-to-end flow when all systems are connected up-and-running. Example,

Let's take the typical **payments' processing landscape**. Different systems include,

- Front-Office [FO]: Client-facing application from where the user initiates a payment.
- Middle-office [MO]: System which acts as a router to send the payment to appropriate back-office for processing.
- Back-Office [BO]: The actual payment processing happens in the back-office, i.e., including clearing and settlement. There might be different systems for processing, clearing & settlement, reporting, etc.

Taking this as example,

- **System Testing:** Testing of individual systems and sub-systems.
- **Integration Testing:** Testing of integration [data exchange] between FO-MO, MO-BO, BO-FO, and sub-systems as well.
- **E2E Testing:** Considering the complete landscape, testing for E2E payment processing originating from FO and ending at reporting via MO and BO.

Most of the projects have multiple systems that would interact to achieve the end goal. Hence Testing can be done at different levels.

Situational & Practical

Why do you love testing?

It started without an option. As a fresher we are assigned to a Testing team without any preference. We have no option. Yeah! We are ‘pushed’ into Testing at career start.

Once started, it has been a roller coaster ride. It is different from what it seems – a tester is required to excel in different aspects – test process, techniques, planning, automation, mindset, logical understanding, end-user behavior, client interaction, status reports, tools, programming, methodology, etc.

Testing too is a technical activity. There are multiple aspects to it – functional, performance, security, debugging, tools, automation, etc.

Testing is an activity that makes the software/product better.

How will you feel if you bought a new smartphone at a high price only to find a fault when used? Frustrated. Disappointed. Cheated. Angry. Sad? Yeah! A tester’s job is to prevent you feeling all those negative aura 😊

What is the most challenging situation you have had in your Testing experience?

As you might have guessed, it depends on each tester’s experience. The challenge can be technical, process-oriented, people’s related, domain specific OR anything within your Testing experience.

- Testing a technically-challenging application such as social-connected applications, application involving multiple third-party interfaces, cloud-big data-IoT-Analytics-etc. applications.
- Maintaining a balance between design/execution efficiency and defect detection. Improving the Test coverage. Sticking to agile principles.
- Resources on unplanned leaves, some even abscond. Estimation gone wrong, requiring weekend & extended working hours. Rating discussions with unhappy employees.
- A team of freshers with limited domain knowledge. Planning for knowledge transfer sessions.

The thumb rule – be realistic & tell your genuine challenges!

#Tip: Don't appear desperate!

When you interview with the "please, please hire me" approach, you appear desperate and less confident. Reflect the three Cs during the interview: cool, calm, and confident.

#Tip: Don't be cocky

Attitude plays a key role in your interview success. There is a fine balance between confidence, professionalism, and modesty. Even if you're putting on a performance to demonstrate your ability, overconfidence is as bad, if not worse, as being too reserved. All the interview tips in the world won't save you if you come off as unpleasant to work with.

#Tip: Don't be too familiar

The interview is a professional meeting to talk business. This is not about making a new friend. Your level of familiarity should mimic the interviewer's demeanour. It is important to bring energy and enthusiasm to the interview and to ask questions, but do not overstep your place as a candidate looking for a job.

What's your approach to finding defects?

Frankly speaking there is no silver bullet. No one tip, trick or technique. There is no one process or method. No single approach or strategy. Software Testing aimed at finding bugs is a vast discipline in itself. It takes years of experience, just to learn new aspects and get better at it every year.

In order to find defects, we need to cover some basic test buckets,

1. Go one module (or functionality) at a time. Quickly verify the happy flow.
2. Check all the validations.
3. Verify expected errors using no/invalid data, i.e., the Test Data.
4. Make sure you have covered all functional flows (i.e., options like Save, Edit, Submit, Cancel, Open, Close, Refresh, etc.)
5. Cover all possible alternate flows.
6. Check for other out-of-the-box features like concurrency, in-transit data, history/audit, compatibility, etc.
7. A tester's 'attention to details' & 'logical thinking' comes in handy here.
8. At last, do some ad-hoc tests with the intent of finding defects.

The application is currently in production and one module require code changes, i.e., Change request. Is testing the module-under-change enough to ensure quality delivery? Why or Why not?

No. Testing the module-under-change is a must-have whereas the changes might have an impact on the integration with the other modules. Most of the development teams follow modular coding wherein same functions are re-used in different modules of the application. If a change is made to a particular module, it doesn't guarantee that the change is isolated to that module.

That's where 'Regression Testing' comes to rescue. After a thorough impact analysis of the changes, Test team needs to verify all the impacted areas – be it within same module OR outside it (ensuring that there are no side effects of the code changes).

Tip: Test team's understanding and Inputs from the development team are a must to identify all the impacted areas!

How will you start Testing without any functional specification or any related documents?

No wonder there are sometimes these kind of situations, ex. Resource attrition, no-documentation-with-agile-projects, etc.

The only hope in these situations is 'Exploratory Testing'. Since there are no documents to refer, refer the application directly 😊 Explore it, Test it. Gradually the flows make sense and we actually start testing it.

Other option is to sit with Business analysts and developers to get application understanding – listening to experts instead of reading a document.

Tip: Personally, I like to build a document thereafter, so that someone joining after me doesn't have to face this kind of challenge!

What if I give you a team of all Freshers?

Ideally every team should maintain a bell curve. That being said, some think - 'Freshers are lenient, reckless & lack professionalism/experience'. But that is the whole point. Everyone was a fresher once. Freshers are enthusiastic to embark on their professional journey, are not bounded by professional processes, can think out-of-the-box. Just that they need proper guidance, relevant trainings & due diligence.

Though I would prefer a bell curve in my team to maintain the balance, at the same time I would love to mentor freshers on their path to professional success. Who knows I might also learn a few things from the younger lot ☺ As per me, the only challenge would be to convince freshers that "There can be a bright career in Software Testing as well." since most of the freshers (even some experienced ones) believe that 'There is no career in Software Testing'. Once convinced, it is just a matter of training them to come up to speed.

Ever had an argument with your Manager? What was the case & how did you handle it?

The most common: after all the hard work you get a bad rating. Manager states that it's 'relative'. There is no 'weakness' in IT industry, only improvement areas.

- Argument over some Test metrics, be it the status report, defects metrics, productivity, design efficiency. The conflict in understanding.
- Some process doesn't make sense to you?
- Micro-management is demotivating?
- Manager playing favoritism & supporting his aides?
- Too demanding? He/She just needs the work done, even if that means you doing extended hours or working on weekends.
- Task allocation or Project management conflicts?
- Manager: People are too careless for you to be strict?
- People don't understand how this works. It's relative. Even if you are good, only the best of the best gets 1 – rest we have to maintain a pyramid.

It can be anything. The Manager – Employee relation is somewhat tricky. Employee thinks Manager is clever/shrewd, Manager think employee has attitude problems. Guess "communication-to-understand" is the only key.

#Tip | Don't end your job search with the first offer – unless it is your dream job.

Many people get slack in their efforts after the first Offer. But it is not the period to get slack. The prospective HRs might ask – “If you already hold an offer, then why you are looking for other offers now?”

“Good things come to those who wait. But better things come to those who work for it.”

Job change, i.e., joining a new organization, is an important decision. The new organization will be your half-home for coming so many years. So obviously it should be good. Yeah! With first Offer, now you have a good opportunity. But... You have time to search for a still better opportunity. After all, your professional future depends on it. And you don't switch so often. If you are looking for a long-term commitment, why not try your best to achieve the best? Makes sense?

IMP: Make sure to inform previous organization as soon as you get ‘better’ offer, to avoid inconvenience for you & them both.

One of the most common Testing situations – How do you handle timeline crunch?

Since Testing is the last step before client demo, the Test team has to make-up for the delays encountered till the build is deployed in Test environment. Now how do you handle crunch timelines without impacting the quality?

- Risk-based Testing: Prioritize the Test efforts [executing the cases in order of priority] and communicate the same to all the stakeholders.
- Utilize Test automation to the fullest – to cover all regression aspects.
- Consider some buffer in planning phases to accommodate the usual delays.
- The most common: Extended & Weekend working hours.
- Plan & make efficient use of working 9 hours.
- Convey the situation as-is to the stakeholders, asking for some time to complete testing.
- Quicken the defects turn-around time by communicating it to the development counterparts.

Whatever be the approach, always take a retrospective look once the testing is complete in order to avoid delays the next time.

What to do if defect is not reproducible

Now that's a tricky one. Many a times we face these kind of one-off bugs 😊 which peep-out and then hide somewhere. "It was a one-off bug and now not reproducible – so what can I do?" Wrong! Though one-off but still it is present somewhere in the software and as a Tester it is our responsibility to investigate it. How?

- 'Think' of the exact steps and then try to replicate.
- Environment plays a vital role. Think of the environment, what has changed since then.
- Logs! Yes, logs are a great source of narrowing down on the specific module.
- Collaborate with a developer to validate the specific code.
- Test Data can be a root-cause.
- Identifying if there is a pattern to it. Say, after every 3 attempts.

Sometimes these one-off bugs can prove costly in the live environment. So, it is important to analyze carefully.

Note: Invest your time in a one-off bug depending upon its severity. It doesn't make sense to work on a low one-off defect for a complete day ;-)

#Tip | Don't Fret

Frustrated by no-calls? Nah! Don't fret. It is a part of job search. Don't take the rejections, let-downs, and no-responses personally. You are not alone; everybody faces it one time or the other. We only see the success stories of job search but not the timeline & hard-work put in.

No matter the roadblocks and disappointments in your job search, just keep learning & moving forward. CONSISTENT efforts are the key to any success. One day – it will be yours!

#Tip: Don't give up!

If you've had a bad interview for a job that you truly think would be a great fit for you (not just something you want badly), don't give up! Write a note, send an email, or call the interviewer to let him or her know that you think you did a poor job of communicating why you think this job would be a good match. Reiterate what you have to offer the company, and say that you'd like an opportunity to contribute. Whether this strategy will get you a job offer depends on the company and on you. But one thing's for sure: If you don't try, your chances are exactly zero.

The QA team starts testing a software/product and there are “way too many” defects. Every other scenario is failing, new flows are explored & clarifications sought. What would be the strategy now?

It's a complex application with tight schedule. Too many defects add cherry on the cake. The blame game starts. But at the end, development & QA are expected to work as a team & deliver the product/software. What are the options?

- Reject the build. But that would mean delay in delivery.
- Get into a war room, daily. It helps to triage the defects & increase the seriousness.
- Prioritize. Work as a team & ensure priority flows are working as soon as possible.
- Identify super-devs & super-QAs to take up some additional tasks & ensure minimum turn-around.
- Involve the higher management (risk aware) just in case something goes wrong. It shouldn't come as a surprise.
- Since timelines are fixed, make sure turnaround time is as minimum as possible – be it defect fix, retesting, business clarifications, anything. Everybody has to be on their toes.

Note: Retrospective w.r.t. estimations, build quality, management screw-up, etc. is altogether a different discussion.

What to do if Agile User story is delayed?

First things First | Update Product Backlog & Re-prioritize - When a product backlog item has undone work at the end of an agile sprint, it should first technically be put back onto the product backlog. The Product Owner re-orders the backlog. Typically, unfinished items end up at the top, but not always.

Undone means Undone | The Undone Perspective - This model keeps things simple and keeps teams from gaming the numbers. The idea is to put back ‘complete’ story (do not resize the items to represent only the remaining undone work) back on the product backlog for re-prioritization. Do not include any effort spent on the undone user story in the velocity calculation of the current sprint. On average, over a number of sprints the velocity figures will average out in any case.

But something got done | The Done perspective - If we can take some work to done so it can be inspected, do it! Work was done and we want to know how much work remains in the release. Split the User story. There are many cases where the team realizes that a PBI can be decomposed into even smaller chunks of done work. If teams cannot break a story further into work that can be delivered as done and remaining undone user story, then it should go back on the Product Backlog wholesale. The team may choose to re-estimate due to the new knowledge they now have.

#Tip | Don't just wait for a response

Just completed a fantastic interview? And can't seem to get feedback from the recruiter? Oh! This is a common situation now-a-day. The position goes on hold, or they find a better candidate or any other reason. Yeah, it's really unfortunate.

Two ways to get over this: 1. Keep following-up and 2. Keep applying to other opportunities to keep a pipeline of other Interviews - so you don't feel down.

#Tip: Don't trash talk

Don't speak ill about your former company or co-workers, at least not in a direct way. Interviewers get turned off when you opt for the low road.

Why? Because problems are never one-way. If you speak ill about your company or co-workers – interviewers might think that you are also a part of problem.

Do you have working experience in an Agile environment? What's your Team composition?

We do follow Scrum methodology with 2-week sprints including Sprint Planning, Daily Stand-ups, Demo sessions & Sprint reviews. The roles include,

Product Owner: Champion for Business! The communication bridge b/w the team and stakeholders – writes customer-centric user stories, prioritize, maintains Product Backlog, Demo sessions, define releases, communicates team status, etc.

Scrum Team: Champions for sustainable development & QA! Responsible for delivering potentially shippable increments of product at the end of each Sprint. A mix of software engineers, architects, programmers, analysts, QA experts, testers, and UI designers.

Scrum Master: Champion for Scrum! Coach the team, the product owner, and the business on the scrum process and look for ways to fine-tune their practice of it.

Scrum team is the car, product owner - the driver and Scrum Master - the chief mechanic :-)

Do you have any experience with Office friction? When a colleague was not happy with you. Or anything else?

"A person who is always fighting OR the one who never fights – both are not good in the long run."

Disagreement is the key to innovation. Debate is a powerful means to understand different perspective. Friction is important to develop inter-personnel skills.

With a lot of people working together, friction is bound to happen. And it has happened with me as well. It is a part & parcel of workplace.

It is not so much about the friction – but the learning that you get out of it. It can be a debate about which technical approach to take, or as simple as work timings. What if a junior is rebellious? Not adhering to certain policies? Or a senior who doesn't understand your perspective? It can be anything.

We don't learn as much from the success, as we learn from our failures. Friction is good, if you are learning something. Something practical.

Being a QA Lead, how do you handle a non-performing employee?

Do your best in a tough situation,

- Don't ignore the problem. Like a bad sore, if ignored too long - it can lower the morale and productivity of other employees.
- Don't make any assumptions.
- Listen. Communication is the key – talk with the person in private & find out the real problem. There may be external, personal factors influencing their performance.
- Keep it professional. Give clear, behavioral feedback. Coach the employee and lay out the plan together. Set consequences if things don't change.
- Help the problematic employee to get back on track.
- Follow-up. It's going to take time and ongoing help to change habits. Keep track of the performance in the specific feedback areas.
- Encouragement is important. Praise and reward positive change.
- Don't poison the well. Respect confidentiality - Don't trash talk to other employees. Just don't do it.
- If there isn't improvement, take action. Formally act on continued underperformance - work through the company's processes, if necessary.
- Lead by example: When you are open, honest and working, employees are more likely to do the same.

And hope for the best!

Ever worked with a challenging client? Who is adamant & rigid? About the design, requirements, process or timelines. What was the situation & How did you tackle it?

Sometimes we do get that unfortunate project (fortunately there is much to learn from it) where the client is uncompromising – they think they are Google who knows everything OR just that you are a labor working for them.

- Don't panic. Don't react. Instead respond. Think it through what client is demanding & analyze your options at hand.
- Consult your seniors on how to handle the situation.
- Be flexible in accommodating client requests, to whatever extent possible. Two adamant sides won't solve the problem.
- Pull in subject matter experts to talk it through with client, hoping that he/she will at least listen to the respective SMEs.
- Maintain a balance between your team & the client. No one should suffer for the other.
- Report it to the higher management to take care at their level.
- We are no laborers. Clear it with the client or else say No.

Guess there are so many different ways to either make it or break it.

#Tip | Do you have any questions for me?

One of the common ways to conclude any interview.

NEVER go to an Interview & say you don't have any questions when you reach that part. Great questions allow you to probe deeper into their problem, & allow you to steer the conversation to more opportunities to sell yourself as the solution. Make use of it.

Have a question prepared. What are the Product lines? How is the Org structured? How is the work like here? What is the role you are looking for? What is the technology stack you are using? Or any specific Q about recent company development like merger/acquisition/new office/etc. It's one of the best opportunities to showcase your interest in the position.

How do you plan resourcing?

Resourcing, i.e., in simple terms how many people are required for how many days to deliver a product.

The first pre-requisite to resource-planning is **estimation**.

Say the total effort estimation comes to around 150PDs, i.e., 150 Person-days. A single person would take 150 working days to deliver the product.

The second pre-requisite to resource-planning is the **timelines**.

Say the project timelines are for a month, i.e., the product needs to be delivered in a month's time. 20-working days.

Combining both estimations & timelines,

A single person would take 150 days. But we want it delivered in 20 days. The calculation says we need $150/20 = 7.5$ people to deliver.

Adding buffer as risk contingency, we need 8 people to deliver a product estimated at 150 PDs in a month's time.

Tip: It's not that easy though. Multiple other factors come into play – third-parties, systems involved, test automation, timelines, methodology being followed, etc.

What levels of testing have you worked on?

Before answering, first you should be aware of the levels of testing – i.e., unit tests, system testing, integration testing, end-to-end tests, user-acceptance-testing and business validations.

Next, we need to be true about our experience at difference levels. Say,

"System testing makes up the most of my professional experience including Selenium automation & sometimes database validations using SQL. Have also worked on integration tests as part of ABC & XYZ projects which involved web service testing using SOAPUI. Worked on end-to-end testing as part of MNO project which involved validation of end-to-end payment processing. With respect to UAT, we only did Sanity tests in the UAT environment before handing it over to the UAT team."

This gives interviewer an idea about your knowledge of Test levels and your practical experience of different levels-process-and-tools.

What is different between Web and Mobile testing?

Actually, Mobile testing brings in more complexities in terms of device fragmentation, cellular networks, mobile data, hardware complexities, different OS...there are way too many combinations when it comes to Mobile – different companies, hardware, software OS, screen size, networks, bandwidth, geographies, etc. That's where testing on emulators in the cloud helps.

What if,

Battery is drained out? Suddenly there is no (or minimal) network? What about call/message interruptions? Will the App behave similarly for Android & iOS? What if there isn't enough memory left?

Yeah! Guess Mobile Testing is much more challenging than Web.

How do you prove the leadership that Testing was 'completed' diligently?

We do have a 'Requirement Traceability Matrix' to track requirements coverage and Test execution. Additionally, the Daily Reports and the Overall Test Report helps in tracking.

Interviewer: Let me rephrase - Say you have 100 Test cases. The team execute it all & you share the Test Report. But how you as a lead know that 100 test cases were executed diligently and not just for the sake of it?

Being a Lead, I work closely with my team members. I know the type of defects being raised, the clarifications being sought, the challenges being faced – which builds the overall confidence in the quality of testing. I couldn't think of any formal document/process to know about it.

What's the major difference between Agile & Traditional mindset?

As the name says, Agile is more responsive. Traditional approaches were more rigid in accommodating the changes.

Agile team calls for more collaboration and communication instead of documentation & well-defined milestones.

Agility embraces the change. It is more realistic. Traditional mindset is to first freeze the requirements and then accept changes only in later releases.

Instead of top-down management, Agile encourages bottom-up collaboration. It encourages self-organizing teams which are self-driven.

#Tip | ‘Effective Communication’ is one of the differentiators

Every Job description has an implicit requirement of ‘Good written & verbal communication’. Get better at your written & spoken English to communicate effectively.

How does it matter? There is too much competition in the tech space. There are MANY people out there with similar skill-sets. Then how do you differentiate yourself in an interview? By communicating your ideas/experience/knowledge to the interviewer in an effective way. Just one of the many ways to stand apart from the crowd.

As you grow up the ladder, communication skills get equally important.

What can be done for Defect Prevention? I.e., to avoid defects reaching the Test phase itself.

- Analyze the requirements well & good. During refinement, discuss in detail about the possible impact of changes on other dependent modules. Might uncover some ambiguities.
- Build a good Unit Testing process (which is never done diligently. Developers know that it will be tested by QAs anyway).
- Test-driven development (TDD) might help in ensuring that the developed requirement has passed the test.
- Code reviews might help (but are seldom done due to timeline crunch).
- Retrospect about the release and implement the best practices. Learn from the past mistakes, improvise.

What metrics do you capture in your QA project?

Metric, i.e., measurement. Are metrics important in Software Testing? Yes, of course. Without metrics, how do you measure, define, showcase or report your efforts or the product quality?

- Estimating a Testing project, i.e., the Test Estimation.
- Requirement Traceability Matrix (RTM) to measure the requirements coverage.
- ‘Defects’ are at the epicenter of any Testing effort. So, defect metrics makes sense. Example – Total defects, P1/P2/P3 bifurcation and Defect leakage.
- Test Execution metrics (or Daily Execution Status) to measure the progress of QA efforts and the time remaining.
- Test Automation coverage metrics.
- Test Automation ROI [Return-on-Investment]
- Agile velocity and Burndown charts

How to handle multiple third-party dependencies?

Third-party as in – any organization other than you and the client. Multiple third-party interactions are most common in End-to-End testing. Say a payment originates from the front-office (managed by Org-A) >> is routed by Mid-office (Org-B) >> processed by Back-office (Org-C) >> Cleared & Settled (Org-D) >> Reported for Account Statement (Org-A). In between there can be other third-parties as well to fetch the FX rates, or some other inputs.

- Communication & Coordination is the key here. Since multiple Orgs have different work culture and might have different locales – coordination is extremely important.
- Quick & clear communication among all helps in sharing the information/status.
- A daily catch-up meeting with all third-parties is important to gauge the readiness for today & progress till now.
- Any downtime has to be zeroed-in and resolved quickly, for others to progress.
- Having a Command-central is important to resolve any conflicts and avoid deadlocks.
- Integration between different systems should be clearly defined and communicated, to avoid any rework.

What do you do if a defect is caught by end-user in production environment?

Ah! The worst scenario for a Software Tester. There are situations when defects leak into production and are subsequently caught by end-users. Now what?

First & foremost, depending on the priority/severity - it has to be fixed & delivered as per the defined SLAs. For that, we need to replicate it in the test environment in order to identify the root cause.

It is fixed. Fix impact is analysed. Defect is retested thoroughly and a regression is performed on the impacted functionalities. You don't want it to reoccur OR impact any other working functionality. Automation is run to ensure all the happy flows.

And the night goes by.... 😊

Once delivered – now the blame game starts 😐 who missed it & why. Why it was not caught during testing. Once we know the reason – test cases are updated to include different permutations & combinations. The knowledge gap is filled.

What are some of the agile challenges?

Every project is following 'agile' now-a-days, at least this is what they think. Call it Agilish-waterfall OR Pseudo-agile.

Like everything else, Agile also is not a silver-bullet. It also has challenges.

- First & foremost – Implementation challenges. Not everybody understands agile as it was envisioned.
- From QA-perspective, the preference to shorter delivery cycle puts a timeline pressure on the QA team.
- Communication (with Client, Product Owner, Internal, etc.) is not always open & consistent.
- Slowly the focus shifts from 'Quality Product Delivery' to just 'Quick Product Delivery'.
- No process, i.e., anything to everything, in the name of being agile.

#Tip | 'Email Subject' is very important

The e-mail you send to a recruiter is the first step in your interaction. And the recruiter will read your mail depending upon its Subject. General advice is to include "Position + Location + Experience" in the mail subject. For example, "Selenium Automation | Bengaluru | 5 Years" should be good.

And if you have short notice, serving notice, etc., include that too in the subject line itself. Why? Because notice period is one of the major criteria now-a-days!

#Tip: Follow-up

Done with the interview? What next? Follow-up with a personalized thank-you email. Saying that you are thankful for the opportunity to interview with ABC company and would like to know the next course of action.

Would like to understand the techniques you have used while providing estimates. Apart from the simple/medium/complex division of scenarios and assigning them some hours.

Estimation, i.e., approximation - determines how much money, effort, resources, and time it will take to build a specific system or product.

The basic principle is to,

1. Break down the requirements (Use cases/Epics/Stories/etc.) into tasks.
2. Estimate these tasks in light of expertise available, systems involved & dependencies.
3. Sum up.

For Step-2, there are different techniques available. Guess S/M/C with hours (based on prior experience) is the most commonly used. Some use planning poker, Fibonacci, T-shirt sizes, buckets, etc. to categorize the efforts needed – more popular with Agile.

What are the best testing practices?

In simple terms - anything that helps you understand the product & client requirements better OR helps to find defects OR build confidence in what's built.

There is no specific list. Every tester, project or Orgs have their own set of best practices. To name a few,

- **Shift-left Testing:** Test early in the SDLC when cost of fixing is relatively less.
- **Test Automation:** To reduce redundant manual efforts so that it can focus on more effective testing.
- **Collaboration:** Intra-team, inter-team and client collaboration.
- **CI-CD:** To enable quick release cycles.
- **Structured process:** It can be agile scrum, V-model, JIRA lifecycle, release timelines, product roadmap, etc.
- **Retrospectives:** a great way to improve on the process.
- **Exploratory Tests:** allocate some time in order to understand the product and business better.

The list can go on-and-on...

The motto is to keep 'improving'. Anything that helps you test better is a best practice.

When to stop testing?

One answer is – NEVER. Once you are done with your planned test cycle, take some time for exploratory tests and understand the product.

And formal answer would be – when you meet the exit criteria. Timeline and Open issues are one of the major exit criteria to provide a positive sign-off.

Generally, the decision to stop testing for a particular cycle/release is based on multiple factors like timelines, open issues, test execution percentage and coverage, etc.

How do you define 'Software Quality'? what's your version of quality?

A defect-free product! But what if it doesn't meet the requirements?

Conformance to requirements! But what if requirements are too complex for a user to use?

Fit to use and fitness for purpose!

Quality is a moving target based on the targeted market, end-users, competition, evolving technology, etc.

Some common myths & bottlenecks around Testing

- Many think that 'Testing is easy'.
- QAs are inferior to Developers. Hierarchy: Support >> Testing >> Developer.
- QAs are paid less than developers.
- I don't know Automation, I am a 'Manual Tester' | I am an Automation Engineer, I will not work in 'Manual Testing'.
- I am not from CS/IT background - Let's do a Testing course and get a Job.
- Management: Test it and give a sign-off. No-Go is not an option.
- Dev: Don't worry, we can always eat up a weeks' time of QA cycle.
- Management: Reluctant in investing on QA. Build Good-Test Some-and-Deploy.
- QAs: Testing is boring.
- Now-a-days: Test Automation = Selenium automation suite.
- Management: Shift-left | Agile | Automation - are good 'buzzwords', keep repeating them without proper implementation.
- Deadlock: Hiring-Look for 'Test Automation', anybody can do Manual Testing | QAs-Learn 'Test automation' just to clear interviews.
- QAs: Some think 'Test Automation' doesn't work | Others think Manual Testers' job is easy.
- All: Bug missed >> Questions 'only QA' about Why.
- Hiring: No calls for 'Manual-only' testers. Which is then interpreted as 'Manual Testing is dead'.
- Testing = only 'Find defects'.
- Testing will give 100% bug-free product.

#Tip: Freshers looking for Testing job

First, being from a non-CS/IT background doesn't mean you cannot excel in Testing as a career. Computer Science background is NOT mandatory for ANY IT role - be it testing or development. It all depends on your knowledge and skills. I have seen many people from BA/Non-CS backgrounds excel in their dev/test careers.

Second, don't just opt for Testing because you don't want to write code. DO NOT restrict yourself to "manual tester" just because you want to avoid coding. If you are willing to join IT as a BTech Engineer - don't be afraid of coding. Even QA Testing will need coding upto some level. Otherwise, it will become tough to sustain after initial 3 years. It is no rocket science - just need some practice and consistent daily efforts for 1-hour.

Build your profile highlighting the education and then choosing a career path. Let me be frank that the 'Fresher' opportunities in IT are too competitive. So, it is advisable to highlight some unique pointers in your resume that differentiates your profile - training, certification, personal project, code repository, teaching exp, any specific tool hands-on, education excellence, technical hobby, etc. - whatever separates you from the crowd.

What if you find a major defect just before the release date?

That's a tricky situation.

Why you didn't find it earlier if it is a major defect – let's keep that discussion separate 😊

First, no matter what, you need to report the bug.

Second, what you do depends a lot on,

- How likely it is that the bug will occur in production.
- How much damage it will do if it occurs?
- Whether there is a workaround.

If it's a 'show-stopper' - immediately escalate to stakeholders and block the release.

If it's 'not a show-stopper' (limited damage or a workaround is available) - document it as a known issue in the release notes, then notify stakeholders.

Third, as a vendor plan for an immediate fix post-release or the next version.

Note: Sometimes delaying the release is not an option if there are overriding concerns such as contracted delivery dates with severe penalties.

You must have heard about Shift-left Testing. What does it actually mean and why should we shift-left?

A typical SDLC looks like Initiation >> Requirement Definition & Analysis >> Development & Test Design >> Environment Setup >> Test Execution >> Reporting.

As you might have understood, shift-left testing implies that the QA phase be shifted left as much as possible. I.e., testing has to be started from the Initiation & Requirement definition phase itself.

It requires testers to be involved right from the start – providing valuable inputs to enable correct requirement definition in line with customer needs. A sort of Static testing you can say.

As they say – "*The cost of fixing a bug rises exponentially as you progress through SDLC.*"

How important is 'Root Cause Analysis'?

RCA, i.e., a systematic process for identifying "root causes" of problems to find a way to prevent them in future.

In terms of testing, it is mostly employed for defects-RCA. Many defects can be prevented if action-items from an RCA are actioned upon. Some common root cause can be – environmental, configurations, permissions, invalid test data, incorrect test, ambiguous requirements, code issue, etc.

Defects falling under certain categories can be prevented from the start itself, hence saving both time & efforts and focusing more on valuable testing.

In that sense, RCAs help in continuous improvement of the overall process.

Do you ever freeze the scope in agile scrum?

Some say you freeze it at the start of sprint. But then it defeats the purpose of agility.

Others say, you never freeze the scope. Any change requested during the ongoing sprint should first be analyzed and then accepted/rejected based on its priority + how big is the change to accommodate in the current sprint + your sprint goals.

What are your thoughts on mitigating and avoiding production issues?

The 7-step exercise,

1. Defects are inevitable. They will occur at some time. The perception should change from ‘fix the bug’ to ‘fix the root-cause’ driven from top-to-bottom in the hierarchy.
2. Analyze Requirements. The team should be clear on what’s expected with no ambiguity.
3. Practice frequent code refactoring. It helps in improving and optimizing your product apart from clearing the technical debt.
4. Perform aggressive regression testing. Utilize automation and exploration hand-in-hand to perform rigorous regression.
5. Execute defect analysis. Analyze the defects to identify and avoid the root cause in next iteration. Develop automated tests to verify all previous defects as part of the regression.
6. Consider continuous changes, i.e. continuous integration-test-deployment.
7. If possible, integrate error monitoring software. It helps support team in early-analysis of any failures noticed.

“User Acceptance Tests are generally manual”. Is it correct?

Yes, User acceptance tests are generally manual. Why? Because at the end the software/product which you are building is for the end-users and not some machine.

UAT is generally performed by end-users (unless outsourced) who are more familiar with the business flows exercised on a day-to-day basis. They are well aware of the if's and but's of day-to-day operations. Before production deployment, it makes sense for end-users to test the system for the expected functionality and impact on their daily operations.

Though some of the major business flows can be automated, but ideally it's the customer's team who can actually certify the product as meeting the exit criteria and ready for production deployment.

Got an email from Dev Manager saying that customer has found an issue in one of the features. Verified JIRA if there's any ticket raised, but none. Checked for test case covering that scenario, unfortunately the test case is very generic and didn't cover the scenario. Where do you think this bug should have been found?

First, defects will leak – no problem with that. It's good that you want to learn from it. That's the solution 😊

As I see it, the issue lies in Test coverage. As you mentioned there is no defect or test case covering that scenario – which clearly means that nobody tested that particular scenario. It's a coverage miss.

How to avoid that?

- Think twice, write once. Analyze the requirements in detail >> brainstorm on the test coverage >> write test cases.
 - While execution, don't just stick to test cases. Explore the application beyond your test cases. Not every alternate flow can be documented.
 - For tracking, raise ALL the issues and observations of your test execution session.
-

How do you give 'Business training' to your QA team?

You cannot test 'right' unless you 'understand' the business requirement. What does a particular requirement fulfil?

Business training is not a stand-alone activity rather an ongoing one.

- The initial domain-specific training once you join a project.
- Knowledge transfer sessions – within the team, with business analysts/clients, UAT teams, etc.
- UAT test cases can be referred for better understanding of the usage.
- Defects. Yes, defects too are a good source of how the application is supposed to work. Including UAT and production issues.
- Requirement analysis before you start the test design. Clearing your doubts and ambiguous requirements thereafter.
- Client demo's and subsequent feedback sessions.

Generally, if you pay attention to all these aspects – there should not be a need for exclusive business training. But you never know – exclusive trainings can be arranged for some typical projects.

#Tip: Growth mindset

NEVER portray yourself as a person who is NOT interested to learn new things. Every company wants people who have growth mindset. I.e., who are ready to explore and learn new innovative ways to do tasks better.

“What if we give you only manual testing?”, What if you are asked to do performance testing along with other tasks?”, “You have worked on Java, but we use Python. How comfortable are you?” – are some of the questions to check on your growth mindset.

#Tip: How to get an Interview call for a Relocation case?

Many people want to relocate. But Orgs first prefer local candidates since they can come down for a F2F interview. If you desperately want to relocate – few pointers you need to mention in the Email. One, post telephonic round – you can come down for a F2F interview. Second, you are not looking for a relocation allowance. Third – your reason for relocation. Giving some positive pointers to the recruiter might help to get a call.

Do you have Coverage discussion OR a brainstorming session before you start your Test design?

“Two heads are better than One”.

It is important to discuss the coverage for user stories before writing test cases. Often, we see people distributing the user stories and directly writing the test cases. And then rework it based on the review comments. Reviews help, but precaution is better than cure. And we know how seriously reviews are conducted 😊

It's important to discuss the requirements, and subsequent coverage. It helps in avoiding the rework + also enhances overall team's understanding of the business requirements.

Does your team really act on ‘Lessons Learned’? Or is it just a document...

It is important to have lessons learned session after major milestones in the project. It has three aspects,

- Discussion on the challenges faced.
- Brainstorm to find solution to the challenges and what can we do better the next time.
- ACT!

Most of the times first two steps are executed perfectly well but we miss out on the most IMPORTANT step – ACT. Or there is no tracking of the actions taken.

Note: If same challenges/problems/issues are being discussed in subsequent sessions, you really need to focus on 3rd step. It's of no use to just discuss the problems without any resolution.

What has been your biggest challenge from team management perspective?

‘Challenges faced’ is one of the common interview Q. It helps interviewer to gauge your hands-on exp in tackling tough situations.

From team management perse, few challenging times –

Aligning team member goals with organizational goals – one being the push towards test automation. Not everybody is interested OR good at it. So, need to strike a balance!

A team of freshers. Yeah! Once had a team of freshers – full of energy and enthusiasm but obviously inexperienced, required a lot of handholding, training and supervision – but in the end, the project was successfully delivered.

A lenient & reckless team member. Didn't complete the tasks. Irresponsible. No ownership or work ethics. Did multiple one-to-one conversations to understand the problem, laid out a plan of improvement, helped-and-tracked the progress but eventually got to know that he quit IT (was not interested).

Last & obvious – performance appraisals. Being transparent, recurring touchpoints and no-office-politics are some things that help here.

What according to you is the most difficult part of leading a team?

People management is not easy. You constantly need to maintain that balance between individual aspirations and project/org expectations.

Personally, the most challenging part is to keep the individuals motivated – by work satisfaction – a sense that you are growing as a professional with each passing year.

How?

By listening to them - identifying and working on their professional strengths. Want to work on new technologies/tools? Or more participation in stakeholder communication? Some are good at process improvement, while others might want to become a domain SME. Want to master test craftsmanship? Interested in trainings/mentorship? There are numerous possibilities.

It is when you work with them, hear them out and then work together on the strengths – by assigning relevant work – that people feel content & satisfied with their work.

How do you build a Test strategy?

‘Strategy’, as the name suggests – what’s your approach to testing a software/product?

The first step would be to understand the high-level client requirements and the type of software/product [web/mobile/desktop/cloud-hosted/etc.]. and then the next steps follow,

- **Test methodology:** Are you following agile or a waterfall? How is your test cycle organized?
- **Testing Types** – functional, performance, security, database, API testing, etc.
- **Tools:** What tools you would be using. Say, Selenium, Rest Assured, JMeter, etc.
- **Test Management:** How will you manage requirements, cases, defects, traceability, etc. – HPE ALM, JIRA, Inhouse tools, etc.?
- **Team:** What all team roles and expertise are required.
- **Test Environment:** What will be the build deployment process from dev to production.
- **Test Approach:** How will you focus on Unit tests, static analysis, dynamic testing, risk-based approach, etc.
- **Hardware Req:** Will you use any stub/drivers? Or a mobile emulator/test lab? Virtual machines for test automation? Etc.

#Tip: How to make time to Learn.

We do get some queries saying 'I don't get time to prepare.' OR 'I travel 3-4 hours a day' OR 'I have household chores to complete'. Everybody gets the same 24-hours a day. The key is to identify your non-productive time. If you travel a lot – watch YouTube videos when travelling. Cut down on your television – browse on mobile, read all the posts of a blog instead. Not getting time to practice? Wake up an hour early and practice 1-2 programs daily. Instead of the FM in your car, listen to YouTube tutorials on the way. Sleep late? Think about the preparation then. Watch & read before you sleep. And all of this – until you get your next big QA opportunity.

As a Tester, what's the most important achievement in your career?

There has been many – successful deliveries, automation ROI, being involved from project initiation till closure, training & blogs, guiding and mentoring fresh talent, team awards, etc.

But if I must choose one – it would be one of the projects back in 20XX. When I took over as Module Test lead – the project was headed to fail. Environment issues, unclear requirements, low build quality, too-many defects, adamant client, tight timelines, and what not.

The only savior was – people were willing to make it happen. And then it started,

- Daily war-room meeting with Dev – to discuss open & fixed defects.
- Agreement with BAs to reduce the turn-around time of requirement clarifications.
- Process setup to accept build only after proper unit tests.
- Chat window with release team to quickly action on unwanted environment issues.
- Close tracking of defect ageing – to minimize the defect turn-around time.
- More focus on requirement-based exploratory tests to identify maximum defects.
- Involving higher management for client communication w.r.t. estimations, CRs and the progress.
- And of course – weekend hours

At the end, we delivered – successfully. And learned a lot about crisis management.

Any experience with co-located teams?

With more remote working options, IT is slowly adopting the co-located team structure. Yes, do have experience with co-located teams where-in our team was split geographically – Bangalore/Hyderabad/Mumbai/Gurgaon/US.

Important pointers would be,

- The most important - everybody should be clear of their daily-weekly tasks and targets. Daily stand-up helps here.
- A common chat window helps for any real-time issues.
- Recurring one-to-one with team members helps to align personal targets.
- Open & transparent leadership where anybody is open to contact you regarding anything – any time.

How do you maintain traceability with changing requirements?

The main motive of traceability is to ensure that we don't miss testing a requirement. How to ensure that? By tracking that you have executed cases corresponding to each requirement.

In case of changing requirements, we need to update the corresponding test cases OR write new ones and then pull these in your test campaign. For building traceability matrix, any tool like ALM can be used to check on any missing coverage.

The catch: Now that direct requirements are covered, what about the existing impacted functionalities? That's right – the test team need to diligently put together a regression campaign based on the impacted areas.

With updated/new cases + impact analysis + regression + traceability tool – guess we have it covered now.

Say, you have a situation where freshers/juniors come to you about a problem that senior folks are not supporting them in knowledge sharing. How will you go about it?

Though I am yet to come across such a situation, but I feel the key to any team issue like this is – proper communication.

Communication (to listen more) helps you understand different perspectives around the problem and then take corrective actions.

In this case it is important to hear both sides, and then seek help from the senior folks to bring team mates up to speed. To make them understand that mentoring is one of their responsibility at this level. On the other hand, guiding the juniors on how to make the most of this opportunity by learning from other team members.

Recurring knowledge sharing sessions [say once a month] among the team members help here in developing that attitude – it can be technical/process/retrospective/product/etc.

From QA perspective, what's required to achieve a 2-weeks' delivery cycle?

“There is no agile without Test automation”.

- If you are targeting a 2-weeks' delivery cycle, you must shift-left and invest more in test automation.
- Follow the pyramid – Max unit tests > API automation > basic UI automation.
- Repetitive tasks/regression should always be automated – giving time for testers to do some business/exploratory/UI tests.
- Turn-around time should be kept minimum for any inter-team tasks – be it defect resolution, requirement clarification, environment issues, etc.

The goal should be to have a CI-CD pipeline which trigger your tests as soon as the build is deployed – in order to get quick feedback.

#Tip | How to tackle notice period

Long notice period is a BIG BOTTLENECK in getting interview calls. Orgs are always looking for quick joiners. Yeah! We understand it's a really tough situation.

- *Orgs can wait for you, only if you have that expertise or a niche skill set. If you seriously want to switch the job – upskill. Learn & practice new technologies. Keep an eye on Role/JDs to find those niche skill-sets in your experience range.*
- *Other option is to take release from your current project, update notice to a month, and then let recruiters know that currently you are searching for a project and can be relieved early.*
- *LAST & RISKY option is to resign & search. Build a corpus, prepare well before-hand, update notice to 1-month just to attend few interviews & then take this step once you are confident enough to clear interviews.*

Quick tip: Just to check if 3-months' is the real culprit in your case, try updating notice to 1-month & check if you start getting at least the first recruiter calls.

Any experience with localization/globalization testing? How were the cases structured?

- **Localization:** adapting a product's translation to a specific country or region.
- **Globalization:** product design to support any culture or locale (language, territory or code page).

How? By separating the code (logic) from the messages/content/or information.

Many countries have certain laws for products to align with their localization standards. And you need globalization for business expansion.

Once had a federal project with specific regulation around localization.

- Localization: cases were structured corresponding to each application page to verify the content translations in different languages.
- Globalization: cases were structured to run regression tests for each supported language.

What are some aspects (or measures) to deliver a good quality, quickly?

Yes, that's a broad Q – but to start with,

- **Clear requirements:** The most important. Unless you deliver what client expects, anything else won't work. Work with your client/product/business team to understand the requirements clearly.
- **Shift-left:** Test early. Test more at Unit & API level and then moving to system-integration-UI tests.
- **Test Automation:** There is no agile without test automation. Want a quick delivery? Automate your tests.
- **Test coverage:** Don't lose sight of your test coverage mapped to client requirements. Build an effective test design strategy to cover alternate/corner cases as well.
- **Defects:** Yes, defects are a great source of learning – on where to focus more & where to increase the coverage. Don't skip root-cause analysis.
- **Retrospective:** There is no top, only further heights to reach. Retrospective helps to avoid mistakes and highlight areas of improvement.

What would be some corner cases for a mobile app - feature testing?

Yeah, a broad Q but still let's give it a try.

- The most common - Interruptions – incoming call/message, etc.
- Limited memory left/Battery down.
- Varying connection speeds for different operators [3G/4G/5G/Wi-Fi/etc.]
- P&C of App permissions
- Error message configurations
- Monitoring the app crash report.
- User mobility (in case the app uses GPS or other mobile sensors)
- Backups & recovery – app upgrade, battery down, offline usage, etc.

Some straight-forward would be,

- Device fragmentation & OS platforms
- UI/UX Testing
- Functional Testing
- Localization/Internationalization
- Security & Performance

How will you go about resolving a conflict between two people within your team?

This reminds me of the movie 'Chak De' – climax. Where two ppl in your team have a conflict and you as a coach must do something to win the final.

Communication is the key to resolving any conflict. Don't make any assumptions - have a clear discussion with both the team members to understand the real problem – personal/professional/ego/etc.

Once you know the problem - it can be tackled – by proper work allocation and defining individual responsibilities. And reminding them that 'Team player' is one of the important criteria to grow in the professional ladder.

What could be some reasons that nothing happens after clicking a button in the application?

Debugging: identifying a problem, isolating the source of the problem, and then either correcting the problem or determining a way to work around it.

Sounds easy? But a task in itself. As a tester, it is important that you can debug a problem to zero-in on the root cause. For the given problem statement, there can be multiple reasons –

- No binding: just a show-piece button with no corresponding binding to trigger any action.
- Javascript failure: The associated Javascript validation fails silently without throwing any errors.
- HTML/CSS Issue: The button is clickable only at certain dimensions OR not interactable at all.
- Browser-compatibility: Tech-stack used isn't working with older browser versions/specific browser.
- Pre-requisites: Say it's a flash object and it isn't enabled on your machine. Or say Javascript execution is disabled.
- Firewall/Security/Blockers: Some security software or firewall settings are blocking the call-to-action. Some browser add-on you installed might be blocking the click.
- Cache/Browser refresh: Some cache entries disallowing button click. Or a browser refresh loading the page correctly again.
- Code-issue: it can be a code issue as well

After production, will you pick all test cases which you have written before the production for next cycle or only new requirements test case will be considered?

The new feature test cases + regression test cases from previous release.

Regression Testing (No side-effects)

You never know a change in one function (new requirement or defect fix or enhancement or change request) can impact multiple areas of the software. It's our duty as a Test team to ensure everything (apart from the new changes) impacted is working as expected. In other words, to ensure that previous delivered functionality is working even after the new changes. As you might have guessed, knowing the impact (Impact analysis) is a must-have to perform effective regression tests!

Why we need retesting when we already go through regression testing?

Retesting (Recovery Health check-up, post medicine)

This is the simplest to understand. What do you do in testing? Obviously find & log defects. After that? Yeah! Developer will fix the defect. As a Test team you need to verify that the defect fix is working fine, in other words you need to 'retest' the defect based on its steps to reproduce. Simple, right?

Regression Testing (No side-effects)

The code is developed >> Build is deployed >> Sanity is performed >> Full testing is done >> Defects are logged, fixed & retested. What else? Yeah! Truth is stranger than fiction, and so is the Software. You never know a change in one function (defect fix or enhancement or change request) can impact multiple areas of the software. It's our duty as a Test team to ensure everything (apart from the change) impacted is working as expected. In other words, to ensure there are no new defects introduced. As you might have guessed, knowing the impact (Impact analysis) is a must-have to perform effective regression tests!

How I handle team conflicts?

"Managing conflict is one of the biggest challenges a project manager faces."

- Solve the underlying problem: Take the time to thoroughly understand what is causing it. E.g., lack of training, poor communication, unclear expectations or goals, etc.
- Acknowledge the person: Some conflicts occur because a person's ideas and feelings are not being acknowledged as important. By taking the time to acknowledge your team member's problem, you could prevent any ensuing conflict from occurring.
- Call a meeting: Ask each party to present their side. The formal structure of a meeting helps people structure their thoughts. By getting everyone in the same room you have a better chance of coming to a resolution sooner than later.
- Listen: By gathering more information through listening, you'll be better equipped to solve conflicts.
- Understand each team member's viewpoint - listen and understand, to truly get to the bottom of the conflict. Now, it is time to ask the team for a solution. Since, everyone is in agreement that completing the project successfully takes priority, each team member would also be aware that the resolution strategy they are offering, is truly beneficial for everyone involved. Giving the respect and space to share their thought process also adds responsibility and accountability to each team member.
- Exercise authority when required. In high-risk situations, you cannot afford to keep the conflict dragging, hence it is advisable that you wield authority in order to maintain your stand on the proposed solution.

Conflict in the workplace is an ever-present fact. By implementing effective conflict management practices, you can turn your challenges and disagreements into positive resolutions for everyone.

#Tip | I am not getting interview calls.

One of the common plights among job hunters. But if you think through it, you might get some insights.

- *There are seriously a lot of job seekers out there. And your only break-through is your skills & resume. Resume might be missing keywords which recruiters are looking for.*
- *Your current notice period is a big bottleneck, if its 3 or 2-Months.*
- *Many orgs hire via a recruitment consultancy, which don't disclose the company name until your profile is shortlisted. And you are applying to selective companies only.*
- *High CTC expectation. Try not to show your ECTC on the first impression.*
- *Bad luck! Sometimes recruiters get too many profiles to filter through it. Yes, your resume was overlooked!*
- *You don't have 'something' different in your resume. Like rewards, certifications, blogs, social presence, publications, anything.*
- *You are too impatient expecting a call-interview-offer-job in a weeks or some weeks' time and then getting frustrated.*
- *You are not trying it 'everyday'. Yes, every single day!*

Job search in itself is a long-term job ;-) which needs sincerity & continuous improvement. Don't be impatient. Keep your calm, think it through, learn & be a better version in every interview you face.

How I will manage a large complex project end to end?

It depends on the size of the organization and the risks involved. For large organizations with high-risk projects, serious management buy-in is required and a formalized QA process is necessary. For medium-size organizations with lower risk projects, management and organizational buy-in and a slower, step-by-step process is required. Generally, QA processes should be balanced with productivity, in order to keep any bureaucracy from getting out of hand. For smaller groups or projects, an ad-hoc process is more appropriate. A lot depends on team leads and managers, feedback to developers and good communication is essential among customers, managers, developers, test engineers and testers. Regardless of the size of the company, the greatest value for the effort is in managing requirement processes, where the goal is requirements that are clear, complete and testable.

Which all Testing levels have you worked on?

Before answering, first you should be aware of the levels of testing – i.e. unit tests, system testing, integration testing, end-to-end tests, user-acceptance-testing and business validations.

Next, we need to be true about our experience at difference levels. Say,

"System testing makes up the most of my professional experience including Selenium automation & sometimes database validations using SQL. Have also worked on integration tests as part of ABC & XYZ projects which involved web service testing using SOAPUI. Worked on end-to-end testing as part of MNO project which involved validation of end-to-end payment processing. With respect to UAT, we only did Sanity tests in the UAT environment before handing it over to the UAT team."

This gives interviewer an idea about your knowledge of Test levels and your practical experience of different levels-process-and-tools.

Which difficult situation do you come across while handling a team?

Answer to this depends on your personal experience with team handling. Because anything fake will be caught in next counter-Q. If you have actually handled a team, of course there would have been challenges - team member thinking that you play favoritism, micro-management, everyone wanting to do test automation, someone wants to lead a module, team facing technical challenges w.r.t tools to be used, inter-team conflict, etc. etc. The list is endless...

So, you need to pick up a challenge from your personal experience and then discuss on how you solved it.

What if Developers reject your defects and you are able to reproduce it on your machine?

Obvious – Pick up the phone (or walk up to the developer) and discuss.

If it is something related to requirements – clear the understanding with product owner. Product owner holds the authority to call a requirement issue as valid/invalid.

If environmental – reproduce the issue with same environment setup and let know these environment variables to the dev.

Most of the times, developers reject a bug because it is not reproducible on their system. In that case compare the setup – browser, user, environment, exact steps, test data, etc. If it is reproducible on your system – there has to be a catch.

Tip: Checking the logs at failure time helps a lot!

What you will do if you have conflict with your manager on attitude or any other opinion?

The most common: after all the hard work you get a bad rating. Manager states that it's 'relative'. There is no 'weakness' in IT industry, only improvement areas!

- Argument over some Test metrics, be it the status report, defects metrics, productivity, design efficiency. The conflict in understanding.
- Some process didn't make sense to you?
- Micro-management is demotivating?
- Manager playing favoritism & supporting his aides?
- Too demanding? S/he just needs the work done, even if that means you doing extended hours or working on weekends.
- Task allocation or Project management conflicts?
- Manager: People are too careless for you to be strict?
- People don't understand how this works. It's relative. Even if you are good, only the best of the best gets 1 – rest we must maintain a pyramid.

It can be anything. The Manager – Employee relation is somewhat tricky. Employee thinks Manager is clever/shrewd, Manager think employee has attitude problems. I guess "**communication-to-understand**" is the only key.

Will have a 'one-to-one' discussion with my manager to understand his perspective and put across mine. Open communication is the key to solve many of the team problems including this.

#Tip: I am not strong in coding

Many people state that "My coding isn't as strong". It simply means that you have already under-estimated yourself. Because it just might be lack of support in learning, or continuous efforts, practice, lack of motivation, etc. That doesn't mean you are NOT strong in coding :) That's a big under-statement to break the confidence.

Personally, I feel that coding isn't rocket-science. It's just the initial reluctance to move out of comfort zone and learn something new from scratch after 'Years' of experience. But if you think about it - coding is just another tech/learning.

Start learning it through the medium in which you are comfortable – books, blogs, videos, classroom training, one-to-one sessions, etc. Not everybody is comfortable watching random videos or reading blogs. That's where you need to select YOUR medium of learning. Don't hesitate to invest 'some' amount - after all its for your overall long-term career growth.

If yesterday the application was working fine and today it's not working, then what steps you will take to find out the root cause?

- First – check if recently any deployments happened. If yes, what were the changes deployed which might impact the functionality.
- Checking the application logs to get the error and then finding its reason.
- There might be environmental issues such as server load, memory issues, etc. Try restarting the application servers.
- Once these factors are ruled out – check your test data, steps taken, user permissions, etc. Try to identify what parameter changed from yesterday.
- Once you get the steps to reproduce – it is then easy to zero in on the exact root cause by discussing it with the developer.

What will you do when release is near, and you must execute all test cases?

- Adopt risk-based testing, i.e., prioritize the Test efforts based on P1-P2-P3 test cases and communicate the same to all the stakeholders.
- Utilize test automation to the fullest to run regression tests in parallel.
- The most common: Extended & Weekend working hours.
- Plan & make efficient use of working 9 hours.
- Quicken the defects turn-around time by communicating it to the development counterparts.
- Convey the situation as-is to the stakeholders, asking for some time to complete testing.
- Consider some buffer in planning phases to accommodate the usual delays.

Whatever be the approach, always take a retrospective look once the testing is complete in order to avoid delays the next time.

What are the different reasons a service can go down?

There might be different reasons,

- Server downtime
- Bad Test data
- Inconsistent dependent APIs
- Requests overload

What you can do?

- Check status codes that are not HTTP 200 OK to identify API transactions that fail.
- Monitor CRUD operations like POST/PUT/DELETE
- Validate payloads using JSON Schema validation
- Check payload data (using JSON Path or XPath)
- Identify latency by checking API response times

When there is out of memory exception then what would be your next step and how do you find the root cause?

Few Causes:

- Java heap space --- object could not be allocated in the Java heap.
- GC Overhead limit exceeded --- the garbage collector is running all the time and Java program is making very slow progress.
- Requested array size exceeds VM limit --- the application (or APIs used by that application) attempted to allocate an array that is larger than the heap size.
- Metaspace --- Java class metadata is allocated in native memory.
- Request size bytes for reason. Out of swap space: --- when an allocation from the native heap failed and the native heap might be close to exhaustion

Identify Root cause: If cannot find it from the stack traces or logs - Generate a heap dump on OutOfMemoryError exception > Reproduce the problem > Investigate the issue using the heap dump file, it has all information about the memory usage of the application.

How did you handle a difficult situation?

One of the Q to gauge your real-time experience with tough situations. It can be anything,

Way too many defects during a release impacted the test progress and product quality. War-rooms were set up for daily discussion on open issues, defects turn-around time was tracked for quick resolution, unit tests were made more streamlined, effective collaboration between Bas, Dev and QAs, etc.

A team member not performing well. Had a one-to-one discussion to understand the concern. Chalked out a plan for recovery with proper milestone tracking. If it didn't work, gave warnings and checked with HR for the company policy on PIP.

Crunch timelines for one of the releases. Utilized test automation to the fullest to cover regression tests. Prioritized the test efforts based on P1-P2-P3 test cases. Extended hours and some weekend. But yes delivered it on time.

You need to think of a tough situation you were in and what steps were taken to tackle it. It can be anything – technical, process, team, etc.

How to deliver quality product when there is less time?

Since Testing is the last step before client demo, the Test team has to make-up for the delays encountered till the build is deployed in Test environment. Now how do you handle crunch timelines without impacting the quality?

- Consider some buffer in planning phases to accommodate the usual delays.
- Prioritize the Test efforts (risk-based testing) and communicate the same to all the stakeholders. I.e., execute the Prio-1 cases first followed by P2 and P3.
- The next important task to do is to utilize test automation effectively. Test Automation should cover your regression quickly.
- The most common: Extended & Weekend working hours.
- Highlight the delays to buy some extra time.
- Plan & make efficient use of working 9 hours.
- Quicken the defects turn-around time by communicating it to the development counterparts.
- Convey the situation as-is to the stakeholders, asking for some time to complete testing.

Whatever be the approach, always take a retrospective look once the testing is complete in order to avoid delays the next time.

What makes You a Good Tester?

A good Tester is built on not just one, but many important traits.

Curiosity. Or in other words - an appetite to learn. Always curious about anything new (or different).

Technical. A good Tester is technically strong. Technically doesn't mean only programming - he/she understands the Tech stack.

Observation. A keen observation to identify the anomaly ;-)

All these are important, but - "**Being Logical**" has the utmost importance. I have never seen an 'irrational' good Tester :) We as Testers have to understand the business, technology, product & process. Being logical helps in every area :)

Why Manager is expected to have hands-on automation experience when all s(he) has to do is managing?

Industry has moved to automation-centric interviews & hiring, but is it really required at Manager+ Level?

A manager should know about what s(he) is managing. One of the biggest bottlenecks to automation success is its mis-understanding at higher levels. When you have hands-on experience, you know -

- Automation takes time, it's not auto-magic.
- How to set realistic targets w.r.t. automation.
- Invest in automation from a long-term perspective.
- The challenges faced with automation.
- And, when the team member is bluffing about some automation aspect 😊

Overall, if a manager has hands-on automation experience – it's really good.

QA is NOT just to give the Sign-off!

"We have a release on XYZ date. Please provide the sign-off by ABC date."

A single line (either written or oral) from the higher management (often delivery managers) defeats the overall purpose of software testing.

- We have a release on XYZ date --- it has already put a pressure on the QA/Testing team about the time-box available for test activities.
- Please provide the sign-off --- it sounds like people just want the sign-off to go live instead of a test report providing information about the build quality. It is taken for granted that we need a sign-off.
- By ABC date: What about the quality of testing? Apart from the QA team itself, most of them are not bothered about it.

Projects with this attitude are bound to fail in the long-run. Projects who don't pay attention or don't give due importance to the QA/Testing activities are bound to fail in the long-run.

What's your approach to Testing – Execution or Defect Identification?

Test Design & Execution: Understand the requirements >> Write all necessary Test cases >> Execute the Test cases >> Mark them either Pass or Fail >> If failed - raise a defect >> Report.

Defect Identification: Understand the requirements >> Write all necessary Test cases >> Execute the Test cases and explore the application with the aim to find defects >> Raise defects >> Report.

At first glance, both looks the same. But there is a slight difference between priorities.

What are some of the common Software Testing myths?

- The first & foremost – Software Testing require less technical know-how. E.g. A Software Tester doesn't need to know programming OR how to code.
- Software Testing is inferior to Software Development. OR Software Tester salaries are less when compared to developers.

- Software Testing is boring. Don't know why.
- Manual Testing is Dead. But it is not about Manual Testing – instead about the Tester skills. Both manual & automation skills are required going-forward.

How do you bring order to exploratory testing?

By definition, exploratory means experimental, probing or investigative. People usually treat their ad-hoc tests as exploration. To some degree – yes. But exploratory tests are much more than that...

- **Session-based:** Uninterrupted block of test time with a particular mission. Charters are used to define the scope for exploration.
- **Scenario-based:** Parallel scenario identification as you explore the application.

Can automation QA do without QA concepts?

As far as automation is concerned – s/he can manage with his/her programming skills. Agile anyways applies to all irrespective of your skillset.

But somewhere I feel it is important to know about the QA concepts. Testing is not just about requirement-cases-execution-defects. Your domain knowledge and business understanding play an important role. Additionally, your attitude towards 'quality' is of utmost importance.

For many people Agile means following Scrum.

Agile approach stands for evolving requirements and solutions through collaboration. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.

Any development method following this approach is 'agile' – and Scrum is just one of it.

Some other agile methods that I have heard of – Extreme programming, Kanban, Lean Software development, Scrumban, etc.

And then there are some development practices which support these agile methods – Daily scrum, TDD, BDD, Continuous Integration, Timeboxing, etc.

How will you handle if QA team finds 2 major defects just before the release date?

Unfortunate scenario. Why? Because critical/major defects should always be caught earlier in the test phase. But you never know with multiple deployments...

Now that the situation has come up,

- Document the defect in detail – environment, configs, build version, test data, user setup, steps, expected and actual results, any logs, screenshots, etc.
- The next step would be to identify if any workaround is available for that functional flow, or is it a blocker.
- Then it's time to inform the internal stakeholders – scrum master, agile team, managers, business analyst and product owner. As a QA, it is our responsibility to inform stakeholders about the build quality.

Whether to go ahead with the release OR delay it is entirely management decision, but our inputs will play a critical role in decision-making.

- Say there is a workaround available, then we can go-ahead with the release and deliver a hot-fix in the next deployment window.
- If there is no workaround, the impacted functionality/customers [business impact] need to be assessed. Can we take that risk? If Yes, go ahead. Else – delay the release.

The Go/No-Go decision will be taken after discussion between internal POCs and the Client.

Note: Such situation happens. It's good that at least issues were caught before go-live. But it does need a retrospective as to when it was introduced in the system and if it could have been caught earlier.

Ever done a mistake? What was it and what did you learn out of it?

The idea is to gauge your observant and learning attitude.

Everybody makes mistakes. If you can re-collect any one from your personal experience – that will be the best.

Example – Once there was a project where I neglected the important of regression testing. Just executed the cases without any exploration. Later on, business identified a major regression defect in the same module. It came back to me in the root-cause analysis meeting. That day, I accepted that regression testing is very important before going live/demo.

Do you follow Agile? What's the Sprint duration and different activities?

Yes, we do follow agile with a 2-week sprint model. Agile squads [Dev + QA + BAs + DBAs + Scrum Master + etc.] are distributed according to the products/modules/projects. We have daily scrum in the morning (or evening) to discuss on daily activities. Before the start of sprint, we have sprint planning meeting to discuss which user stories should be picked up for current sprint. Then at the sprint end we have sprint demo to the client/product owner for the done features.

Regarding retrospective, we usually don't have it after every sprint but every alternate sprint, i.e., monthly. Also, we have a reoccurring backlog grooming session as well with the Product Owner as & when it is scheduled.

What will be your top 5 to 10 scenarios to test ABC application?

One of the Q frequently asked in any interview. Either they will open the application like e-commerce/travel/etc. OR just ask you to give scenarios verbally.

Before you start giving the test scenarios, take a pause and think about the application and its modules/functionalities.

- Homepage rendering on different browsers – covering both web and mobile.
- Does it require registration? Sign-Up scenario.
- Most of the applications need a login, verify login with different role users [authorization + authentication] – sign-in and sign-out.
- Testing of 'Search' within the application, i.e., search results with different test data combinations.
- Test to check any broken links, i.e., different categories listed, header & footer notes, etc.
- Any integration with outside-apps like OTP/Email verification/etc.

Apart from these, interviewers are impressed if you cover other aspects as well apart from just functional cases,

- UI/UX design, i.e., easy-of-navigation, colour-scheme, image-rendering, scroll bars, font-size, neat-and-clean vs. cluttered, etc.
- Localization and Globalization testing, i.e., if the application is to be used in different geographies/languages.
- Page rendering performance measurement
- Cross-browser testing
- Security aspects like cookies, certificate, secure connection, etc.
- Tests on handheld devices, say a smartphone.

You are managing a team where there are people more experienced than you. How would you handle it?

It's all about defining the roles and responsibilities. With experience they must be having a lot of knowledge about domain-tech-products-process-etc. The task would be to identify how well can we leverage their experience and to balance it with their aspirations.

Agree there might be few conflicts but then which team doesn't have it. It is all about being transparent and having a clear discussion about the roles, responsibilities, problems and solutions. It is the lack of communication that poses a greater problem.

And for the unmanageable, one-off case – will have to play the authority card, i.e., look I am leading this team and these are your roles/responsibilities and goals.

A new member joins your team. How will you go about training them?

The training should focus on - what is it that a new team member needs to start contributing to a project?

- Organization structure, i.e., where does this project fits in the overall organizational structure.
- Project overview covering different applications and their role.
- Knowledge about the technology stack being used, i.e., test management tool, test automation tools, defect management tool, internal tools, etc.
- Individual application(s) sessions with hands-on practise, say let's start with sanity tests and then functional tests.
- And one very important aspect – domain training. Even more important for a tester.

We can plan for all these items and prepare a new joiner kit to be used in any subsequent future hiring. Once done, we are good to go to utilize the his/her expertise.

What if we give you performance testing? Or Security testing? i.e., some new work which you haven't worked on yet.

Awesome! That would be great. Just that hope you give me the time to prepare and train.

I see the need for full-stack QAs who can do functional testing, test automation, performance engineering, etc. Looking at the trend it would be a great opportunity for me to up-skill and grow in the career graph.

Iterating again, the only concern I have is about the timeline. I am no expert in these areas, so would need some time before I can start contributing. The expectation shouldn't be from Day-1. Rest I am open to new learning and aligning with project work.

What were some of the enhancements that you brought in? It can be technical, process or management perspective.

One of the popular interview Qs – helps to gauge your innovative mindset.

Start with any technical change that you might have brought,

- Introducing any **new tech/tool** – cucumber/code analysis/performance/Jenkins/etc.
- **Upgrading** the existing tech stack – Junit to TestNG | HTTPClient to Rest-Assured | Selenium upgrade | Java upgrade | CI-CD implementation | etc.
- Any **process** changes – shift-left testing, moving from UI to API testing, collaboration with Dev/BAs, setting the review process, aligning with the agile process, root-cause analysis sessions, domain learning sessions, etc.

- **Trainings** – any trainings you might have delivered OR started the process – technical/domain/process.
- **Management** – one-to-one discussions, team retrospectives, fun events, team re-organization, etc.

Think it through and the best would be to keep handy few enhancements that you might have brought in the project. That way it will be easy for you to answer!

How do you do Goal setting? How to measure goals?

Goal setting is an important activity. According to me, goals act as a bridge between organizational growth & vision and individual's aspiration & career growth.

The best way is to align individual and team's goal with company's vision. **How?**

1. Understand the Company/Business Unit goals – technology adoption, process changes, domain, re-organization, etc.
2. Define how your team can contribute to the above BU goals – adopting latest technology, driving sales, product development, test automation, domain expertise, etc.
3. Once you have the team goals, rest is just the mapping of these goals with individual's skills & strength.
4. Important point to note – Communication is very important. Have a discussion with each team member on what are the BU/Team goals for this year and how team member's individual skills can help achieve it + discussion on any personal career aspirations in line with BU goals.

How to measure?

Generally, it's a good practise to revisit goals progress on a quarterly basis [if not monthly]. If you have put definitive goals with clear criteria – you can easily track the progress, quarterly.

Any challenging conversation you have had?

Oh Yes, everybody has their share of challenging times 😊 and related conversations,

- Conversation with the manager regarding my aspirations and what I need to do to achieve those. And then the appraisal discussions.
- A team member not performing up-to-the-mark. Had a one-to-one conversation with him/her to understand what's the real issue – personal/professional – and then chalked out a plan for improvement.
- Interacting with a tough client who always used to bargain on the estimated efforts and then we need to provide a detailed analysis.

- Conversation with all internal stakeholders on a delivery miss or a production defect. The subsequent root-cause analysis and retrospective measures.
- Convincing the product owner/scrum master on why we cannot include a particular user story now in the running sprint. Or why this particular test approach is not feasible.
- Communicating that test automation needs proper planning, time & resources, maintenance, regular execution and tracking in order to be successful in the long-run.
- Not to forget the daily tough situations we get in – onshore-offshore miscommunication, extended work-hours, tight timelines, framework discussions, etc.

What was the conclusion? Conclusion is always a win-win situation, a middle ground where both the parties understand each other's point and agree on a plan that is beneficial to both!

Note: You just need to re-collect a few of your challenging times and the conversations you had around it.

Why is it that UAT team find defects most of the time even when you have completed System Testing & signed-off?

There can be multiple reasons to it,

- **Business outlook:** UAT team is closer to customer experience and how the system will actually be used by end-users. Though System test team should also derive cases based on real-world usage but still there are few gaps left which UAT team discovers.
- **Test Data:** One of the important factors for a successful test. Often UAT testers have access to production-like data which uncovers defects that are missed by System Testing team.
- **Timing:** The time of testing also plays an important role. Unit tests will focus on code-level checks, system tests focus on the modules, integration & functionalities. Once the system is deployed for UAT, it is already free from basic defects and the focus is to test end-to-end business flow.

And then there can always be regression defects introduced in the system in later phases of testing which are then caught in UAT.

#Tip | If you are an ‘Immediate Joiner’

Are you already serving notice period? Or an immediate joiner? Whoa! Companies are hunting for such candidates because chances are high you will join their firm & not ditch. So why not flaunt your notice?

Short notice period or immediate joiner is now-a-days one of the major criteria in shortlisting candidates. Flaunt it on your profile, job portals, and as part of the subject line in every mail you send to a recruiter.

Companies are ready to trade few skills for an immediate joiner. After all it takes a lot of money, time & resources to find a candidate and nobody wants the last-minute surprise.

Ever missed a deadline? How did you handle it?

Once in a while there are projects that miss the deadlines. There can be multiple reasons ranging from resourcing, technical, infrastructure, process, etc.

What's important is how did you handle a deadline miss.

The first step is to **identify the bottleneck**, i.e., identify the reason for delay. Say, once there was a project with frequently changing timelines along with too-many defects.

Next step is to **resolve the issue**. Say, building a requirement process to streamline changes and to focus more on build quality by implementing rigorous unit & API tests.

In parallel, the most important aspect is **communication**. Delay shouldn't come as a surprise (or shock) to stakeholders at last minute. Keep all the stakeholders informed if you foresee that the timelines are slipping and might be impacted. Keeping stakeholders informed avoid any escalations later on.

Last, it's time for retrospective to avoid timeline miss in future deliveries, i.e., to learn and correct the mistakes.

Ever helped a teammate, technically? What was it?

Important Q to gauge your team work. Don't stop at just one...

Yes, as a team it all works on give-and-take. Sometimes I help others in the team and vice-versa.

The first that I can think of is **training** all the new joiners with respect to project applications, tools & technologies used, etc.

Next there have been many incidents when people get stuck somewhere in **test automation**, be it scripting OR framework changes, etc. Few times I might have helped like suggesting workarounds to solve the problem [say click, actions class, robot class, javascript, etc.]. Other times it is not just one-sided help, we usually brainstorm to get to a conclusion.

Functionally, there have been many incidents where I took a lead in explaining the functionalities to team members.

Apart from that, I won't say help but **enhancements**. Whenever someone is stuck or comes with a problem – we try to solve it collectively – say rerun failed cases automatically, use rest services instead of UI wherever possible for test data setup, ad-hoc tests coverage, etc.

Think it through, and everybody have helped someone some time – you just need to re-collect!

What will you do if your team member misses a bug into production?

Unfortunate! But defects will leak. "Absence of defects is a fallacy".

Now that it has already happened, I would rather concentrate on next important pointers,

- What is the **issue and its impact**? i.e., priority and severity. What's the SLA for production fix - how to accommodate it – what should be the test coverage – etc.
- **Retrospective**: Why it was missed? Or at what stage it should have been caught? Identifying the root cause is the first step to rectifying it.
- **Corrective measures**: How can we increase the test coverage to ensure such defects are not missed again. Can we a production coverage list? Or alternate flows checklist? A process changes, say review? Etc.
- **Meeting**: It is unfortunate to get production issues. People shouldn't take it lightly. A team meeting might help here to convey the seriousness if an issue occurs in production – it impacts the reputation!

One should NOT pin-point someone but assess the situation and take corrective measures. That's how you improve as a team.

How do you deal with a 'bad feedback' from the Client?

Feedback is a way to improvement, just that how you take it constructively.

First, don't take it personally. Think from the client's perspective and you might understand why such feedback.

Take down the pointers [from feedback] to work upon. It might be about the build quality or a timeline miss or budget overflow etc.

Build a plan to tackle all the above pointers.

Note: Sometimes it just might be a misunderstanding which needs to be sorted out via proper communication/discussion between stakeholders.

Ups-and-downs are part-and-parcel of a project. What's important is how you handle it to either make it or break it. And Yes, always keep the internal stakeholders aware of what's happening with the client side.

#Tip: Introduce Yourself

The very first ask to kick-start the interview. The best way to introduce yourself in an interview is to start with your educational background >> Projects worked on (Domains) >> Testing types/levels involved in (diversity or stability) >> tools used so far (technical) >> and current roles & responsibilities. This way you can cover the overall experience w.r.t. technology, domains, tools and roles.

E.g., “I am a B. Tech engineer from ABC university. Started my career with ABC company in 2010, moved to XYZ company in 2015. I have worked on Banking and Capital Market domains involving system testing, end-to-end testing, UAT, and test automation. Have hands-on experience in Selenium test automation with Java including TestNG, Maven, Git, Jenkins, AWS, etc. Currently I am playing a role of Test Engineer as part of agile squad responsible for test design & execution, team management and test automation.”

If you are leading a team, how do you know which team member is better? What are some of your parameters?

A tough one indeed. Since there are multiple parameters to relatively assess the team members. Few pointers,

First & foremost – **Testing skills**. When I say skills, it means the thought process a person has when testing an application.

- Some testers are able to identify tough defects by exploring different alternate flows while others stick to just test cases.
- Few understand the requirements from client's perspective and ask Qs which are logical.
- Some collaborate with development-business-DB-etc. teams very well to understand how the system works.

Independent: The next important parameter is independence, i.e., if the team member is able to take complete responsibility of the assigned task and understands the accountability. You need not follow up again & again. He/She delivers without question.

Team Work: How well he/she is within the team. A helping team member will always take the lime-light.

Pro-active: Few members wait for the tasks while others are proactive to ask about it or find ways to do it better.

Big Picture: Not everybody understands the bigger picture of business landscape-technology-applications-internal hierarchy-etc.

Stakeholder communication: Networking is an art not everybody is good at. There are only a few who gel up with most of the people. It is NOT important to know many people, what's more important is that how many people know about you!

Note: These is not an exhaustive list. All these traits work hand-in-hand to build a professional that climbs up the ladder faster than the others.

If I ask your manager, what would he/she say is your biggest strength and biggest improvement area?

Everybody has their own strengths and improvement areas. Just make sure it is something that won't impact your candidature. Improvement area [or weakness] should be like it can be turned in your favour 😊

Strengths [easy part],

- A team-player who is always ready to help.
- Quick-learner, understands quickly and won't ask it again.
- Responsible, give me a task and you need not follow it up.
- Technical, I am a tech-person who loves to explore new tools & technologies.

Improvement areas/weakness [tough part],

- Learn to say No, I am usually flexible enough to accommodate any requests which sometimes backfires in the sense that I have to stretch.
- Big-picture, my manager always used to tell me – “You are good but to be at the next level - understand the big picture”.
- Networking, expand your presence among development-business-client teams.
- Personally, I feel I have missed few chances because I am non-political. But that is how I am!

What is your career inclination? Managerial track or the Technical track? And why...

One of the important Q seeing the change in QA landscape with Agile.

- a. I am a people's person. I like to work with a team, manage their aspirations in line with company goals, handle project deliveries, business communication and reporting. I feel my communication and people management skills are good enough to pursue the managerial track.
- b. I am a techie. I love to explore new tools and technologies. I learn by trying different technical approaches. Also, with the advent of agile – QA landscape is quickly adopting the technology-first approach. I feel my skills are more aligned to pursue the technical track.
- c. Both! My dream role would be to engage people for project deliveries and also, I don't want to leave the technical track. Would love to work in a role where have to manage a team and also get some hands-on exposure to technology as well.

The choice is completely personal in line with your career goals and aspirations!

Database & SQL Qs

Different clauses used in SQL

An SQL clause is defined to limit the queried results to certain specified conditions.

- **GROUP BY:** used in aggregation to arrange identical data into groups, the GROUP BY clause follows the WHERE clause in a SELECT statement and is followed by the ORDER BY clause
- **HAVING:** used to specify a search condition in a GROUP BY clause, HAVING can be used in the absence of a GROUP BY clause by using a WHERE clause
- **ORDER BY:** sorts the result set in ascending (default) or descending (using DESC keyword) order
- **WHERE:** used to define the condition of the records to be extracted

SQL Constraints

Constraints are statements used to establish the rules for all records in the table. If any action violates a constraint, that action will be aborted. Constraints are defined while creating the database itself with CREATE TABLE statement, or after the table is created, by using ALTER TABLE statement.

- **NOT NULL:** indicates that the column is required to have some value; it cannot be left null.
- **UNIQUE:** ensures that each row and column have unique value; no value is being repeated in any other row or column.
- **PRIMARY KEY:** used in association with NOT NULL and UNIQUE constraints to identify a particular unique record.
- **FOREIGN KEY:** used to ensure the referential integrity of data in the table and also matches the value in one table with another using PRIMARY KEY.
- **CHECK:** used to ensure the value in columns obeys specified conditions.

Aggregate functions in SQL

SQL Aggregate functions determine and calculate values from multiple columns in a table and return a single value.

- **AVG():** Returns the average value from specified columns.
- **COUNT():** Returns number of table rows.
- **MAX():** Returns the largest value among the records.
- **MIN():** Returns smallest value among the records.
- **SUM():** Returns the sum of specified column values.
- **FIRST():** Returns the first value.
- **LAST():** Returns last value.

Any idea about Triggers?

Triggers in SQL is kind of stored procedures used to create a response to a specific action performed on the table such as INSERT, UPDATE or DELETE. You can invoke triggers explicitly on the table in the database.

Action and Event are two main components of SQL triggers. When certain actions are performed, the event occurs in response to that action.

What is SQL Injection?

SQL Injection is a type of database attack technique where malicious SQL statements are inserted into an entry field of database in a way that once it is executed, the database is exposed to an attacker for the attack. This technique is usually used for attacking data-driven applications to have access to sensitive data and perform administrative tasks on databases.

DELETE, DROP and TRUNCATE

- DELETE removes some or all rows from a table based on the condition. It can be rolled back.
- TRUNCATE removes ALL rows from a table by de-allocating the memory pages. The operation cannot be rolled back
- We can use the DELETE command with WHERE clause but cannot use the TRUNCATE command with it.
- DROP command removes a table from the database completely. Cannot be rolled-back.

Query - Details of a student from Students table whose name start with K?

```
SELECT * FROM Student WHERE StudentName like 'K%';
```

What do you mean by Stored Procedures?

A stored procedure is a collection of SQL statements that can be used as a function to access the database. We can create these stored procedures earlier before using it and can execute them wherever required by applying some conditional logic to it. Stored procedures are also used to reduce network traffic and improve performance.

Query - Select random rows from a table

Using a SAMPLE clause we can select random rows,

```
SELECT * FROM TableName SAMPLE(10);
```

#Tip | Job Search is a long-term commitment

Thought finding a new job was easy? Nah!

You need to understand that job search is a long-term commitment (though you might be lucky). Job search demands daily due diligence for a period of time, without being bogged down by the no-calls & rejections. The motivation & momentum should not die.

How? Keep going through interview experiences of fellow testers and prepare the common Qs asked. Engage in discussions and keep learning. Many people get de-motivated by no-calls and then drop the idea of job search. But think from a future perspective. Daily 15-30 mins to build your career is the least you can do.

There are in numerous companies & opportunities out there – but it takes time to hit the jackpot. Till then, keep trying!

Query - Return each name only once from a table

Need to use the DISTINCT keyword,

```
SELECT DISTINCT Name FROM tableName;
```

Order of SQL SELECT

Order of SQL SELECT clauses is: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY.

Only the SELECT and FROM clauses are mandatory.

Any idea about UNION, MINUS, UNION ALL, INTERSECT

- **MINUS:** returns all distinct rows selected by the first query but not by the second.
- **UNION:** returns all distinct rows selected by either query
- **UNION ALL:** returns all rows selected by either query, including all duplicates.
- **INTERSECT:** returns all distinct rows selected by both queries.

Query - To get the second largest value from a given column of a table

```
SELECT MAX(COLUMN_NAME) FROM TABLE_NAME  
WHERE COLUMN_NAME < (SELECT MAX(COLUMN_NAME) FROM TABLE_NAME);
```

```
SELECT Max(Marks) from Students  
WHERE Marks < (SELECT Max(Marks) from students);
```

Popular Database Management Systems

Oracle | MySQL | Microsoft SQL Server | PostgreSQL | Sybase | MongoDB | DB2 | Microsoft Access | etc.

Difference between NULL value, Zero, and Blank space

- Null value is a field with no value which is different from zero value and blank space.
- Zero is a number
- Blank space is the value we provide. The ASCII value of space is CHAR(32).

Test for NULL Values

A field with a NULL value is a field with no value. NULL value cannot be compared with other NULL values. Hence, it is not possible to test for NULL values with comparison operators, such as =, <, or <>. For this, we have to use the IS NULL and IS NOT NULL operators.

- `SELECT column_names FROM table_name WHERE column_name IS NULL;`
- `SELECT column_names FROM table_name WHERE column_name IS NOT NULL;`

SQL - Write a query to fetch the 3rd highest Salary from the Employee table.

Query-1. Select Name, Max (Salary) As Salary from Employee Where Salary < (Select Max (Salary) from Employee Where Salary < (Select Max (Salary) from Employee));

Query-2. There should be just 2 Salaries greater than it.

Select * from Employee E1 where 2 = (Select Count(Distinct(E2.Salary)) from Employee E2 where E2.Salary > E1.Salary)

Query-3. (Oracle) Sort descending >> Then pick row number 3.

Select * from (Select Name, Salary, ROW_NUMBER() OVER (Order by Salary Desc) As RowNum from Employee) as Temp where Temp.RowNum = 3;

Query-4. (SQL Server). Top 3 by sorting in descending order >> Top 1 after sorting in ascending order, i.e. 3rd highest.

Select Top 1 Salary from (Select Distinct Top 3 Salary from Employee Order By Salary Desc) As Temp Order By Salary Asc

Query-5. (MySQL). Sort descending >> Skip top 2 & then fetch next record, i.e. 3rd highest salary.

Select Salary from Employee Order By Salary Desc LIMIT 2, 1;

Difference between Having and Where clause

Where clause is used to fetch data from a database that specifies particular criteria.

Having clause is used along with 'GROUP BY' to fetch data that meets particular criteria specified by the Aggregate functions.

Where clause cannot be used with Aggregate functions, but the Having clause can.

Query - Change a value of the field 'Salary' as 7500 for an Employee_Name 'John' in a table Employee_Details

UPDATE Employee_Details set Salary = 7500 where Employee_Name = 'John';

Query - Select all the even number records from a table

Select * from table where id % 2 = 0

Query - Fetch alternate records from a table

Records can be fetched for both Odd and Even row numbers using mod operation.

Select studentId from (Select rowno, studentId from student where mod(rowno,2)=0);

#Tip | Job search is the best time to learn & upskill

Almost everyone gets into the comfort zone of daily work, and learning takes a backseat. And when you start job search, you get to know about the industry trends. Job Search is the period when you align your skills to the industry.

Don't just keep sending the same resume to the recruiters for 6-Months. And then crib about no-calls. Don't just send-and-wait, improve your learning & skills in the meantime. How? Job descriptions and interviews you face are great source of pointers to learn. It will surely help you in the long-run!

Query - To display current date

There is a built in function in SQL called GetDate() which is used to return current timestamp.

```
Select getdate();
```

What is Case Function?

Case facilitates conditional inquiries just like an if-then-else statement. In SQL case works with either the select or update clauses.

```
SELECT Name,  
CASE WHEN Sal > 0 AND Sal <= 100000 THEN 1  
WHEN Sal > 100000 AND Sal < 250000 THEN 2  
WHEN Sal > 250000 AND Sal < 500000 THEN 3  
ELSE 99  
END AS Category  
FROM Employee;
```

Query – Get third-highest salary

```
Select TOP (1) salary from  
(Select DISTINCT TOP (3) salary from Employee ORDER BY salary DESC) AS emp  
ORDER BY salary ASC;
```

Subquery,
Select DISTINCT TOP (3) salary from Employee ORDER BY salary DESC
will select the top 3 salaried employees in the table listed in descending order.

Now picking the top 1 from that list will give you the highest salary not the 3rd highest salary. Therefore, the second query reorders the 3 records in ascending order and then selects the top record (which will now be the lowest of those 3 salaries).

Not all databases support the TOP keyword. For example, MySQL and PSQL use the LIMIT keyword, as follows,

```
Select Salary from  
(Select DISTINCT Salary from Employee ORDER BY Salary DESC LIMIT 10) AS Emp  
ORDER BY Salary LIMIT 1;
```

SQL – What's the difference between Where-and-Having | Primary-and-Unique key

Where clause cannot be used with aggregates, but the Having clause can.

```
SELECT Student, SUM(score) AS total FROM Marks GROUP BY Student  
HAVING total > 70
```

Generally, we use WHERE prior to GROUP BY and HAVING after GROUP BY. The Where clause acts as a pre-filter and Having clause as a post-filter.

- **Primary Key:** uniquely identifies each row in a table. It enforces integrity constraints to the table. Only one primary key is allowed for a table. It does not accept any duplicate OR NULL values.
- **Unique Key:** uniquely identifies each row in a table. But a table can have more than one unique key unlike primary key and Unique key constraints can accept only one NULL value for column. Generally used when we want to enforce unique constraint on a column (or a column group) which is not a primary key.

SQL: We have a table with employees' data, write a query to reverse the gender column value.

Approach-1: Use intermediate temporary value, i.e., update all M to T > update all F to M > update all T to F.

Approach-2: Single query,

```
Update EmployeeTable set "gender" = (case "gender" when 'male' then 'female' else 'male' end);
```

SQL & Database understanding is a must-have skill for testers. It is always preferable to do some database checks when doing functional/web testing.

SQL query to add a column to table

ALTER TABLE command is used to alter any table layout,
ALTER TABLE table_name
ADD column_name datatype;

Example,

```
ALTER TABLE Customers  
ADD Email varchar(255);
```

SQL: What are the different types of SQL commands? Or what's DDL, DML and DCL?

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.

Data Definition Language (DDL): changes the structure of the table like creating a table, deleting a table, altering a table, etc.

CREATE / ALTER / DROP / TRUNCATE

Data Manipulation Language (DML): used to modify the database. It is responsible for all form of changes in the database.

INSERT | UPDATE | DELETE

Data Control Language (DCL): used to grant and take back authority from any database user.

GRANT | REVOKE

Transaction Control Language (TCL): only used with DML commands like INSERT, DELETE and UPDATE only.

COMMIT | ROLLBACK | SAVEPOINT

Data Query Language (DQL): used to fetch the data from the database.

SELECT

SQL: Any idea about Union, Union All, Intersect and Minus operations?

These are known as Set operations in SQL,

UNION: combine the results of two or more SELECT statements, eliminating duplicate rows. The number of columns and datatype must be same in both the tables.

*SELECT * FROM FirstTable UNION SELECT * FROM SecondTable;*

UNION ALL: similar to Union but it also shows the duplicate rows.

*SELECT * FROM FirstTable UNION ALL SELECT * FROM SecondTable;*

INTERSECT: combine two SELECT statements and return the records which are common from both SELECT statements. The number of columns and datatype must be same.

*SELECT * FROM FirstTable INTERSECT SELECT * FROM SecondTable;*

MINUS: combines results of two SELECT statements and return only those which belong to the first set.

*SELECT * FROM FirstTable MINUS SELECT * FROM SecondTable;*

JAVA Qs

Is Java a compiled language or an interpreted one?

Both. Yes, Java is both compiled as well as interpreted language. This is what makes it platform-independent. Java Code >> Compile (javac) >> Bytecode (portable code) >> Interpret (platform specific) >> Machine Code >> Execute.

Normal:

1. Java Compiler (javac) compiles Java code to Bytecode (which is platform-independent).
2. Java Interpreter (java) interprets this bytecode (line-by-line) & convert it to Machine language.
3. Execute.

Exception: JIT (Just In Time) Compiler

1. JVM maintains the count for no of times a function is executed.
2. If it exceeds the limit then JIT directly compiles the Java code into Machine language. No Interpretation.

In General,

- Compile: Source code >> Optimized Object Code (can be machine code or other optimized code)
- Interpret: Source Code >> Machine Code (to be executed)

What are bin & src folders in Java Automation?

'bin' and 'src' are not exclusive to Test automation. These are generic Java folders created to segregate the written-code and the compiled-code.

- **src**: Folder where our written code resides. Human readable code files.
- **bin**: Folder where the compiled files reside, that the Java Virtual Machine (JVM) executes... E.g. .class, .jar etc.

In simple words: Developer writes the Java code, i.e. .Java files >> These .Java files are stored in src folder >> Compile the code >> Java code is converted to Byte-code which can be read by JVM, i.e. .class files stored in bin folder.

JDK vs JRE vs JVM

JDK: Java Development Kit which provides the environment to develop and execute (run) the Java program.

- Development Tools (provide an environment to develop your java programs)
- JRE (to execute your java program).

It includes - Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) and other tools needed in Java development.

JRE: Java Runtime Environment provides the minimum requirements for executing (not develop) a Java application; it consists of the Java Virtual Machine (JVM), core classes, and supporting files. Note: You might have noticed that you associate a JRE [which contain all the libraries] with your test automation framework.

JVM: Whatever Java program you run using JRE or JDK goes into JVM and JVM is responsible for executing the java program [byte code] line by line.

Understanding < public static void main(String[] args) >

- **public:** Access modifier, making the main() method available globally so that JVM can invoke it from outside the class [as it is not present in the current class].
- **static:** Keyword, making the main() method class-related [not object] so that JVM can invoke it without instantiating the class.
- **void:** Keyword, used to specify that a method doesn't return anything. As soon as the main() method terminates, the java program terminates too.
- **main:** Identifier, that the JVM looks for as the starting point of the java program. It's not a keyword.
- **String[] args:** Parameter, stores Java command line arguments in an array of type String class. Note: the name <args> is not fixed, user can use any name.

Java 'This' keyword

For the Page Object classes, we generally define the constructor like,

```
public class BasePage
{
    WebDriver driver;
    public BasePage(WebDriver driver) throws Exception
    {
        this.driver = driver;
    }
}
```

Any idea why we use 'this' keyword? 'this' keyword in Java is used to refer to the current Object whose Method or constructor is being invoked. Simply stated – to distinguish between current object's instance variable and method parameter. i.e.

- static WebDriver driver;
- public BasePage(WebDriver driver)

Both statement contains the ‘driver’ variable. ‘this’ keyword here is used to refer to the current object’s instance variable (static WebDriver driver), i.e. to refer current object’s variable within the method.

What's 'Static' in Java? Why to use it?

Static, as in fixed. Applying it to Object-Oriented tech - something that doesn't change for every object, it's fixed for all objects.

Generally, fields are created separately in memory for each instance of a class, i.e., Object variables. But anything declared using static keyword belongs to the class instead of individual instances (objects). What does it mean? That every object of this class shares this same copy of the variable, method, etc. We can apply static keyword with variables, methods, blocks and nested class. The benefit – memory management of course.

```
public class Student{
    private String Name; //Object variable
    private int Age; //Object variable
    private String StudentId; //Object variable
    public static int NumberOfStudents = 0; //Class variable

    public Student(String name, int age, String studentId) {
        this.Name = name;
        this.Age = age;
        this.StudentId = studentId;
        NumberOfStudents++; //Increase the no of students whenever an object is created.
    }
}
```

The most common example is << public static void main(String args[]) >> declared static because it must be called before any object exists. Making a method static in Java is an important decision. Does it make sense to call a method/variable, even if no object has been constructed yet? If so, it should be static.

Static entity,

- Will be initialized first, before any class objects are created.
- Is accessed directly by the class name and doesn't need any object.
- Can access only static data. It cannot access non-static data (instance variables).
- Can call only other static methods and cannot call a non-static method.

Caution: Generally, it is bad practice to set the WebDriver instance as static. Instead create a base class that each test classes extend so that each test class has its own instance of WebDriver to be used (this is especially important with parallel execution), then just declare/define your WebDriver variable within the base class.

Can Static methods be over-ridden in Java?

Over-riding [run-time polymorphism] - A subclass provides an implementation of a method in superclass and which implementation to execute is decided at run-time according to the object [subclass OR superclass object] used for call.

No! Static methods cannot be over-ridden. Though you can declare a method with the same signature in a subclass.

Strange? Isn't that over-riding? No! It won't be overridden in the exact sense, instead, that is called 'method hiding'.

Why? Because there won't be any run-time polymorphism. Method call will be resolved at compile time itself depending on the class used to create object.

Note: Don't confuse it with overloading [compile-time polymorphism]. Static methods can be overloaded with same name, but different parameters.

What does 'Abstract' mean in Java?

Abstraction is a process of hiding the implementation details and only show important functionality. 'abstract' keyword is used to create an abstract class and method. The purpose of an abstract class is to specify the default functionality of an object and let its sub-classes to explicitly implement that functionality. Thus, it stands as an abstraction layer that must be extended and implemented by the corresponding sub-classes.

- Abstract class can't be instantiated (it needs to be extended) << abstract class A {} >>
| an abstract method contains a method signature, but no method body. << abstract void methodName(); >>
- Abstract class is used to provide default/common method implementation to all the subclasses.
- If you are extending any abstract class that have abstract methods, you must either provide the implementation of all abstract methods or make this sub-class abstract.
- An abstract class can have both abstract and non-abstract (or concrete) methods.
- If a class have abstract methods, then the class should also be abstract.
- An abstract class can have data member, abstract method, concrete method (body), constructor and even main() method.
- An abstract class may have static fields and static methods.
- Abstract method can never be final and static.

Abstract class and methods are usually declared where two or more subclasses are expected to do a similar thing in different ways through different implementations. Most common e.g. Shape as the abstract class >> its implementation provided by the Rectangle and Circle classes.

Note: Abstract classes are not Interfaces. They are different.

Some pointers about Interface!

- Interface can't have constructor. Cannot create object for interface.
- The fields of interfaces are public-static-final. Cannot declare as private or protected and must be initialized (final).
- Interface cannot implement another interface. Interface cannot extend a class. But can extend other interfaces.
- Interface cannot be declared as final (otherwise it will block extending them in other interfaces).
- An interface allows only public, abstract, default & static modifiers in a method declaration. No final, static & default methods added since Java 1.8.
- Cannot reduce the visibility of the methods [public] while implementing (over-riding a method) an interface.
- A class can implement multiple interfaces.

Why can't we create constructor for interface?

No, interface cannot have constructors. In order to call any method, we need an object – and since there is no need to have object of interface, there is no need of having constructor in interface.

Note: Constructor is being called during creation of object.

The impact of Java final keyword on variable-method-and-class.

- **final variable:** nothing but constants. We cannot change the value of a final variable once it is initialized. It will result in compilation error in case you try to modify the value.

Note: final variable that is not initialized at the time of declaration must be initialized in the constructor otherwise it will throw a compilation error.

- **final method:** cannot be overridden. Even though a sub class can call the final method of parent class, but it cannot override it [compilation error].
- **final class:** cannot extend a final class.

Trivia: final, finally and finalize are three different terms. final is used in exception handling and finalize is a method that is called by JVM during garbage collection.

The impact of Java static keyword on variable-method-and-class.

Static members belong to the class instead of a specific instance, i.e., access it without object.

- **static variable:** also known as class variables, i.e., common to all the instances (or objects) of the class. Only a single copy of static variable is created and shared among all the instances of the class.
- **static method:** can access static variables without using object (instance) of the class. Also, they can directly call only other static methods.
- **Static block:** to do the computation in order to initialize static variables, i.e., declare a static block that gets executed exactly once, when the class is first loaded.
- **static class:** A class can be made static only if it is a nested class. It doesn't need a reference of Outer class. Note: a static class cannot access non-static members of the Outer class.

Note: static is used for a variable or a method that is same for every instance of a class.

Java Collections

Collections. A 'Group of things'. It can be a group of variables, constants, classes, objects, anything. Since it is all about 'Objects' in 'Object Oriented programming' - it has to be a collection of Objects.

Set, List, Queue, ArrayList, Vector, LinkedList, HashSet, LinkedHashSet, TreeSet, etc. are all Collections. These form a part of the 'Java Collection Framework', a set of interfaces & classes to ease your implementation of different data structures. Each collection holds the data of some type structured as per its naming convention.

When to use different Collection types?

Collection framework provides interfaces and class implementations that enable easy data handling, i.e. store, retrieve and manipulate the data very effectively and easily. The basic types are Lists, Set & Maps.

Lists: no of items written one-by-one, usually ordered. Can be accessed using index. Say you want to store a list of web elements (e.g., all the hyperlinks).
Implementations - ArrayList, LinkedList, Vector.

Set: a collection of unique items, i.e., no duplicates. Say you don't want duplicates in a database. Implementations - HashSet, LinkedHashSet, TreeSet.

Maps: a collection of 'Key - Value' pairs. Say you want to save key-value pairs like,

- Username – STS | Password - ***** | Environment - SIT URL | Browser - Firefox
- Implementations - HashMap, LinkedHashMap, TreeMap.

The specific type to use depends on Test requirement, Test data management, multi-threading, sync-async, performance, etc.

What's the difference between Java Map and Hashmap?

Simply stated – Map is an interface, whereas HashMap is one of the implementations of this Map Interface.

- **Map Interface:** public interface Map<K,V>

An object that maps keys to values. A map cannot contain duplicate keys; each key can map to at most one value. The Map interface provides three collection views, which allow a map's contents to be viewed as a set of keys, collection of values, or set of key-value mappings.

- **Hashmap:** public class HashMap<K,V> extends AbstractMap<K,V> implements Map<K,V>, Cloneable, Serializable

Hash table-based implementation of the Map interface. This implementation provides all of the optional map operations, and permits null values and the null key. (The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls.) This class makes no guarantees as to the order of the map; in particular, it does not guarantee that the order will remain constant over time.

Difference between Java array and ArrayList?

'ArrayList' is one of the class in Java Collection framework which was introduced as a dynamic array. Since an 'array' is static in nature i.e., you cannot change its size once created, So, if you need an array which can resize itself then you should use the ArrayList. Add OR Delete objects from an ArrayList as required.

ArrayList provides more ways for iteration i.e., accessing all elements one by one. You can only use loop e.g., for, while, enhanced for and do-while to iterate over an array but you can also use Iterator and ListIterator class to iterate over ArrayList.

#Tip | Keep a Track of interviews

Keep a track of all the interviews that you attend. Track as in – type of rounds, common questions, topics, etc. The easiest way is to note all this in a Smartphone Note App as soon as you are done with the interview.

And what to do with these notes? Obvious – prepare for common trends, find the elaborative answers to the Qs, etc. It helps in preparation and also keeps the motivation & momentum going. Soon you will feel much more confident attending ANY interview.

#Tip: Keep trying to get Interviews scheduled

If you don't have few interviews lined up and got rejected in a recent interview - it will feel magnified because you have very little in your "pipeline" and you start feeling desperate. Aim to have more Interviews lined-up. Easier said than done, but aim for it. If you have some in pipeline then even if you get rejected in one, you know you have others lined-up and this will keep you motivated.

Difference between Java Set and List?

- List is an ordered collection that maintains the elements in insertion order while Set is a type of unordered collection, so elements are not maintained any order.
- List allows duplicates while Set doesn't allow duplicate elements.
- List permits any number of null values while Set permits only one null value in its collection.

List implementations: ArrayList, LinkedList etc.

Set implementations: HashSet, LinkedHashSet, TreeSet etc.

Return type of Array and List

Both Array and List returns an Object of respective classes. Array extends the "Object" class.

The use of Hashmap

Hashmap collection type is used extensively in an automation framework & script.

Why? Because of its simplicity & easy usage.

A Map-based collection class like Hashmap used for storing <Key & value> pairs makes it easy to capture & access the values. For example – using a Hashmap to store/access application's page labels & values. Tool-Selenium | Scripts-300 | Status-Automated | Environment-QA. As simple as that.

It also resembles how we, as humans, actually read values – we look at the field label or a header and then the value. Easy to access the value via field label.

How important is 'Exception Handling' in an automation framework?

Yeah, you don't want your test script to end abruptly without any info to debug. It leads to a lot of frustration later on.

The best approach is to use 'exception handling' to properly handle any exceptions thrown during the test execution.

You might want to continue execution despite minor exception, or follow an alternate path, or print a custom message or simply fail the test script with a screenshot.

Java's exception handling keywords – try, catch, throw, throws and finally – comes to rescue here.

It is important that you use them intelligently in order to build a robust framework which is easy to debug and report any exceptions.

Java Catch, Try, Finally, Throw

Exception handling is really important. You don't want your script execution to end abruptly without any info to debug.

Java's exception handling keywords – try, catch, throw, throws and finally comes to rescue here.

- **Try:** try block contain statements which may generate exceptions.
- **Catch:** catch block defines the action to be taken, when an exception occurs.
- **Throw:** used to throw an exception explicitly. Execution stops on encountering throw statement, and the closest catch statement is checked for matching type of exception.
- **Throws:** Any method that is capable of causing exceptions must list all the exceptions possible during its execution, so that anyone calling that method gets a prior knowledge about which exceptions are to be handled. Otherwise, you get compile time error saying unreported exception XXX must be caught or declared to be thrown.
- **Finally:** exception occur or not, finally block will always execute. I.e., run any cleanup type statements, no matter what happens in the protected code.

Java Exception class hierarchy

All objects within the Java exception class hierarchy extend from the Throwable superclass.

- indirectly thrown by the JVM
- can be directly thrown via a throw statement.
- only Throwables (or an inherited subclass) can be caught via a catch statement.

Throwable > Errors | Exceptions

- **Error:** "indicates serious problems that a reasonable application should not try to catch.", i.e., general practice is not to explicitly catch Error classes in code, since they should be dealt with through a change in the application architecture or refactoring.

E.g., AssertionError | LinkageError | ThreadDeath | VirtualMachineError

- **Exception:** "conditions that a reasonable application might want to catch.", i.e., typical errors that occur from time to time in most applications. E.g., `ArithmeticsExceptions` and `IllegalArgumentException` are found in the Exceptions subclass category.

E.g., CloneNotSupportedException | InterruptedException | ReflectiveOperationException | RuntimeException

#Tip | Know About the Company

Going for an interview? Always read about the prospective company. The geographic locations, headquarters, business units, product line, technology, etc. Any latest news about it. Just a brief would do.

It helps in answering the Question “Why do you want to join our firm?” Say the company is working on latest technologies, or a similar domain that you have worked on earlier, or have heard good about it from your friends, the company is recently named as best place to work for, etc. A quick glace on company website & news will give you enough insights.

Your homework reflects in your answer, and interviewers like people who have done some homework about the company. Better than sitting clueless!

How to create a custom exception in Java?

Java exceptions cover almost all general exceptions; however, we sometimes need to supplement these standard exceptions with our own – say Business logic exceptions OR if you wish to catch and provide specific treatment to a subset of existing Java exceptions.

How to create? Create a class which extends `java.lang.Exception` [checked] OR `java.lang.RuntimeException` [unchecked] class and create a constructor which calls the ‘super’ constructor.

Example,

```
public class CustomException extends Exception
{
    public CustomException(String errorMessage)
    {
        super(errorMessage);
    }
}
```

Note: We can call super constructor without any parameter or can also add a `java.lang.Throwable` parameter.

What's POJO?

POJO - Plain Old Java Object. As the name says – it's an ordinary Java Object, not a special object (Enterprise JavaBean).

POJO is usually used to describe a class that doesn't need to be a subclass of anything, or implement specific interfaces, or follow a specific pattern. I.e., only have a default constructor, private property and corresponding setter and getter methods.

Why use POJO? For effective encapsulation & abstraction - in public classes use setter/getter methods instead of public fields. An example would be to create a POJO (with set/get methods) to store the test data for a test script.

Strings are immutable in Java, why? And how to make them mutable?

Immutable, i.e., unchanging over time or unable to be changed. As simple as that.

Once created, you cannot change the value of String object in Java. It's final. It is set aside in a memory area called “String constant pool”. You can just have multiple reference variables pointing to the same object – but you cannot change its value.

```
String s1 = "java";
s1.concat(" rules");
System.out.println(s1); --- Yes, it will still print "java".
```

The benefit?

- **Security:** it's easier to operate with sensitive code when values don't change.
- **Synchronization:** immutable objects can be shared across multiple threads running simultaneously. They're thread-safe.
- **Hashcode:** the hash is calculated and cached during the first hashCode() call and the same value is returned ever since.
- **Performance:** "String constant pool" enhances the performance by saving heap memory and faster access.

How to make them mutable? use StringBuffer [thread-safe] OR StringBuilder class instead.

```
StringBuffer s1 = "java"; s1.append(" rules"); System.out.println(s1);  
StringBuilder s1 = "java"; s1.append(" rules"); System.out.println(s1);
```

This shall give you "java rules".

Difference between defining s = "text" and new String("text")

The basic difference is,

- new String("text"); explicitly creates a new and referentially distinct instance of a String object;
- String s = "text"; may reuse an instance from the string constant pool if one is available.

You very rarely would ever want to use the new String(anotherString) constructor.

"String(String original): Initializes a newly created String object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string. Unless an explicit copy of original is needed, use of this constructor is unnecessary since strings are immutable".

Any Java design pattern used in your automation framework?

A design pattern systematically explains a general design that addresses a recurring design problem in object-oriented systems. You might not be aware, but we do use some popular design patterns in test automation,

- **Factory pattern:** The most common – PageFactory - returns a Page Object with its fields fully initialized. It follows the Java Factory pattern - create object without exposing the creation logic.

- **Builder pattern:** Ever used '(new Actions(driver)).moveToElement(element).click().perform()'? Yeah! That's the builder pattern. Built on top of method-chaining [invoke multiple methods on the same object as a single statement, since methods return 'this' reference for a class instance] to ensure the thread-safety and atomicity of object creation. The inner static class is used for setter methods and then build to return an instance of outer class.
- **Singleton pattern:** restrict the instantiation of a class and ensures that only one instance of the class exists in the JVM. How? Make the constructor private and return a static class-object from your getInstance() method. Say, I login to rest [start services] before the start of TC execution and then use the same object [getInstance] to perform rest operations within the TC.

Java Collections – Map | Set | List.

The Collections Framework [collection of interfaces and classes] help in storing and processing the data efficiently.

- **List:** an ordered Collection (a sequence) where elements can be inserted or accessed by their position in the list, using a zero-based index. E.g., ArrayList | LinkedList | Vector. You might have used List to store a list of Web Elements.
- **Set:** A Collection that cannot contain duplicate elements. E.g., HashSet | TreeSet | LinkedHashSet. Might have used Set to operate on Window handles (since it doesn't allow duplicates).
- **Map:** A Collection that maps keys to values [cannot contain duplicate keys]. E.g., HashMap | TreeMap | LinkedHashMap. Might have used HashMap for data storage-read-and-update operations.

How is Set/HashSet implemented in the backend to achieve uniqueness?

Set: a collection of distinct objects, i.e., a set will never contain duplicate elements.

Basic: When we try to add a duplicate element to a set using add() method, it returns false, and element is not added to hashset, as it is already present.

How add() method checks whether the set already contains the specified element?

Logic: A HashSet internally creates a HashMap | The add() method of HashSet actually calls put() method on internally created HashMap object – key: specified element + value: constant Object “PRESENT”.

- **HashMap put(Key, Value)** - returns the previous value associated with key OR null [if there is no mapping].
- **HashSet add()** - check the return value of map.put(key, value) with null value to identify if the element is already present or not.

How are Java objects stored in memory?

The Java heap space is the area of memory used for dynamic memory allocation to store objects instantiated by applications running on the JVM.

- **Young Generation** – where all new objects are allocated and aged. A minor Garbage collection occurs when this fills up.
- **Old or Tenured Generation** – where long surviving objects are stored. When object in young generation reach its threshold age, the object is moved to the old generation.

Some pointers,

- If heap space is full, Java throws java.lang.OutOfMemoryError
- Heap space needs a Garbage Collector to free up unused objects.
- Use command "jmap" to take Heap dump and "jhat" to analyze that heap dump. [Use Profiler and Heap dump Analyzer tool to understand Java Heap space and how much memory is allocated to each object]
- For strings, we have a 'String Constant Pool' memory area allocated within Heap space.

Note: Java Heap space is different than Stack which is used to store call hierarchy and local variables.

Explain what collections you have used in project?

- **List:** to store a list of web elements (when a locator returns more than 1 web element using *driver.FindElement*)
- **Hashmap:** to store config key-value pairs like Env-URL | Username: name | Password: pwd | Client: XYZ | etc.
- **Set:** Generally used to store WindowHandles since it doesn't allow duplicate values.

Which mechanism is used in HashMap?

It works based on the hashing principle - mechanism of assigning unique code to a variable or attribute using an algorithm to enable easy retrieval. A true hashing mechanism should always return the same hashCode() when it is applied to the same object.

HashMap has an inner class called an Entry Class which holds the key and values.

- **Get:** First, it gets the hash code of the key object, which is passed, and finds the bucket location.
- **Put:** applies the hashCode to the key >> indexFor() method is used to get the exact location to store the Entry object >> if key.equals(k) is true, it will replace the value object inside the Entry class and not the key. This way, it prevents the duplicate key from being inserted.

Can we put duplicate values in any Set, Will it give any error?

The meaning of "sets do not allow duplicate values" is that when you add a duplicate to a set, the duplicate is ignored, and the set remains unchanged. This does not lead to compile or runtime errors: duplicates are silently ignored.

Difference between ArrayList and HashMap in Java?

- ArrayList implements List Interface while HashMap is an implementation of Map interface.
- ArrayList maintains the insertion order while HashMap does not maintain insertion order.
- ArrayList allows duplicate elements while HashMap doesn't allow duplicate keys but does allow duplicate values.

Does list maintain insertion order?

Both the ArrayList and LinkedList maintain the elements insertion order which means while displaying ArrayList and LinkedList elements the result set would be having the same order in which the elements got inserted into the List.

Difference between an array and an ArrayList in Java?

The main difference between array and ArrayList is that the array is static and the ArrayList is dynamic. We cannot change the size of the array once created, but ArrayList can increase its size automatically.

Differences between ArrayList and LinkedList?

- LinkedList can be iterated in reverse direction using descendingIterator() but to iterate over the ArrayList in reverse direction we need to write our own code.
- Insertions and Deletions are faster in LinkedList as compared to ArrayList because there is no need of resizing array and copying content to new array if array gets full which makes adding into ArrayList of O(n) in worst case, while adding is O(1) operation in LinkedList in Java. So, if we have more add or delete operations then we should go with Linked list.

- LinkedList has more memory overhead as each node holds both data and address of next and previous node but in ArrayList each index only holds actual object or we can say data.
- ArrayList internally uses a dynamic array while LinkedList internally uses a doubly linked list.
- ArrayList get method is faster than Linked list so fetching data or search operations is good in ArrayList.

Can we extend multiple classes in Java? Or does Java support multiple inheritance?

Multiple Inheritance is a feature of object-oriented concept, where a class can inherit properties of more than one parent class.

NO, Java doesn't support multiple inheritance.

Why? Simple, **to prevent ambiguity**. Consider a case where class B extends class A and Class C and both class A and C have the same method display(). Now java compiler cannot decide, which display method it should inherit. To prevent such situation, multiple inheritances is not allowed in java. Example, the diamond problem – Class B & C extend A and then Class D extends both B and C.

How can we achieve multiple inheritance in Java?

Java supports multiple inheritance of 'type', which is the ability of a class to **implement more than one interface**. An object can have multiple types: the type of its own class and the types of all the interfaces that the class implements. This means that if a variable is declared to be the type of an interface, then its value can reference any object that is instantiated from any class that implements the interface.

Update: Java 8 supports default methods where interfaces can provide default implementation of methods. And a class can implement two or more interfaces. In case both the implemented interfaces contain default methods with same method signature, the implementing class should explicitly specify which default method is to be used or it should override the default method.

How to implement a Singleton design pattern?

Used to restrict the instantiation of a class and ensures that only one instance of the class exists in the JVM. In other words, a singleton class can have only one object at a time per JVM instance.

Say you want to make a rest API connection and then use the same session details for further requests, i.e., create an object of rest connection and then use the same object for subsequent requests.

How?

1. We want to restrict object creation using constructor > **Make constructor private.**

```
private SingletonClass() {}
```

2. Make a **private static instance** (class-member) of this singleton class. Static because there should only be one copy.

```
private static SingletonClass SINGLE_INSTANCE = null;
```

3. Write a **static/factory method** that checks the static instance member for null and creates the instance. At last, it returns an object of the singleton class.

```
public static SingletonClass getInstance()
{
    If (SINGLE_INSTANCE == null)
    {
        Synchronized (SingletonClass.class)
        {
            If (SINGLE_INSTANCE == null)
            {
                SINGLE_INSTANCE = new SingletonClass();
            }
        }
    }
}
```

Why double check for null? Because there might be two threads running which gets inside the first if statement concurrently when the instance is null.

Java: Can we have a return statement in the catch or, finally blocks?

Yes! We can write a return statement in both catch and finally block.

How does it work? ‘finally’ block is always executed (the only exception is System.exit()) hence any return statement in try/catch is overridden by the one in finally block.

Note: a rule of thumb – never return from finally. Eclipse, for example, shows a warning for that snippet: “*finally block does not complete normally*”.

Any idea about Hashmap Collision concept?

Collision, i.e., crash between two things – hash code in this case!

Background: Hashmap stores key-value pairs in what is known as buckets. When a value is added, the hashCode() and hash() methods are called to compute the hash value which ultimately boils down to an index in the internal array or what we call a bucket location.

Problem: A hash code collision is a situation where two or more keys produce the same final hash value and hence point to the same bucket location or array index. Why? Coz two unequal objects in Java can have the same hash code.

Solution: Java implements a hash code collision resolution technique. If the hash codes of any two keys collide, their entries will still be stored in the same bucket. After finding the bucket location with the final hash value, each of the keys at this location will be compared with the provided key object using the equals API.

Java: What's the difference between extend and implement keywords?

Extends: to indicate that a class is derived from the base class using inheritance, i.e., extend the functionality of the parent class to the subclass. In Java, multiple inheritances are not allowed due to ambiguity. Therefore, a class can extend only one class.

Implements: used to implement an interface. To access the interface methods, the interface must be “implemented” by another class. A class can implement any number of interfaces at a time.

I.e., *Extends* is used when inheriting a class, and *Implements* is used when a class is implementing an interface.

Note: An interface can ‘*extend*’ any number of other interfaces.

Java Programs

Commonly asked

Java Program to Find Odd or Even number

Logic: Iterate through the array, check if the no is divided by 2 or not. How? By getting the remainder when divided by 2.

```
public static void main(String args[])
{
    int a[] = {1,2,5,6,3,2};
    for(int i=0;i<a.length;i++)
    {
        If ( a[i] % 2 != 0)
            System.out.println(a[i]+” : Odd”);
        else
            System.out.println(a[i]+” : Even”);

    }
}
```

Java Program to Swap two numbers without using third variable

Logic: The arithmetic operators for addition and subtraction can be used to perform the swap without using a third variable.

```
public static void main(String args[])
{
    int x = 10, y = 50;
    x = x + y;
    y = x - y;
    x = x - y;
    System.out.println("After swap x=" + x + "and y=" + y);
}
```

Write a java program to find Prime number

Logic: Looping from 2 to number/2, since a number is not divisible by more than its half. Inside the for loop, check if the number is divisible by any number in the given range (2...num/2).

- If divisible, flag is set to true and we break out of the loop. I.e., not a prime number.
- If it isn't divisible by any number till the loop ends, flag is false and it's a prime number.

```

public static void main(String[] args)
{
    int num = 29;
    boolean flag = false;
    for (int i = 2; i <= num / 2; ++i)
    {
        if (num % i == 0)
        {
            flag = true;
            break;
        }
    }
    if (!flag)
        System.out.println(num + " is a prime number.");
    else
        System.out.println(num + " is not a prime number.");
}

```

Java Program to Find Factorial of a Number

Logic: Factorial of n is the product of all positive descending integers, denoted by n!.

$$4! = 4 * 3 * 2 * 1 = 24$$

Using loop,

```

public static void main(String args[])
{
    int i, fact=1;
    int number=5;
    for(i=1;i<=number;i++)
    {
        fact=fact*i;
    }
    System.out.println("Factorial of "+number+" is: "+fact);
}

```

Using recursion,

```

static int factorial(int n)
{
    if (n == 0)
        return 1;
    else
        return(n * factorial(n-1));
}

```

```
public static void main(String args[])
{
    int i,fact=1;
    int number=4;
    fact = factorial(number);
    System.out.println("Factorial of "+number+" is: "+fact);
}
```

Java Program to Reverse Number

Logic:

- Take the number's modulo by 10
- Multiply the reverse number by 10 and add modulo value into the reverse number.
- Divide the number by 10.
- Repeat above steps until number becomes zero.

```
// Function to reverse the number
static int reverse(int n)
{
    int rev = 0; // reversed number
    int rem; // remainder
    while(n>0)
    {
        rem = n%10;
        rev = (rev*10) + rem;
        n = n/10;
    }
    return rev;
}
```

- rev = 0 | n = 1234
- rev = (0*10) + 4 = 4 | n = 123
- rev = (4*10) + 3 = 43 | n = 12
- rev = (43*10) + 2 = 432 | n = 1
- rev = (432*10) + 1 = 4321 | n = 0

Java Program to print Fibonacci Series

The Fibonacci series is a series of elements where, the previous two elements are added to get the next element, starting with 0 and 1.

Input: N = 10

Output: 0 1 1 2 3 5 8 13 21 34

Using Recursion,

```
// Function to return the sum of previous 2 numbers
static int fib(int n)
{
    if (n <= 1)
        return n;
    else
        return fib(n - 1) + fib(n - 2);
}

public static void main(String args[])
{
    int N = 10;
    for (int i = 0; i < N; i++)
    {
        System.out.print(fib(i) + " ");
    }
}
```

Logic: Recursively iterate from value N to 1:

- Base case: If the value called recursively is less than 1, then return 1 the function.
- Recursive call: If the base case is not met, then recursively call for previous two value as: $\text{fib}(N - 1) + \text{fib}(N - 2)$;
- Return statement: At each recursive call (except the base case), return the recursive function for the previous two value as: $\text{fib}(N - 1) + \text{fib}(N - 2)$;

Write a java program to find Palindrome number

Logic: First reverse digits of number, then compare the reverse of number with actual number. If both are same, then return true, else false.

```
public static void main(String args[])
{
    int r, sum=0, temp;
    int n=454;
    temp=n;
    while(n>0)
    {
        r=n%10;
        sum=(sum*10)+r;
        n=n/10;
    }
}
```

```
if(temp==sum)
    System.out.println("palindrome number ");
else
    System.out.println("not palindrome");
}
```

Write a java program to find number of digits in given number.

Logic: Create an integer (count) > initialize it with 0. Divide the number with 10, till the number is 0 and for each turn > increment the count.

```
public static void main(String args[])
{
    Scanner sc = new Scanner(System.in);
    int count = 0;
    System.out.println("Enter a number:");
    int num = sc.nextInt();
    while(num!=0)
    {
        num = num/10;
        count++;
    }
    System.out.println("Number of digits: "+count);
}
```

Write a java program to find the duplicate words and their number of occurrences in a string.

Logic:

- Convert string to lowercase & split using space delimiter.
- Create empty HashMap of type String & Integer
- Iterate through String array
- Check whether particular word is already present in the HashMap using containsKey method.
- If it contains, then increase the count value by 1 using put(K, V) method.
- Otherwise insert using put() method of Map with count value as 1.
- Finally, print Map using keySet() or entrySet() method.

```

public static void main(String[] args)
{
    String str = "This is a program to find duplicate words in a string, again! a program";
    String[] words = str.toLowerCase().trim().split(" ");
    Map<String, Integer> duplicateString = new HashMap<>();
    int count = 1;
    for (String x : words)
    {
        if (duplicateString.containsKey(x))
        {
            duplicateString.put(x, duplicateString.get(x) + 1);
        }
        else
        {
            duplicateString.put(x, count);
        }
    }
    System.out.println("Duplicate Words in a String :");
    for (Map.Entry a : duplicateString.entrySet())
    {
        int val = (Integer) a.getValue();
        if (val > 1)
        {
            System.out.println(a);
        }
    }
}

```

Write a java program to count the number of words in a string

Using String.split() method

Logic: Solution uses the regular expression "\s+" to split the string on whitespace. The split method returns an array, the length of the array is your number of words in a given String.

```

public static int countWordsUsingSplit(String input)
{
    if (input == null || input.isEmpty())
    {
        return 0;
    }
    String[] words = input.split("\s+");
    return words.length;
}

```

Note: \s+ will find one more space and split the String accordingly.

Write a java program to count the total number of occurrences of a given character in a string. Or to find duplicate characters in a string.

Using charAt() method and loop,

- Iterate throughout the length of the input String
- Check whether each character matches the character to search
 - If yes, increment the count
 - else do nothing

```
public static void main(String args[])
{
    String input = "aaaabbccAAdd";
    char search = 'a';
    int count=0;
    for(int i=0; i<input.length(); i++)
    {
        if(input.charAt(i) == search)
            count++;
    }
    System.out.println("The Character "+search+" appears "+count+" times.");
}
```

Using Hashmap,

For each character in the input string, we will put it in our Map with value 1, if it is seen for the first time. If the character repeats or the character is already present in our Map, we update the value of the character in the map by adding 1 to it.

```
public static void main(String args[])
{
    String input = "aaaabbAAAAcccded";
    char search = 'a';
    Map<Character, Integer> hash = new HashMap<Character, Integer>();
    for(int i=0;i<input.length();i++)
    {
        if(hash.containsKey(input.charAt(i)))
            hash.put(input.charAt(i), hash.get(input.charAt(i))+1);
        else
            hash.put(input.charAt(i), 1);
    }
    int result = hash.get(search);
    System.out.println("The Character "+search+" appears "+result+" times.");
}
```

Write a java program to reverse a string.

Using StringBuilder/StringBuffer,

```
public static String reverseString(String str)
{
    StringBuilder sb=new StringBuilder(str);
    sb.reverse();
    return sb.toString();
}
```

Using Reverse iteration,

```
public static String reverseString(String str)
{
    char ch[]=str.toCharArray();
    String rev="";
    for(int i=ch.length-1;i>=0;i--)
    {
        rev+=ch[i];
    }
    return rev;
}
```

Write a java program to reverse each word of a given string.

Logic: By using reverse() method of StringBuilder class, we can reverse given string. By the help of split("\\s") method, we can get all words in an array. To get the first character, we can use substring() or charAt() method.

Input: my name is khan
Output: ym eman si nahk

```
public static String reverseWord(String str)
{
    String words[] = str.split("\\s");
    String reverseWord="";
    for(String w:words)
    {
        StringBuilder sb=new StringBuilder(w);
        sb.reverse();
        reverseWord+=sb.toString()+" ";
    }
    return reverseWord.trim();
}
```

Write a java program to reverse a given string with preserving the position of spaces

Input: "abc de"

Output: "edc ba"

Logic:

- Create a string to store results. Mark the space position of the given string in this string.
- Insert the character from the input string into the result string in reverse order.
- While inserting the character check if the result string already contains a space at index 'j' or not. If it contains, we copy the character to the next position.

```
string reverses(string str)
{
    int n = str.size();
    string result(n, '\0');
    for (int i = 0; i < n; i++)
        if (str[i] == ' ')
            result[i] = ' ';

    int j = n - 1;
    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] != ' ')
        {
            if (result[j] == ' ')
                j--;
            result[j] = str[i];
            j--;
        }
    }
    return result;
}
```

Write a Java Program to find the second-highest number in an array.

Logic-1: sort the array in descending order and then return the second element which is not equal to the largest element from the sorted array.

```

static void print2largest(int arr[], int arr_size)
{
    int i, first, second;
    if (arr_size < 2)
    {
        System.out.printf(" Invalid Input ");
        return;
    }
    Arrays.sort(arr);
    for (i = arr_size - 2; i >= 0; i--)
    {
        if (arr[i] != arr[arr_size - 1])
        {
            System.out.printf("The second largest " + "element is %d\n", arr[i]);
            return;
        }
    }
    System.out.printf("There is no second " + "largest element\n");
}

```

Logic-2:

- Initialize highest and secondHighest with minimum possible value.
- Iterate over array.
- If current element is greater than highest
 - Assign secondHighest = highest
 - Assign highest = currentElement
- Else if current element is greater than secondHighest
- Assign secondHighest =current element.

```

public static int findSecondLargestNumberInTheArray(int array[])
{
    int highest = Integer.MIN_VALUE;
    int secondHighest = Integer.MIN_VALUE;
    for (int i = 0; i < array.length; i++)
    {
        if (array[i] > highest)
        {
            secondHighest = highest;
            highest = array[i];
        }
        else if (array[i] > secondHighest && array[i] != highest)
            secondHighest = array[i];
    }
    return secondHighest;
}

```

Write a Java Program to remove all white spaces from a string.

Logic: We can either use “replaceAll” method to remove all spaces OR iterate over the char array to add only the non-space values to new String.

```
public static void main(String[] args)
{
    String str = "India Is My Country";

    //1st way
    String noSpaceStr = str.replaceAll("\s", ""); // using built in method
    System.out.println(noSpaceStr);

    //2nd way
    char[] strArray = str.toCharArray();
    StringBuffer stringBuffer = new StringBuffer();
    for (int i = 0; i < strArray.length; i++)
    {
        if ((strArray[i] != ' ') && (strArray[i] != '\t'))
        {
            stringBuffer.append(strArray[i]);
        }
    }
    String noSpaceStr2 = stringBuffer.toString();
    System.out.println(noSpaceStr2);
}
```

Write a Java program to find the longest substring from a given string which doesn't contain any duplicate characters?

Logic: We start traversing the string from left to right and maintain track of:

- the current substring with non-repeating characters with the help of a start and end index
- the longest non-repeating substring output
- a lookup table of already visited characters
- For every new character, we look for it in the already visited characters. If the character has already been visited and is part of the current substring with non-repeating characters, we update the start index. Otherwise, we'll continue traversing the string.

```

String getUniqueCharacterSubstring(String input)
{
    Map<Character, Integer> visited = new HashMap<>();
    String output = "";
    for (int start = 0, end = 0; end < input.length(); end++)
    {
        char currChar = input.charAt(end);
        if (visited.containsKey(currChar))
        {
            start = Math.max(visited.get(currChar)+1, start);
        }
        if (output.length() < end - start + 1)
        {
            output = input.substring(start, end + 1);
        }
        visited.put(currChar, end);
    }
    return output;
}

```

Java program to calculate SUM of numbers in an array.

Logic: Create an empty variable (sum) > Initialize it with 0 > Traverse through each element of the array and add each element to sum > Print sum.

```

public static void main(String[] args)
{
    int [] arr = new int [] {1, 2, 3, 4, 5};
    int sum = 0;
    for (int i = 0; i < arr.length; i++)
    {
        sum = sum + arr[i];
    }
    System.out.println("Sum of all the elements of an array: " + sum);
}

```

Java program to find the sum of digits of a number.

Input: 95318

Output = 9+5+3+1+8 = 26

Logic:

- Declare a variable (sum) to store the sum of numbers and initialize it to 0.
- Find the remainder by using the modulo (%) operator. It gives the last digit of the number (N).

- Add the last digit to the variable sum.
- Divide the number (N) by 10. It removes the last digit of the number.
- Repeat the above steps (2 to 4) until the number (N) becomes 0.

```
public static void main(String args[])
{
    int number, digit, sum = 0;
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the number: ");
    number = sc.nextInt();
    while(number > 0)
    {
        digit = number % 10;
        sum = sum + digit;
        number = number / 10;
    }
    System.out.println("Sum of Digits: "+sum);
}
```

Java program to multiply 2 numbers without using * operator.

Suppose, we want to multiply 3 by 4 which gives 12 as the result. The same can be achieved by adding 3 four times i.e. ($3 + 3 + 3 + 3 = 12$) or by adding 4 three times i.e. ($4 + 4 + 4 = 12$). Both give the same result. Therefore, we can implement the logic using recursion.

```
public static void main(String args[])
{
    int x=4, y=4, sum=0;
    for(int i=1;i<=x;i++)
    {
        sum=sum+y;
    }
    System.out.println("The multiplication of "+x+" and "+y+" is: "+sum);
}
```

Find substring in a string without using contains.

`String.indexOf()`: The `.indexOf()` method is a bit more crude than the `.contains()` method, but it's nevertheless the underlying mechanism that enables the `.contains()` method to work. It returns the index of the first occurrence of a substring within a String, and offers a few constructors to choose from:

- `indexOf(int ch)`
- `indexOf(int ch, int fromIndex)`
- `indexOf(String str)`
- `indexOf(String str, int fromIndex)`

Find duplicates in an array.

Logic: Duplicate elements can be found using two loops. The outer loop will iterate through the array from 0 to length of the array. The outer loop will select an element. The inner loop will be used to compare the selected element with the rest of the elements of the array. If a match is found which means the duplicate element is found then, display the element.

```
public static void main(String[] args)
{
    int [] arr = new int [] {1, 2, 3, 4, 2, 7, 8, 8, 3};
    System.out.println("Duplicate elements in given array: ");
    for(int i = 0; i < arr.length; i++)
    {
        for(int j = i + 1; j < arr.length; j++)
        {
            if(arr[i] == arr[j])
                System.out.println(arr[j]);
        }
    }
}
```

Checking anagram strings.

An anagram of a string is another string that contains the same characters, only the order of characters can be different. LISTEN – SILENT are anagrams.

Logic: Sort both strings > Compare the sorted strings.

```
static boolean areAnagram(char[] str1, char[] str2)
{
    int n1 = str1.length;
    int n2 = str2.length;
    if (n1 != n2)
        return false;
    Arrays.sort(str1);
    Arrays.sort(str2);
    for (int i = 0; i < n1; i++)
        if (str1[i] != str2[i])
            return false;
    return true;
}
```

Find first non-repeated character in a given string.

Input: TesterTested

Output: 'R'

Logic:

- Make a hash_map which will map the character to their respective frequencies.
- Traverse the given string using a pointer.
- Increase the count of current character in the hash_map.
- Now traverse the string again and check whether the current character hasfrequency=1.
- If the frequency>1 continue the traversal.
- Else break the loop and print the current character as the answer.

```

class Parikshram
{
    static final int NO_OF_CHARS = 256;
    static char count[] = new char[NO_OF_CHARS];
    static void getCharCountArray(String str)
    {
        for (int i = 0; i < str.length(); i++)
            count[str.charAt(i)]++;
    }

    static int firstNonRepeating(String str)
    {
        getCharCountArray(str);
        int index = -1, i;
        for (i = 0; i < str.length(); i++)
        {
            if (count[str.charAt(i)] == 1)
            {
                index = i;
                break;
            }
        }
        return index;
    }

    public static void main(String[] args)
    {
        String str = "TesterTested";
        int index = firstNonRepeating(str);
        System.out.println(index == -1 ? "Either all characters are repeating or string"
        + "is empty" : "First non-repeating character is " + str.charAt(index));
    }
}

```

How to find largest and smallest element in Array?

Logic: Traverse the array iteratively and keep track of the smallest and largest element until the end of the array.

```
public static void main(String args[])
{
    int large, small, i;
    int a[] = new int[]{1, 2, 3, 4, 5};
    int n = a.length;
    large = small = a[0];
    for(i=1; i < n; ++i)
    {
        if(a[i] > large)
            large = a[i];

        if(a[i] < small)
            small = a[i];
    }
    System.out.print("\nThe smallest element is " + small );
    System.out.print("\nThe largest element is " + large );
}
```

Find two numbers of which the product is maximum in an array.

Logic-1: A Simple Solution is to consider every pair and keep track of the maximum product.

```
static void maxProduct(int arr[], int n)
{
    if (n < 2)
    {
        System.out.println("No pairs exists");
        return;
    }

    int a = arr[0], b = arr[1];
    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n; j++)
            if (arr[i] * arr[j] > a * b)
            {
                a = arr[i];
                b = arr[j];
            }

    System.out.println("Max product pair is {" + a + ", " + b + "}");
}
```

Logic-2: Sort input array in increasing order,

- If all elements are positive, then return the product of the last two numbers.
- Else return a maximum of products of the first two and last two numbers.

```
static void maxProduct(int arr[], int n) {
    Arrays.sort(arr);
    int num1, num2;
    int sum1 = arr[0] * arr[1];
    int sum2 = arr[n - 1] * arr[n - 2];
    if (sum1 > sum2) {
        num1 = arr[0];
        num2 = arr[1];
    }
    else {
        num1 = arr[n - 2];
        num2 = arr[n - 1];
    }
    System.out.println("Max product pair = " + "{" + num1 + "," + num2 + "}");
}
```

Java program to read from file line by line.

Using the Java BufferedReader class is the most common and simple way to read a file line by line in Java. It belongs to java.io package. Java BufferedReader class provides readLine() method to read a file line by line.

```
public static void main(String args[]) {
    try {
        File file=new File("Demo.txt");
        FileReader fr=new FileReader(file);
        BufferedReader br=new BufferedReader(fr);
        StringBuffer sb=new StringBuffer();
        while((line=br.readLine())!=null)
        {
            sb.append(line);
            sb.append("\n");
        }
        fr.close();
        System.out.println("Contents of File: ");
        System.out.println(sb.toString());
    }
    catch(IOException e) {
        e.printStackTrace();
    }
}
```

SELENIUM, TESTNG & Cucumber

Selenium Galaxy

Selenium is the de-facto tool for Web Automation now-a-days. But it doesn't work in isolation. There are other tools that complete the Selenium Galaxy of automation.

- **Selenium Webdriver:** web browser automation.
- **IDE like Eclipse/IntelliJ:** a text editor with additional support for developing, compiling and debugging applications.
- **TestNG:** a testing framework inspired from JUnit and NUnit.
- Build Automation like **Maven/Ant/Gradle:** automating the creation of a software build and the associated processes.
- **Cucumber:** A Behavior Driven Development (BDD) framework.
- **Jenkins:** for Continuous Integration.

Some other not-so-popular tools to add,

- **AutoIT:** to work on windows-based popups.
- **Apache POI:** to work on Office documents (read/write).
- **ExtentReports/ReportNG:** generate HTML logs & reports.

Selenium Positive & Negative aspects

"Selenium", the de-facto top automation testing tool. The Positives are quite easy,

- Open-source testing tool.
- Capability to operate on almost every OS.
- Supports multiple languages.
- Independent of the language that the web application is using.
- Supports a range of browsers.
- Very dynamic developer & helping community and user base.
- Robust element locators.
- Convenient to implement frameworks that revolve around OOP.
- Support for integration of open-source frameworks like TestNG + Version control using Maven, Jenkins.
- Execute simultaneous tests leveraging various browsers on various machines.
- Supports Web and Mobile Web Applications.

But what about the negatives? Instead of calling it 'Negative', call it as 'Limitations',

- No dedicated official technical support.
- Supports only Web based applications.
- Dependence on third-party tools for complete benefit, unlike proprietary all-in-one tools.
- Limited support for Image-based Testing, Captcha and Bar code readers.

- Unstable new releases. It evolves with time.
- No implicit Test Tool integration for Test Management.
- No Built-in Reporting facility.
- Lot of challenges with IE browser.

Being an open-source tool, these limitations are okay to live with.

It is always advisable to keep Test Data separate from Automation scripts. In Selenium, How do you read data from an Excel file using Java?

Apache POI is an Open-source Java library to manipulate file formats based on Microsoft Office. We use XSSF (XML SpreadSheet Format) to read an Excel (.xlsx) file format.

1. Since we have to read contents - Open existing excel using a File Input Stream.
2. Get Workbook >> & then specific Sheet.
3. Iterate through every row in excel file.
 - 3.a. Iterate through every cell within each row.
 - 3.b. Print the cell value.
4. Flush & Close the Input Stream.

```
File src = new File("excel file path");
FileInputStream fis = new FileInputStream(src);
XSSFWorkbook wb = new XSSFWorkbook(fis);
XSSFSheet sheet = wb.getSheet(sheetName);
XSSFRow row; XSSFCell cell;
Iterator rows = sheet.rowIterator();
while (rows.hasNext()) {
    row = (XSSFRow) rows.next();
    Iterator cells = row.cellIterator();
    while (cells.hasNext()) {
        cell = (XSSFCell) cells.next();
        System.out.println(cell.getStringCellValue());
    }
}
fis.flush();
fis.close();
```

Which is better to use in Selenium automation? Sleep or Wait?

Both Thread.sleep and Object.wait make the current thread wait for a specified amount of time. This is useful when the current thread needs to wait for other thread before it can proceed. For example, web element might not be clickable or enabled, or in the expected state.

Instead of using Thread.sleep, a better approach is to check for right state. Selenium provides a set of common ExpectedConditions that can be leveraged on top of webdriverwait. Unlike Thread.sleep (where locks are not released), in Object.wait locks are released as the thread goes to sleep. This approach helps make the test more stable and reliable.

What's 'DataProvider' in TestNG? Any examples...

What does the name 'DataProvider' suggest? Yeah! Something that provides data. Test Data is important in Software Testing, say you want to login to an application – it needs a username and password values – it's nothing but the Test data.

Method1 – Data Provider method – annotated with @DataProvider - Name 'LoginData' – returns an array of objects.

```
public class DataProviderClass {  
    @DataProvider(name = "LoginData")  
    public static Object[][] dataProviderMethod() {  
        return new Object[][] { {"Username1", "Password@1"}, {"Username2", "Password@2"}  
    };  
    }  
}
```

Rows: # times test needs to be repeated.

Columns: # parameters in Test method

Method2 – Test method 'Login' – Executed using data provided by Method1-LoginData.

```
public class TestClass {  
    @TestdataProvider = "LoginData", dataProviderClass = DataProviderClass.class)  
    public void Login(String username, String password) {  
        System.out.println("Username: " + username + " and Password: " + password);  
    }  
}
```

Output –

Username: Username1 and Password: Password@1

Username: Username2 and Password: Password@2

Write data-driven tests (same test run multiple times with diff. Test data).

How do you read data from a Web Table using Selenium?

Simple, identify the table >> Go to row-2 >> then column-3 >> and then read data ;-) For dynamic table it's a bit different – just before going to specific cell we need to know the no of rows & columns rendered. Post that, steps will be same to fetch specific data.

Identify the table,

```
WebElement Table = driver.findElement(By.ByClassName.className("dataTable"));
```

Read specific cell value,

```
WebElement cell = driver.findElement(By.xpath("//table/tbody/tr[2]/td[3]"));
System.out.println("Row 2 and Column 3: "+cell.getText()+" .");
```

Fetch # of rows in a table,

```
List<WebElement> rows = Table.findElements(By.tagName("tr"));
rows_count = rows.size();
```

Fetch # of columns in a row,

```
List<WebElement> columns = rows.get(i).findElements(By.tagName("td"));
col_count = columns.size();
```

Iterate through the table,

```
for(int i= 0; i<rows_count; i++){
List<WebElement> columns = rows.get(i).findElements(By.tagName("td"));
    col_count = columns.size();
    for(int j=0; j< col_count; j++){
        cellText = columns.get(j).getText();
        System.out.print(cellText+ " ");
    }
    System.out.println("");
}
```

Why do we use PageFactory?

Why do we use PageFactory? Yeah! To initialize the web elements.

But why can't we use the simple,

```
LoginPage loginPage = new LoginPage(driver);
```

Instead of using,

```
LoginPage loginPage = PageFactory.initElements(driver, LoginPage.class);
```

This is one query that I forgot to study about. Use of PageFactory is NOT mandatory and it seems to be a stylistic choice. The difference is quite simple,

- Initialize elements to avoid NullPointerExceptions.
- Use of annotations for readability.

Why will you get NullPointerExceptions? Because you might forget to initialize a WebElement and use it. I.e.

```
private WebElement test;  
public void tptest() {  
    test.sendKeys("NullPointer"); //test element is not initialized yet.  
}
```

PageFactory automatically initializes the Page element whenever a call is made. How?

Simply by replacing a by b.

- a. test.sendKeys("NullPointer");
- b. driver.findElement(By.id("test")).sendKeys("NullPointer"); //or can use name instead of id.

This way, you won't get a NullPointerException. Though you might get 'ElementNotFound' one :P

Selenium assert or verify? Which one do you prefer?

Basic Difference: 'assert' terminates the execution whereas 'verify' marks the step as fail but continue the execution.

It is recommended to use a combination of assert and verify.

- **assert:** to check if the entry criteria are met. E.g., Login successful OR If the window is available - then further validate the fields - else stop. Just like a Sanity check :P to continue further execution.
- **verify:** to validate the individual steps within a test case.

'verify' helps in gauging the script stability, i.e., if the script is executing till the end. We can then analyze the individual step failures together.

Using 'assert' you go step-by-step in script fixing.

Note: There are arguments for/against both in the technical landscape :)

What all concepts/tools do you use to create Selenium logs & reports?

Before we list some options, let's look at Listeners first.

Listeners: As the name suggests, these are the spectators that watch over your test execution (or listening) and react (log/report) to various events like click, type, move, etc.

- **WebDriver Listeners:** listen the events triggered by webdriver like beforeClickOn, afterClickOn, beforeFindBy, afterFindBy, etc. and take actions.
- **TestNG reporter:** Reporter is a separate class in TestNG to log some text after the desired action.
- **TestNG Listeners:** TestNG provides many types of listeners but mostly used ones are ISuiteListener & ITestListener.

- **ExtentReports:** Extent Reports is a customizable HTML report which can be integrated into Selenium WebDriver using JUnit and TestNG frameworks.
- **Log4j:** Apache Log4j is a Java-based logging utility. One of several Java logging frameworks.
- **ReportNG:** a simple HTML/XML reporting plug-in for the TestNG framework.

How do you avoid OR resolve Selenium's StaleElementReferenceException?

The common reasons for this exception,

- Element has been deleted entirely.
- Element is no longer attached to the DOM.
- Webpage on which the element was part of has been refreshed.
- Element has been refreshed by a JavaScript or Ajax call.

How to avoid it? By adding appropriate waits. Initialize OR Find the WebElement after waiting for its visibility or when it enables.

How to resolve it? Simple, discard the current reference you have and replace it by locating the WebElement once again. And don't forget to add relevant wait conditions this time :-P

What is a WebDriver?

--public interface WebDriver extends SearchContext --

WebDriver is an interface which represents an “idealised web browser”.

Why interface? Because every browser has their own logic to perform actions such as launch, close, load URL, handling web elements, etc. Same operations are performed in different ways by different browsers. It's difficult to manage (considering browser changes as well) if implemented at WebDriver level. Therefore, WebDriver is built as an interface which consist of all basic methods which could be performed on a browser. And then implemented by respective browser drivers.

Methods: close() | findElement(By by) | findElements(By by) | get(java.lang.String url) | getCurrentUrl() | getPageSource() | getTitle() | getWindowHandle() | getWindowHandles() | manage() | navigate() | quit() | switchTo()

Implementing Classes: ChromeDriver, EdgeDriver, EventFiringWebDriver, FirefoxDriver, InternetExplorerDriver, OperaDriver, RemoteWebDriver, SafariDriver

#Tip: Know your learnings

Interviewers are keen to know about your learnings. It could be practical experience from projects or self-learned from books, blogs or trainings.

Be ready with your learnings from all perspectives – technical, process, domain, testing, etc. It is better to be prepared for what's coming!

#Tip: Know your success stories & challenging times

"What is the most challenging situation you had faced?" is one of the most common interview question now-a-days. So, it is important that you have few examples ready from your personal experience.

Same goes for project success, i.e., enhancements or innovations that you brought to the project which yielded good results.

Be ready with these two examples!

What's the sequence of TestNG annotation execution?

The hierarchical rep of execution order of TestNG annotations is S-T-C-M,

```
BeforeSuite
  > BeforeTest
    > BeforeClass

      > BeforeMethod
        @Test – method1
      > AfterMethod
      > BeforeMethod
        @Test – method2
      > AfterMethod

    > AfterClass
  > AfterTest
AfterSuite
```

This sequence will be clear if you look at sample TestNG xml file,

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="suite" verbose="1" >
  <test name="test" >
    <classes>
      <class name="class-1"/> --- each class has multiple methods @test-1 | @test-2 | @test-n
      <class name="class-2"/>
      <class name="class-3"/>
    </classes>
  </test>
</suite>
```

Selenium-Java: How to identify all broken links on a web page?

Broken links - links that don't work. Say page is no longer available, page was moved without a redirect, URL structure of a website was changed, etc.

Steps to identify broken links,

1. Identify all the links - usually image `` and anchor tags `<a/>` on a web page containing href attribute.
2. Iterate through all links checking for response code when sending a HTTP request. Based on the response we can figure out if the link is broken or not.

How? `HttpURLConnection` class from Java helps here - used to make HTTP requests to the webserver hosting the links.

```
HttpURLConnection connection = (HttpURLConnection) url.openConnection();
connection.connect();
String response = connection.getResponseMessage();
connection.disconnect();
```

Trust only those URLs that returned OK, rest should be verified once.

TestNG Soft-Asset & Hard-Assert

Assertion: a statement of fact. In testing terms, your actual validation [expected = actual] which marks the script as pass or fail.

TestNG provides us with two kind of assertions,

HardAssert: instantly throws an exception when a assert statement fails and jumps to the next test in the test suite. “Assertion” class provides various assertion methods like assertEquals | assertTrue | assertFalse | assertNull | assertNotNull | etc.

But what if you want to continue the test case execution even if some of the assertions fail?

SoftAssert: does not throw an exception when an assert fails and would continue with the next step. It basically collects all the assertions throughout the @Test method. “SoftAssert” class also provides various assertion methods like assertEquals | assertTrue | etc.

Note: You need to instantiate a SoftAssert object within a @Test method, and call softAssert.assertAll() at the end in order to collect all assertions & mark the case as fail.

How to wait for a page to load in Selenium?

It is important to wait for a page to load in Selenium before interacting with the web elements in order to avoid NoSuchElementException.

- Using document.readyState API for checking on whether a page has loaded.

```
((IJavaScriptExecutor)driver).ExecuteScript("return
document.readyState").Equals("complete"));
```

- Using Explicit wait condition,

```
(new WebDriverWait(getDriver(),
10)).until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector(".selector")));
```

How do you connect & run database queries via Test automation (Java)?

We need to use APIs which helps to interact with database like JDBC [Java Database Connectivity] – a Java API used to connect and interact with Database.

Before building a connection, make sure to add connector dependency in your POM [Ex. MySQL Connector].

DriverManager – to establish a connection | Connection – to create a Statement object | Statement – to execute the query | Resultset – to store the query results.

- a. Load & register the driver | Class.forName("com.mysql.jdbc.Driver");
- b. Establish connection | Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/Employee","root", "root");
- c. Create a Statement object | Statement st= con.createStatement();
- d. Execute a query & get the Resultset | Resultset rs = st.executeQuery("Select * from Employee");

Note: PreparedStatement [instead of Statement] has methods to bind various object types, including files and arrays.

How to read config file?

.properties files are used in Java programs to maintain project configuration data, database config or project settings, URLs & credentials, etc. Each parameter is stored as a key-value pair on one line.

To read the config file, object of Properties class is used, provided by Java itself.

```
Properties CONFIG = new Properties();
FileInputStream fs = new FileInputStream(PropertyPath);
CONFIG.load(fs);
String driverPath = properties.getProperty("driverPath");
```

The benefit: If any information is changed from the properties file, you don't need to recompile the java class. We can configure things that are prone to change over a period without need of changing anything in code.

TestNG Listeners

Listener: interface that listen to a specific incident in the selenium script and behave accordingly, used for logging purposes and creating reports. Some of the popular TestNG listeners,

- **ITestListener**: listens to specific events (depending on its methods) and executes the code written inside the method. Can also log the events onto the reports. Methods: OnStart | onTestSuccess | onTestFailure | onTestSkipped | onTestFailedButWithinSuccessPercentage | onFinish
- **IReporter**: medium to generate custom reports, contains a method called generateReport() which is invoked when all the suites have run.
- **ISuiteListener**: works on the suite level, listens to the start and end of a suite execution. Methods: onStart | onFinish.

Note: 'ITestResult [interface that describes the result of the test] and 'ITestContext' [contains all the information about the test run] are generally used when implementing ITestListener.

Selenium Actions vs Java AWT Robot class

The basic & major difference - Selenium uses the WebDriver API and sends commands to a browser to perform actions (through the "JSON wire protocol"). Whereas Java AWT Robot uses native system events to control the mouse and keyboard.

Since Robot class uses the native system events, it will actually move the mouse cursor instead of just generating Mouse Event.

Actions: User-facing API that allow you to build a chain of actions and perform them which is based on the WebDriver API. It just mimics the keyboard & mouse actions on a browser instead of actually moving the cursor.

Note: It is always preferred to use Actions class rather than using the Robot class (Keyboard or Mouse directly). Say, you have parallel execution running – Actions object will be tied to specific driver instance, whereas Robot events will be performed on whatever window is open.

Selenium: How can we switch from one window to another?

- String currentWinHandle = driver.getWindowHandle(): to get the window handle of the current window.
- Set<String> allWinHandles = driver.getWindowHandles(): to get the window handles of all the open windows in a Set.
- driver.switchTo().window(winHandle): to switch to other window

Complete Code:

```
String parentWindow = driver.getWindowHandle();
Set<String> handles = driver.getWindowHandles();
For (String windowHandle : handles)
{
    if(!windowHandle.equals(parentWindow))
    {
        driver.switchTo().window(windowHandle);
        <!--Perform your operation here for new window-->
        driver.close(); //close child window
        driver.switchTo().window(parentWindow); //control back to parent window
    }
}
```

How to assert that an image is properly loaded?

a. Using JavaScript executor (to check if image width is > 0)

```
WebElement ImageFile = driver.findElement(By.xpath("//img[contains(@id,'Test
Image')]"));
Boolean ImagePresent = (Boolean) ((JavascriptExecutor)driver).executeScript("return
arguments[0].complete && typeof arguments[0].naturalWidth != \"undefined\" &&
arguments[0].naturalWidth > 0", ImageFile);
if (!ImagePresent)
    System.out.println("Image not displayed.");
else
    System.out.println("Image displayed.");
```

b. Using isDisplayed method

```
if (driver.findElement(By.xpath("//img[contains(@id,'Test Image')]")).isDisplayed())
    System.out.println("Image not displayed.");
else
    System.out.println("Image displayed.");
```

What is iframe?

The <iframe> tag (inline frame) defines a rectangular region within the HTML document in which the browser can display a separate document, including scrollbars and borders. An inline frame is used to embed another document within the current HTML document. Often used to insert content from another source, such as an advertisement, into a Web page.
Example,

```
<html>
  <head>
    <title>HTML Iframes</title>
  </head>
  <body>
    <p>Document content goes here...</p>
    <iframe src = "/html/menu.htm" width = "555" height = "200">
      This text is present in an iFrame.
    </iframe>
    <p>Document content also go here...</p>
  </body>
</html>
```

Difference between relative and absolute path? Which one is fast?

- **Absolute Xpath:** It uses Complete path from the Root Element to the desire element.
- **Relative Xpath:** You can simply start by referencing the element you want and go from there.

Relative Xpaths are always preferred as they are not the complete paths from the root element. (//html//body). In future, if any web element is added/removed, then the absolute Xpath changes. So always use Relative Xpaths in your Automation.

Example,

```
<html>
  <body>
    <input type ="text" id="username">
  </body>
</html>
```

Absolute: html/body/input | **Relative:** //*[@id="username"]

How to travel through siblings in Selenium. Syntax for this.

Sibling, i.e., brother or sister. What does it mean in HTML DOM context? Elements at the same level within the DOM structure. We can use either preceding-sibling or following-sibling to traverse up & down at the same DOM level.

- //button[contains(.,'Arcade Reader')]/following-sibling::button[@name='settings']
- //button[contains(.,'Arcade Reader')]/preceding-sibling::button[@name='settings']

#Tip: Last Round rejection is unfortunate, but move-on.

Sometimes people do get a message saying ‘rejected in last round’ or ‘Candidature is on hold’. It is unfortunate. It is rare that a recruiter comes back after the candidature is put on hold. But don’t get upset. At least you got an interview call and appeared. Just move on, keep learning and never settle until you join the next firm.

If the recruiter comes back, take it as a surprise & be happy!

#Tip: Listen

One of the most crucial interview tips: Listen. From the very beginning of the interview, your interviewer is giving you information, either directly or indirectly. If you are not hearing it, you are missing a major opportunity. Good communication skills include listening and letting the person know you heard what was said. Observe your interviewer, and match that style and pace.

Additionally, ask questions after listening to demonstrate your analytical skills about a problem at hand.

How to fetch data from property file?

'.properties' files are mainly used in Java programs to maintain project configuration data, database config or project settings etc. Each parameter in properties file are stored as a pair of strings, in key and value format, where each key is on one line. You can easily read properties from some file using object of type Properties.

```
File file = new File("D:/SoftwareTestingStudio/ReadData/src/datafile.properties");
FileInputStream fileInput = new FileInputStream(file);
Properties prop = new Properties();
prop.load(fileInput);
WebDriver driver = new FirefoxDriver();
driver.get(prop.getProperty("URL"));
driver.findElement(By.id("Email")).sendKeys(prop.getProperty("username"));
driver.findElement(By.id("Password")).sendKeys(prop.getProperty("password"));
driver.findElement(By.id("SignIn")).click();
```

What is the difference between @beforetest and @beforemethod?

- **@BeforeTest**: runs before each test class (can contain multiple @Test) declared inside testng.xml
- **@BeforeMethod**: runs before each @Test method declared within a class.

Say you have a class HomePageTest.java where you have 3 @Test methods,

- check application Login
- check Homepage title
- check Homepage elements

Now you give this HomePageTest.java in your testng.xml.

- **@BeforeTest**: code will run once before HomePageTest.java and then before the next test mentioned in testng.xml
- **@BeforeMethod**: code will run before each @Test annotation within your HomePageTest.java file, i.e., 3 times in this case.

Difference between Xpath and CSS?

- The primary difference - with XPath we can traverse both forward and backward whereas CSS selector only moves forward.
- CSS selectors perform far better than Xpath.
- Xpath engines are different in each browser, hence make them inconsistent.
- IE does not have a native Xpath engine, therefore selenium injects its own Xpath engine for compatibility of its API. Thus, losing the advantage of using native browser features.

However, there are some situations where, you need to use Xpath, like while searching for a parent element or searching element by its text.

How to execute testng.xml from command prompt?

```
cd C:\Selenium\SampleTestNG (Project path)
java -cp C:\Selenium\SampleTestNG\lib\*;C:\Selenium\SampleTestNG\bin org.testng.TestNG
testng.xml
```

For 'java -cp' argument - provide the classpath i.e. path to classes/libraries that the program requires to run.

Explain Selenium Grid concept and commands used for it.

Selenium Grid allows us to execute our tests in multiple machines (physical / virtual) and multiple browsers with different versions, which dramatically speeds up test execution and helps in reducing total amount of time required for test execution.

A grid consists of a single hub, and one or more nodes which can be configured by specifying command line parameters OR by specifying a JSON config file.

- **Hub:** the central point which will receive all the test requests along with information on which browser, platform (i.e. WINDOWS, LINUX, etc) and where the test should be run. Based on the request received, it will distribute them to the registered nodes. To start a hub with default parameters,

```
java -jar selenium-server-standalone-2.48.2.jar -role hub
```

After starting the hub, we can view the status of the hub by opening any browser window and navigating to: <http://localhost:4444/grid/console>.

- **Nodes** are where our tests will run, each Node is machine (can be a physical machine / virtual machine) that we register with the Hub (browser and configuration details).

```
java -jar selenium-server-standalone-2.48.2.jar -role node -hub
http://localhost:4444/grid/register
```

Explain web driver hierarchy.

1. **SearchContext (interface):** The topmost interface of Webdriver which contains only two abstract method findElement() and findElements(). These two methods don't have a method body.
2. **WebDriver (interface)** extends SearchContext interface - many abstract methods like close(), get(java.lang.String URL), quite(), navigate(), switchTo() and others.

3. **RemoteWebDriver (fully implemented class)** - all abstract methods of WebDriver and SearchContext interface implemented. Also two other interfaces JavascriptExecutor and TakesScreenshot abstract methods are implemented in RemoteWebDriver class.
4. Finally, **browser-specific driver classes** like FirefoxDriver, ChromeDriver, IEDriver, SafariDriver, etc.

Why we are writing Web Driver driver = new ChromeDriver(). Why can't we write ChromeDriver() cd = new ChromeDriver();

We generally do it this way because we want to be able to run our tests on multiple browsers. If we declare the driver as a specific driver type, we are then anchored to only that driver. This is not a problem if you only ever need to test on say Chrome for example. But what if you later want your tests to also be able to work with IE, Opera, Firefox, etc.?

First,

- WebDriver is an interface.
- FirefoxDriver()/ChromeDriver() is inheriting RemoteDriver class which implements the WebDriver interface.

We cannot create the WebDriver object, because WebDriver is an interface and not a class.

It is possible to create an object for an interface and instantiate it using any of the classes that implements the interface like this: WebDriver driver = new FirefoxDriver(); OR WebDriver driver = new ChromeDriver(); By using above code, your scripts are now flexible and can use any WebDriver object which is required to invoke particular browser.

If your main tests and other classes define the reference variable of type WebDriver, it allows us to assign the driver object to different browser specific drivers (without change to the test code itself). Thus, allowing multi-browser testing by assigning the driver object to any of the desired browser.

CacheLookUp concept in PageFactory?

When PageFactory initializes a WebElement which is decorated with @FindBy or @FindAll annotation, it creates a Java proxy object, i.e. delay load elements and avoid FindElementBy call during Page Object initialization.

Proxy object has to resolve the actual WebElement to make any call,

- Find the element every time you need it (Which is time consuming because of a FindElement REST call to WebDriver)

- Cache the element from the very first FindElement call to WebDriver and return it in subsequent calls. Note: Not a good idea for elements which are dynamic because referring to an older cached version will result in Stale Element exception.

If you use @CacheLookup annotation, you can instruct Selenium to NOT make a FindElement call to Browser's WebDriver every time and rather used the earlier cached version.

How to check background color in Selenium?

Color can be determined using the CSS value of the element,

- driver.findElement(By.xpath(..blah..")).getCssValue("background-color").toString();
- driver.findElement(By.xpath(..blah..")).getCssValue("color").toString();

How to handle Proxy in Selenium?

Sometimes when you try to access a secure application you might get proxy issues. Until we do not set proxy, we cannot access the application. To handle proxy setting in Selenium - we have a separate class called Proxy.

- Create object of proxy class and set HTTP proxy or FTP proxy based on requirement.

```
Proxy p=new Proxy();
p.setHttpProxy("localhost:7777");
```

- Use DesiredCapability class to customize capability of browser and pass the proxy object.

```
DesiredCapabilities cap=new DesiredCapabilities();
cap.setCapability(CapabilityType.PROXY, p);
```

- While initiating browser pass capability object

```
WebDriver driver=new FirefoxDriver(cap);
```

How to open chrome browser in Incognito Mode?

Add argument "-incognito" in the DesiredCapabilities,

```
ChromeOptions options = new ChromeOptions();
options.addArguments("--incognito");
DesiredCapabilities capabilities = DesiredCapabilities.chrome();
capabilities.setCapability(ChromeOptions.CAPABILITY, options);
```

#Tip: Look for patterns during your Job Search.

If you keep getting the same results and the same feedback, consider that the issue might be your approach. Don't fret! Just think of what YOU can change or adjust or improve.

Results won't change if you keep doing the same thing. "Not getting interview call" – try tweaking your notice period or expected CTC or resume keywords etc. to check the results. "Not clearing technical interview" – read & learn more about the interview Q&As. "Getting rejected in further rounds" – communication might be a problem.

Just like everything else, job search will also have patterns for you to find & improve upon.

#Tip: Make an impact

It's not always what you say, but how you present yourself that makes an impression. Be sure to shake hands firmly and maintain eye contact. During the interview, sit up and stay focused. If your mind starts to wander, it shows.

If we are using Actions class then what is the usage of Robot class?

There is a difference in terms of how do these two works.

- Selenium Actions uses the WebDriver API and sends commands to a browser to perform actions (through the "JSON wire protocol").
- Java AWT Robot uses native system events to control the mouse and keyboard.

If you are doing browser automation, ideally, you don't ever use things like Robot since usually the functionality provided by selenium is more than enough. If there is a technical glitch using actions class in certain environments, then we can use Robot class.

How to handle untrusted certificates in Selenium?

Firefox,

- Create a firefox Profile
- Set the setAcceptUntrustedCertificates() method as true and setAssumeUntrustedCertificateIssuer() method to false

```
FirefoxProfile profile=new FirefoxProfile();
profile.setAcceptUntrustedCertificates(true);
WebDriver driver=new FirefoxDriver(profile);
driver.get("STS URL");
```

Chrome & IE,

Take the help of 'Desired capabilities; to get and accept the SSL certificate error on run time.

- Create An object for Desiredcapabilities class
- Set the ACCEPT_SSL_CERTS as true
- Open the chrome browser with the capability

```
DesiredCapabilities cap=DesiredCapabilities.chrome();
cap.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);
System.setProperty("webdriver.chrome.driver","Chrome driver path");
WebDriver driver=new ChromeDriver(cap);
```

Selenium: How do you handle a dynamic drop-down list?

The most common example - Only after city selection is made > the city options are loaded in the next drop-down list.

Need to handle using the text value to click on, e.g.

```
dropdown.click();
List<WebElement> options = dropdown.findElements(By.tagName("li"));
for (WebElement option : options)
{
    if (option.getText().equals(searchText))
    {
        option.click(); // click the desired option
        break;
    }
}
```

How to handle dynamic elements in Selenium?

Should use JavaScript functions like “starts-with” or “contains” in our element locators to separate the dynamic part of locator from static part.

For example, XPath: //input[contains(@id, '_name')]

**How do you handle a flash which comes up in the website in between like it is a flash file.
How do you check whether it is a flash or not a flash in selenium?**

Need to use some third-party libraries such as Sikuli to automate flash objects.

In a Drop-down, there are several options, and I need to select HYD, how to achieve using Webdriver?

For handling dropdowns, Selenium already provides Select class that has some predefined method which help is a lot while working with Dropdown.

Select city=new Select(driver.findElement(By.id("city")))

- City.selectByIndex(4)

Index starts from 0. Select a value depending on the index, i.e., index as 4, it will select 5th value.

- city.selectByValue("3")

Select a value depending on the ‘value’ tag of the given element.

- city.selectByVisibleText("HYD");

Select the value from the visible text. Straight forward – whatever text we are passing it will simply select that value.

- Get All option from dropdown

```
List<WebElement> dropdown=city.getOptions();
for(int i=0;i<dropdown.size();i++)
{
    System.out.println(dropdown.get(i).getText());
}
```

Type of locators in Selenium

The locator can be termed as an address that identifies a web element uniquely within the webpage.

ID | ClassName | Name | TagName | LinkText | PartialLinkText | Xpath | CSS Selector | DOM

What is the difference between “/” and “//” in Xpath?

- **Single Slash “/”:** used to create Xpath with absolute path i.e. the xpath would be created to start selection from the document root node/start node.
- **Double Slash “//”:** Double slash is used to create Xpath with relative path i.e. the xpath would be created to start selection from anywhere within the document.

Types of wait available in WebDriver

- **Implicit Wait:** used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, the subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

```
driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);
```

- **Explicit Wait:** used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.

```
WebDriverWait wait = new WebDriverWait(WebDriver Reference, TimeOut);
```

- **Fluent wait:** used to wait for a condition, as well as the frequency with which we want to check the condition.

```
Wait wait = new FluentWait(WebDriver reference).withTimeout(timeout,  
SECONDS).pollingEvery(timeout, SECONDS).ignoring(Exception.class);
```

How can we get a text of a web element?

getText() is used to retrieve the inner text of the specified web element. The command doesn't require any parameter but returns a string value. It is also one of the extensively used commands for verification of messages, labels, errors etc. displayed on the web pages.

```
String text = driver.findElement(By.id("Text")).getText();
```

How to select value in a dropdown?

Using WebDriver's Select class.

selectByValue:

```
Select selectByValue = new Select(driver.findElement(By.id("SelectID_One")));  
selectByValue.selectByValue("greenvalue");
```

selectByVisibleText:

```
Select selectByVisibleText = new Select (driver.findElement(By.id("SelectID_Two")));  
selectByVisibleText.selectByVisibleText("Lime");
```

selectByIndex:

```
Select selectByIndex = new Select(driver.findElement(By.id("SelectID_Three")));  
selectByIndex.selectByIndex(2);
```

findElement() and findElements()

- **findElement():** used to find the first element in the current web page matching to the specified locator value. Take a note that only first matching element would be fetched.

```
WebElement element = driver.findElement(By.xpath("//div[@id='example']//ul//li"));
```

- **findElements():** used to find all the elements in the current web page matching to the specified locator value. All the matching elements would be fetched and stored in the list of WebElements.

```
List <WebElement> elementList =  
driver.findElements(By.xpath("//div[@id='example']//ul//li"));
```

driver.close() and driver.quit command

- **close():** closes the web browser window that the user is currently working on OR currently being accessed by the WebDriver. The command neither requires any parameter nor does it return any value.
- **quit():** Unlike close() method, quit() method closes down all the windows that the program has opened. Neither requires any parameter nor does it return any value.

How can we handle web-based pop-up?

WebDriver offers the users a very efficient way to handle these pop-ups using Alert interface. There are four methods that we would be using along with the Alert interface.

- void dismiss(): clicks on the “Cancel” button as soon as the pop-up window appears.
- void accept(): clicks on the “Ok” button as soon as the pop-up window appears.
- String getText(): returns the text displayed on the alert box.
- void sendKeys(String stringToSend): enters the specified string pattern into the alert box.

```
Alert alert = driver.switchTo().alert();
alert.accept();
```

How can we handle windows-based pop up?

Selenium is an automation testing tool which supports only web application testing, that means, it doesn't support testing of windows-based applications. However, with some third-party intervention, this problem can be overcome. There are several third-party tools available for handling window-based pop-ups along with the selenium like AutoIT, Robot class etc.

How to retrieve CSS properties of an element?

Using getCssValue() method:

```
driver.findElement(By.id("id")).getCssValue("color");
driver.findElement(By.id("id")).getCssValue("font-size");
```

How to capture screenshot in WebDriver?

TakeScreenshot interface can be used to take screenshots in WebDriver. getScreenshotAs() method is used to save the screenshot.

```
import org.openqa.selenium.TakesScreenshot;
public void test() throws IOException
{
    File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
    FileUtils.copyFile(scrFile, new File("C:\\CaptureScreenshot\\google.jpg"));
}
```

Navigation commands

- driver.navigate().to("https://www.softwaretestingstudio.in/");
- driver.navigate().refresh();
- driver.navigate().forward(); --- just like clicking on the Forward Button of any browser.
- driver.navigate().back(); --- just like clicking on the Back Button of any browser.

How to scroll down a page using JavaScript?

scrollBy() method is used to scroll down the webpage | executeScript("window.scrollBy(x-pixels,y-pixels)");

```
JavascriptExecutor js = (JavascriptExecutor) driver;
driver.get("https://www.softwaretestingstudio.com");
js.executeScript("window.scrollBy(0,1000)");
```

How to mouse hover over a web element?

Actions class utility is used to hover over a web element in Selenium WebDriver

```
Actions action = new Actions(driver);
action.moveToElement(driver.findElement(By.id("element-id"))).perform();
```

Is there a way to type in a textbox without using sendKeys()?

Yes! Text can be set using JavaScriptExecutor

```
JavascriptExecutor jse = (JavascriptExecutor) driver;
jse.executeScript("document.getElementById('email').value='abc.efg@xyz.com'");
```

How to upload a file in Selenium WebDriver?

Using sendkeys() or Robot class method. Locate the text box and set the file path using sendkeys() and click on submit button.

```
WebElement browse =driver.findElement(By.id("uploadfile"));
browse.sendKeys("D:\\STS\\UploadFile.txt");
```

#Tip: Manual Testing is NOT dead

First, Manual Testing is NOT dead. We as testers need to understand the difference between 'Manual Testing' and 'Skills expected from Testers'. Manual Testing, Test Automation, Database Testing, Performance Engineering, API Testing, etc. are all important. Just that earlier all these were exclusive skills. With agile, the expectation from a Tester is to be proficient with multiple Testing skills. It can be Manual + Automation, Or ETL + Automation, Functional + API + Performance, etc. The more skills = the better opportunities.

Depending upon your interest - you can opt for RPA automation, Performance, API Automation, ETL Testing, Product Testing (e.g., SFDC, AEM, SAP, etc.), etc.

It is all about adding more and more skills to your bucket - because that's where industry is heading - a full-stack QA-Tester!

How to login to any site if it is showing an Authentication Pop-Up for Username and Password?

To handle authentication pop-ups, verify its appearance and then handle them using an explicit wait command.

```
WebDriverWait wait = new WebDriverWait(driver, 10);
Alert alert = wait.until(ExpectedConditions.alertIsPresent());
alert.authenticateUsing(new UserAndPassword(<username>, <password>));
```

Explain how you can switch between frames?

Use driver.switchTo().frame() to switch between frames. By using,

- A number: It selects the number by its (zero-based) index
- A name or ID: Select a frame by its name or ID
- Previously found WebElement: Using its previously located WebElement select a frame

For Database Testing in Selenium Webdriver what API is required?

We need JDBC (Java Database Connectivity) API. It allows us to execute SQL statements.

Write a code to wait for a element to be visible on a page.

We can write a code such that we specify the XPath of the web element that needs to be visible on the page and then ask the WebDriver to wait for a specified time.

```
WebDriverWait wait=new WebDriverWait(driver, 20);
Element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(" <xpath> ")));
```

How to scroll down to a particular element?

To scroll down to a particular element on a web page, we can use the function scrollIntoView().

```
((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView();", element);
```

How to handle multiple windows in Selenium?

A window handle is a unique identifier that holds the address of all the windows. This is basically a pointer to a window, which returns the string value.

- `get.windowhandle()`: helps in getting the window handle of the current window.
- `get.windowhandles()`: helps in getting the handles of all the windows opened.
- `set`: helps to set the window handles which is in the form of a string.
- `switch to`: helps in switching between the windows.
- `action`: helps to perform certain actions on the windows.

Common Exceptions in Selenium WebDriver

- **ElementNotVisibleException**: Although an element is present in the DOM, it is not visible (cannot be interacted with). E.g. Hidden Elements – defined in HTML using `type="hidden"`.
- **ElementNotSelectableException**: Although an element is present in the DOM, it may be disabled (cannot be clicked/selected).
- **InvalidSelectorException**: Selector used to find an element does not return a WebElement. Say XPath expression is used which is either syntactically invalid or does not select WebElement.
- **NoSuchElementException**: WebDriver is unable to identify the elements during run time, i.e. `FindBy` method can't find the element.
- **NoSuchFrameException**: WebDriver is switching to an invalid frame, which is not available.
- **NoAlertPresentException**: WebDriver is switching to an invalid alert, which is not available.
- **NoSuchWindowException**: WebDriver is switching to an invalid window, which is not available.
- **StaleElementReferenceException**: The referenced element is no longer present on the DOM page (reference to an element is now Stale). E.g. The Element belongs to a different frame than the current one OR the user has navigated away to another page.
- **SessionNotFoundException**: The WebDriver is performing the action immediately after 'quitting' the browser.
- **TimeoutException**: The command did not complete in enough time. E.g. the element didn't display in the specified time. Encountered when working with waits.
- **WebDriverException**: The WebDriver is performing the action immediately after 'closing' the browser.

#Tip | Market Yourself

Undoubtedly, there is too much competition out there. The jobs market is too competitive. So how do you stand out? How will a recruiter or interviewer know that you are good? Just from the 2-3 hours of interview? Nah!

Market yourself. Build a LinkedIn profile. Share content. Blog. Network. Publish. Anything that sets you apart from normal resume + interview. It will work wonders in case interviewer has somewhere read about you or your content. Ice-breaker!

What is Selenium “WebElement”?

WebElement is a class that represents a DOM element. E.g., input element, select element, div element, hyperlink element, etc. WebElements can be found by searching from the document root using a WebDriver instance [absolute XPath], or by searching under another WebElement [relative XPath].

WebDriver API provides built-in methods to find the WebElements which are based on different properties like ID, Name, Class, XPath, CSS Selectors, link Text, etc.

How to handle keyboard operations in Selenium using Robot class?

Robot class is a part of the Java API awt package and NOT org.openqa.selenium.

Unlike Selenium which uses the WebDriver API to invoke commands to a browser to perform actions, Robot class uses native system events to control the mouse and keyboard.

E.g., need to handle a window-popup or a combination of modifier keys such as Alt, Shift, etc.

It differs from Selenium which uses the WebDriver API and invokes commands to a browser to perform actions.

Steps: import java.awt.Robot >> Robot robot = new Robot(); >>
robot.<method_name>(parameter);

Keyboard Methods,

- **keyPress(int keycode):** press a given key. say Alphabet A = KeyEvent.VK_A | i.e.,
keyPress(KeyEvent.VK_A);
KeyEvent: a low-level event. In Java AWT, low-level events are events that indicate direct communication like a keypress, key release.
- **keyRelease(int keycode):** releases a given key. E.g., the Shift key pressed using the keyPress(KeyEvent.VK_SHIFT) method needs to release using the keyRelease (KeyEvent.VK_SHIFT) method.

What is Selenium - Actions class? And why do we need it?

Why? Because there are complex interactions like Drag-n-Drop and Double-click etc. which cannot be done by simple WebElement commands. To handle those, we have the Actions class in Selenium.

“The user-facing API for emulating complex user gestures. Use this class rather than using the Keyboard or Mouse directly. Implements the builder pattern: Builds a Composite Action containing all actions specified by the method calls”.

Build method: build the sequence of actions using the build() method and get the composite action. And finally, perform the actions sequence using perform() method.

E.g.

```
Actions actions = new Actions(webdriver object);
actions.keyDown(element,Keys.SHIFT).sendKeys(TextToBeConvertAndSendInUpperCase).keyUp(Keys.SHIFT).build().perform();
```

Note: Actions vs Action | “Action is an Interface”, used to represent the single user interaction to perform the series of action items build by “Actions class”.

What are the different methods available in Selenium - Actions class?

Keyboard Events:

- keyDown(modifier key): Performs a modifier key press.
- sendKeys(keys to send): Sends keys to the active web element.
- keyUp(modifier key): Performs a modifier key release.

Mouse Events

- click(): Clicks at the current mouse location.
- doubleClick(): Performs a double-click at the current mouse location.
- contextClick() : Performs a context-click at middle of the given element.
- clickAndHold(): Clicks (without releasing) in the middle of the given element.
- dragAndDrop(source, target): Click-and-hold at the location of the source element, moves to the location of the target element
- dragAndDropBy(source, xOffset, yOffset): Click-and-hold at the location of the source element, moves by a given offset
- moveByOffset(x-offset, y-offset): Moves the mouse from its current position (or 0,0) by the given offset
- moveToElement(toElement): Moves the mouse to the middle of the element
- release(): Releases the depressed left mouse button at the current mouse location

#Tip | Narrate your real-project experience

Interviewers love reality. Now-a-days every interview will have a set of situational questions. And it is good if you narrate a real-project experience to answer that. Say how did you handle team conflict? Or What if timelines are too challenging? Or ever had a conflict with your manager? Or how to handle a non-performing team member?

For these types of Qs, it is advisable to narrate your real-project experience. E.g., “Yeah, once we had a project where timelines were too strict due to regulatory compliance. First, we did setup effective communication channels between teams, utilized test automation for regression runs, kept an eye on defects turn-around time, had weekly goals, etc. All these efforts helped us to achieve on-time delivery.”

Your personal account of situation [problem – actions - results] is always much better than thinking hypothetically. I.e., “We did this” is much better than “We can do this” 😊

Can we prioritize Test Cases in Selenium test automation? If Yes, how?

Yes. How? Using TestNG – use the priority argument along with @Test annotation.

```
@Test (priority = 0)
public void method1()
{
    //test code
}
@Test (priority = 1)
public void method2()
{
    //test code
}
```

Few pointers,

- Priority can only be passed for @Test annotated methods.
- Lower the priority number; higher is the priority. i.e., 0 will be executed before 1.
- If two or more methods have the same priorities in TestNG, then their running test sequence is alphabetic.
- The test methods with no priority assigned have a default priority equal to 0.

What is the use of JavaScriptExecutor?

JavascriptExecutor is the Selenium interface which is implemented by all the driver classes.

Why? While you execute your Selenium script - at times because of cross domain policies browsers enforce, your script execution may fail unexpectedly. Sometimes web controls don't react well against Selenium commands. Say an element is not interactable, or not visible on webpage (scroll), etc.

Methods,

- executeScript(): executes JavaScript in the context of the currently selected frame or window.
- executeAsyncScript(): executes an asynchronous piece of JavaScript in the context of the currently selected frame or window.

Syntax,

```
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript(Script, Arguments);
```

Best Practise: If you see too many issues due to element not interactable,

1. Catch the exception, take screenshot for your reference, generate a warning in logs.
2. Implement direct click using Javascript executor in catch block.
3. By generating warning, you can check if you are not missing out on an actual issue.

How do you manage browser-driver compatibility, i.e., compatible versions? Say browser gets an auto-update and your scripts start failing due to older driver version.

Problem: To run test automation, we need to have browser drivers, .exe files like chromedriver.exe and geckodriver.exe [windows] OR binary files like chromedriver and geckodriver [Linux], which allows Selenium WebDriver to handle browsers. Also, we need to set the path of these files in our script OR need to add the location to the classpath. But what if some browser version is auto-updated? These steps become cumbersome as we need to carry them out every time the versions change.

Solution: Hence, we use the “WebDriverManager” class in Selenium, to download the binary file (or .exe files) of the driver automatically based on the browser version.

```
WebDriverManager.chromedriver().setup();
driver = new ChromeDriver();
```

- automates the management of WebDriver binaries.
- downloads the appropriate driver binaries, if not already present, into the local cache.
- downloads the latest version of the browser binary, unless otherwise specified.
- Eliminates the need to store driver binaries locally. We also need not maintain various versions of the binary driver files for different browsers.

All we have to do is to add its dependency through Maven or Gradle to download all the necessary drivers.

Selenium: How to run only failed cases?

Method-1: TestNG - IRetryAnalyzer

Create a separate JAVA class implementing TestNG - IRetryAnalyzer interface. And need to override ‘retry’ method [to define the retry count] taking < ITestResult result > as argument. Change your @Test annotations with @Test(retryAnalyzer = RetryAnalyzer.class).

Method-2: Using testng-failed.xml file

After every execution a testing-failed.xml file is created which keeps track of all the failed tests. We can run this file just like we run the testng.xml file.

How to run parallel tests using TestNG?

Process of running the test scripts parallelly rather than one after the other.

- Reduces Time: Running the tests in parallel reduces the overall execution time.
- Allow Multi-Threaded Tests: We can allow multiple threads to run simultaneously.

How? with the help of keyword “parallel”, and assign any of the four values – Methods/Tests/Classes/Instances.

```
<suite name = "Parallel Testing Suite">
<test name = "Parallel Tests" parallel = "methods">
<classes>
<class name = "ParallelTest" />
</classes>
</test>
</suite>
```

The TestNG has a default value of thread = 5 for parallel testing which we can modify,

```
<test name = "Parallel Tests" parallel = "methods" thread-count = "2">
```

Note: You need to be aware of dependent test scripts, to avoid any conflicts while parallel execution.

What is Headless testing? And how can we achieve it via Selenium?

Problem: When we run the Selenium tests on any of the browsers, we generally face some challenges such as slow rendering on the browser, interference of other applications running on the system, etc.

Solution: Headless browser helps in the execution of the Selenium Headless Browser tests in a Non-UI mode. Almost all modern browsers provide the capability to run them in headless mode.

HTMLUnitDriver

HTMLUnitDriver is an implementation of Selenium WebDriver [similar to FirefoxDriver, ChromeDriver, etc.] based on HtmlUnit, which is a Java-based implementation of a web browser without a GUI.

- Add the driver dependency
- Initialize: `HTMLUnitDriver unitDriver=HtmlUnitDriver(true)` **OR** `HTMLUnitDriver driver = new HTMLUnitDriver(BrowserVersion.FIREFOX_68)` **OR** `HtmlUnitDriver unitDriver = new HtmlUnitDriver(BrowserVersion.FIREFOX_68,true);`

Headless Chrome

Selenium provides a class ‘ChromeOptions’ where we can specify certain configurations to change the default behaviour of Chrome – one configuration is the “headless” mode.

```
ChromeOptions options = new ChromeOptions()  
options.addArgument("headless");  
ChromeDriver driver = new ChromeDriver(options);
```

Note: Debugging will not be feasible, as the only way to check what's running on the browser is to grab the screenshots and validate the output.

TestNG: How can we group test cases? Say, Sanity, Regression, etc.?

Using **GROUPS** attribute in TestNG, we can assign the test methods to different groups.

```
@Test(groups = {"Sanity"})  
public void testMethod1()  
{  
    //Test logic  
}  
  
@Test(groups = {"Regression"})  
public void testMethod2()  
{  
    //Test logic  
}
```

TestNG: What are some common assertions provided by TestNG?

- assertEquals (String actual, String expected, String message) and other overloaded data types in parameter
- assertNotEquals (double data1, double data2, String message) and other overloaded data types in parameter
- assertFalse (boolean condition, String message)
- assertTrue (boolean condition, String message)
- assertNotNull (Object object)
- fail (boolean condition, String message)
- true (String message)

TestNG: How can we make one test method dependent on other using TestNG?

Using **dependsOnMethods** parameter inside @Test annotation, i.e., test method will run only after the successful execution of dependent test method.

```
@Test(dependsOnMethods = { "preTests" })
```

#Tip | Negotiate intelligently

Salary negotiation is important. Sometimes you might get underpaid & fret later OR other times you might not get a call/offer because of your high ECTC.

- *Quickly check the 'Salary range' offered by the prospective company. Glassdoor is an easy option.*
- *Always quote ECTC as 'Open for Discussion' in your initial interactions.*
- *Don't ask for unrealistic hikes & jumps. Know your worth in advance.*
- *Questions you can ask the HR before telling your numbers – what is the budget for this position OR what is the average CTC in your company at this level.*
- *When insisted, never quote a 'number'. Instead "It should be a win-win situation wherein I am at par with my colleagues & motivated to put in my 120%" OR a range of percentage hike you are expecting like 30-35%, 35-40%, etc.*

TestNG: How to run a Test method multiple times in a loop?

Using **invocationCount** parameter and setting its value to an integer value.

```
@Test(invocationCount = 10)
public void invocationCountTest()
{
    //Test logic
}
```

TestNG: Any idea about threadPoolSize?

The **threadPoolSize** attribute specifies the number of threads to be assigned to the test method. This is used in conjunction with **invocationCount** attribute. The number of threads will get divided with the number of iterations of the test method specified in the **invocationCount** attribute.

```
@Test(threadPoolSize = 5, invocationCount = 10)
public void threadPoolTest()
{
    //Test logic
}
```

What is Cucumber?

BDD is becoming widely accepted practice in agile software development, and Cucumber-JVM is a mainstream tool used to implement this practice in Java. Cucumber-JVM is based on Cucumber framework, widely used in Ruby on Rails world as well as in Java and .Net.

Cucumber-JVM allows developers, QA, and non-technical or business participants to write features and scenarios in a plain text file using Gherkin language with minimal restrictions about grammar in a typical Given, When, and Then structure.

The feature file is then supported by a step definition file, which implements automated steps to execute the scenarios written in a feature file. Apart from testing APIs with Cucumber-JVM, we can also test UI level tests by combining Selenium WebDriver.

What are the advantages of using Cucumber?

- Cucumber supports a variety of programming languages, including Java, .net and Ruby.
- It serves as a link between commercial and technical language. This can be done by writing a test case in plain English text.
- It enables the test script to be developed without any prior knowledge of programming, as well as the participation of non-programmers.
- Unlike other tools, it functions as an end-to-end test framework.

Which language is used in Cucumber?

Cucumber understands Gherkin. It's a straightforward English representation of the app's functionality. It is used for defining test cases. It is intended to be non-technical and human-readable. It's a domain-specific (DSL), business-friendly language.

The Gherkin language uses several keywords to describe the behaviour of applications such as Feature, Scenario, Scenario Outline, Given, When, Then, etc.

Why do we need to use Cucumber with Selenium?

Cucumber is used in conjunction with Selenium because Cucumber makes the application flow easier to read and comprehend. The most important advantage of combining Cucumber and Selenium is that it allows developers to build test cases in simple feature files that managers, non-technical stakeholders, and business analysts can understand. It allows you to develop tests in Gherkin, a human-readable programming language.

Example of a BDD test in plain text?

Feature: Visit Jobs page in parikshram.com

Scenario: Visit parikshram.com

Given: I am on parikshram.com

When: I click on Jobs page

Then: I should see Jobs page

Files needed in the Cucumber framework?

- Feature File: It has plain text descriptions of single or numerous test situations. Keywords like Then, When, Background, Scenario Outline, Feature, And, But, and so on are used in the tests. As a result, it's a file that keeps track of features and their descriptions.
- Step Definition File: It has the extension .java. It essentially acts as a translator between the test scenario steps provided in the feature file and the automation code. Cucumber searches the step definition file and executes the relevant functions that are assigned to that step when it runs a step described in the feature file.
- TestRunner: .java is the file extension for this file. It connects the feature file and the step definition file. It allows the user to run one or more feature files at the same time. It contains the locations of the step definition and feature files.

Feature in Cucumber?

A project's feature can be described as a stand-alone unit or functionality. A list of scenarios to test for a feature is frequently included with it. The Feature File is a file in which we store features, descriptions of features, and situations to be evaluated. The first line of the feature file must start with the keyword 'Feature' followed by the description. A feature file may include multiple scenarios, and the extension of the feature file must be ".feature."

For an e-commerce website, we can have the following features:

- User registers and signs up on the website.
- User tries to log in to their account using their credentials.
- Users add a product to their cart.
- User clicks on checkout now.
- User pays for their items.
- User logs out from the website.

All these are different features. The website will have many such features. All these features will have a separate Feature File.

Feature file in Cucumber?

Features file contain a high-level description of the Test Scenario in simple language, Gherkin which is a plain English text language. Feature File consists of the following components like:

- Feature: It describes the current test script which has to be executed.
- Scenario: It is steps and expected outcome for a specific test case.
- Scenario outline: Scenario can be executed for multiple sets of data using scenario outline.
- Given: It specifies the context of the text to be executed.
- When: specifies the test action which has to perform.
- Then: Expected outcome of the test can be represented by “Then”

Step Definitions in Cucumber?

A step definition is the actual code implementation of the feature mentioned in the feature file. It connects Gherkin steps to programming code. The mapping between each step of the scenario defined in the feature file and a code of the function to be executed is stored in the steps definition file. A step definition carries out the action that should be performed by the step.

Step definition corresponding to the step “Open Chrome browser and launch the application” may look like the code below,

```
@Given("^Open Chrome browser and launch the application$")
public void openBrowser()
{
    driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("parikshram.com");
}
```

#Tip | Never quote your expected CTC on the first interaction.

Expected CTC is one of the bottlenecks for getting an interview call. If you are asking for a high jump – crash! No call! Try to avoid quoting the ECTC on your first interaction. How?

- *ECTC: Open for Discussion.*

Your current CTC gives recruiter an idea about your package range. And keeping ECTC open for discussion leaves room for discussions at the end.

TestRunner class in Cucumber?

In the Cucumber testing approach, the TestRunner class provides the link between the feature file and the step definition file. The TestRunner class is generally an empty class with no class definition.

```
Package com.sample.TestRunner
import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;
@RunWith(Cucumber.class)
@CucumberOptions(features="Features",glue={"StepDefinition"})
public class Runner
{
}
```

It is advised not to write code under the TestRunner class. It should include the tags @RunWith and @CucumberOptions.

What are the primary keywords in Cucumber?

- Feature: collect relevant scenarios and provide a high-level description of a feature.
- Rule: express a single business rule that should be followed. It adds to the information about a feature.
- Example: practical illustration of a business rule, i.e., a series of steps.
- Given: The given steps are used to describe the system's initial context - the scenario's scene.
- When: To describe an occurrence or an action. It could be a user interacting with the system or an event generated by another system.
- Then: To indicate an anticipated outcome, or result.
- Background: Helps to give the situations that follow it some context. It can have one or more Given steps, which are executed prior to each scenario but after any Before hooks.

Scenario in Cucumber?

Scenario is a fundamental Gherkin structure. Every scenario begins with the keyword "Scenario:" and ends with a scenario title. Every feature can have one or more scenarios, each of which has one or more steps.

Scenario: Verify 'My Orders' Functionality.

Explanation: When a user clicks on the My Orders option, he/she should be taken to the My Orders page.

#Tip: Practice common Q

You will benefit and gain confidence from having thought about the answers, and you may be able to apply them to questions that you didn't anticipate.

If you get a question you can't answer, simply say you don't know. Then say the question is something to which you would like to give more thought and that you are willing to learn what it takes. An employer will respect someone who is honest and open about his or her limitations.

#Tip | Prepare for 'common topics'

You are shortlisted for an interview. You know test automation is everywhere. You focus on test automation & prepare it well. During the interview, you are able to ace test automation related Qs pretty well but fall short on other common topics.

Now, what are these other common topics? First, TESTING concepts is the foundation. Second, most of the projects follow AGILE. Database verifications via SQL is another common skill. API Testing (either automation or not) is yet another common skill.

In chasing Test automation, don't neglect other common skills that are must-have for any QA-Tester. You can find these by looking at some of the job descriptions. Prepare well for these common topics.

Scenario Outline in Cucumber?

In Cucumber, a Scenario outline is used as a parameter of scenarios. This is used when the same scenario needs to be executed for multiple sets of data; however, the test steps remain the same. Scenario Outline must be followed by the keyword 'Examples', which specify the set of values for each parameter. The data is provided by a tabular structure separated by (| |).

Scenario Outline: Upload a file

*Given that the user is on upload file screen.
When a user clicks on the Browse button.
And user enters <filename> onto the upload textbox.
And user clicks on the enter button.
Then verify that the file upload is successful.*

Example:

```
|filename|
|file1|
|file2|
```

Examples keyword in the Cucumber framework?

We can achieve a data-driven approach in Cucumber with the help of the Examples keyword. The Scenario Outline in a feature file should be accompanied by the Examples part which consists of the multiple data set to be passed at the runtime.

Feature: New User Registration

Scenario Outline: Registration Verification Test

Given User navigates to Registration Page

Then User inputs "<Firstname>" and "<Lastname>" and "<Email>"

Examples:

Firstname	Lastname	Email
Deepanshu	Agarwal	abc@gmail.com
Kanchan	Kapoor	xyz@gmail.com

What is the use of the Options tag in the Cucumber Framework?

The Options tag is a part of the TestRunner file in the Cucumber framework, and it takes the form of an annotation named @CucumberOptions.

It has two parameters: glue and feature:

- Feature: The path to the feature file is specified by the feature option.
- Glue: The glue argument is used to provide the step definition file's location.

```

import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;
@RunWith(Cucumber.class)
@CucumberOptions (
features = "src/test/Sample/features",
glue = {"StepDefinitionFile"}
)
public class SampleTestRunner {
}

```

Annotations in Cucumber?

An annotation is a type of text that has been pre-defined and has a specified meaning. It tells the compiler/interpreter what to do when the program runs.

- Given: It specifies the requirements for running the test.
Given I have an account on Parikshram.
- When: It establishes the starting point for any test scenario.
When I log in to Parikshram.
- Then: It contains the expected result of the test which is to be executed.
Then registration should be successful.
- And: Between any two statements, it gives the logical AND condition. AND can be combined with the GIVEN, WHEN, and THEN statements.
When I enter my account number AND CVV.
- But: It denotes a logical OR relationship between two propositions. OR can be combined with the GIVEN, WHEN, and THEN statements.
Then I should be logged in BUT I must enter the OTP.

Hooks in Cucumber?

Hooks are code blocks that execute before or after each Cucumber scenario in the execution cycle. This enables us to better control the development workflow and decrease code redundancy. Setting up the web driver and terminating the web driver session resembles a test setup. When dealing with different scenarios, it's best to do the setup and clean up only once. Hooks are used to bring optimization.

Certain preconditions, such as executing the program, creating a database connection, preparing the test data, and so on, may be required in some cases. There are also several postconditions to be fulfilled, such as ending the database connection, closing the browser, refreshing test data, and logging out of the program. Cucumber handles all of these situations with the use of hooks.

The methods @Before and @After can be used to define hooks anywhere in the project or step definition layers. Before hook is executed before any other test situations, and after the hook is executed after all test scenarios have been completed.

#Tip | Resume, obviously important

Your resume is the key to getting shortlisted for an interview. It should reflect your actual skill set including the tools you have worked on. Ideally the professional summary should contain pointers that would interest recruiters/hiring managers. E.g., Tools, Domains, Testing types, Code repo, Blog, Planning, Process, Technology, roles & responsibilities, etc.

Recruiter generally search for keywords in a resume like – Automation, Performance, JMeter, Selenium, Java, Sanity, Regression, Agile, etc. Make sure to include these keywords in your professional summary.

Apart from general pointers, make sure to add SOMETHING that sets you apart – a popular LinkedIn profile, personal blog, code repository link, technical hobby, events, etc. Anything that you think will add value to your resume.

#Tip: Realistic Resume

Some people put all the keywords in their resume just to get shortlisted. But it won't solve the purpose. Don't overshoot your resume. Keep it realistic. You can add a few keywords but stay true to your hands-on experience and self-learnt knowledge.

What Are Before, After, Beforestep and Afterstep Hooks?

- Before: execute before the feature file execution
- After: executes after the feature file execution
- BeforeStep: executes before each step execution
- AfterStep: executes after each step execution

Tags in Cucumber?

When we only have one, two, or maybe five situations in a feature file, it appears to be simple. In reality, however, this does not occur. In a single feature file, we may have 10, 20, or even more scenarios for each feature under test. They could reflect various purposes (smoke test/regression test), perspectives (developer/QA/BA), and statuses (ready for execution/work in progress).

Tags in cucumber provide a way to run scenarios in a specific sequence from a runner file. Each situation can be labelled with a useful tag. Later, in the runner file, we may specify which tag (and hence which scenario(s)) Cucumber should run. "@" is the first character in a tag. Any relevant content after "@" can be used to define your tag.

Example: '@InitialTest'

Profile in Cucumber?

When testing a feature, cucumber profiles make it simple to define groupings of tests in a feature file so that we can choose to execute only a subset of them rather than all of them. It was created to help people save time. In a cucumber.yml file, the user can reuse commonly used cucumber flags.

We can create Cucumber profiles to run specific features and step definitions. To execute a cucumber profile: cucumber features -p <profile_name>. E.g., cucumber features -p regression.

What is Cucumber Dry Run?

The purpose of the Cucumber dry run is to verify compilation faults and compile the Step Definition and Feature files. Dry run's value might be either true or false. Dry run has the value false by default and it is present in the Test Runner Class file.

If the dry run value is true, Cucumber will check all steps in the Feature file. Within the Step Definition file, it will also check the implementation code of steps in the Feature file.

If any of the steps in the Feature file is missing its implementation in the Step Definition file, a message is thrown. The @CucumberOptions has a dry run parameter that is used to configure the test parameters.

How does the execution start in Cucumber?

Cucumber execution will begin at the support level. In support, it will first load the env.rb file, then hooks.rb, and last start executing feature file scenario steps.

How can you run Cucumber tests parallelly?

The Cucumber JVM Parallel Plugin, which may be used with Serenity BDD, can be used to conduct parallel tests in Cucumber. The plugin will look in the src/test/resources directory for feature files. After that, it will create runners for each file.

How can you run a selected test from a group of tests in Cucumber?

We may execute a single test from a set of tests in the Cucumber framework using the tags idea. This is found in the TestRunner file's @CucumberOptions section. With the use of the @<tagname> keyword, we may tag a scenario in the feature file. A scenario can have one or more tags within the feature file. We can separate test scenarios with the assistance of tagging. We must pass the <tagname> value within the tags argument to execute a selected test in Cucumber, and we must pass the <~tagname> value within the tags parameter to exclude a test from running.

Background keyword in Cucumber?

Background keyword is used to group multiple given statements into a single group. The keyword mostly used when the same set of given statements are repeated in each scenario of the feature file.

How to set priority to tests in the Cucumber framework?

We use Cucumber hooks to control the flow of execution. But this can be modified with the help of the order. Let us take a step definition file having two test methods with @Before annotations. In order to control the sequence of their execution, we can use @Before (order = int) statement. This ensures that the test methods are executed in an incremental manner. This means the test method having order = 1 shall execute before the method having order = 2.

How to generate reports with Cucumber?

We can generate the output/report of the cucumber using different cucumber commands.

```
>cucumber adding.feature --format HTML  
>cucumber adding.feature --out report.html  
>cucumber adding.feature --format pretty
```

The report file will be stored in the project folder itself.

TEST AUTOMATION

Any idea about Jenkins?

Jenkins is an open-source continuous integration (CI) and continuous delivery (CD) solution written in Java. The idea of CI is to merge code from individual developers into a project multiple times per day and test continuously to avoid downstream problems. CD takes this a step further to ensure that all merged code is always in a production-ready state. Jenkins enables developers to automate this process as much as possible -- up to the point of deployment.

Builds can be triggered by various means, for example by commit in a version control system, by scheduling via a cron-like mechanism and by requesting a specific build URL. It can also be triggered after the other builds in the queue have completed.

Running Selenium tests in Jenkins allows you to run your tests every time your software changes and deploy the software to a new environment when the tests pass. This concept is meant to remove the problem of finding later occurrence of issues in the build lifecycle.

How do you calculate automation ROI?

A simple calculation would say,

Total Manual Efforts for n iterations – Total Automation (scripting + maintenance + execution) efforts for n iterations.

As your 'n' (iterations) increase, so does the ROI. There is no universal template.

Tip: We generally measure Test automation ROI on quarterly basis.

GUI Automation vs. API Automation - Which is better?

With CICD gaining momentum - is it possible to maintain continuous testing at the GUI level?

Guess that's where API automation is gaining popularity. Instead of testing the functionality via GUI, focus on testing the business logic at API level itself.

Unlike UI Tests, API automation would be relatively easy to maintain. Executes faster, and can target more code coverage.

Jenkins features

Jenkins is a powerful Continuous Integration tool facilitating technical aspects of continuous delivery.

- Open-source automation server written in Java (portable).
- Server-based system that runs in servlet containers such as Apache Tomcat. Easy Setup: Just run `java -jar jenkins.war`, deploy it in a servlet container. No additional install, no database.
- Easy configuration: Can be configured entirely from its friendly web GUI with extensive on-the-fly error checks and inline help.
- Rich set of plugins with good documentation.
- Tons of integrations: Jenkins integrates with virtually every SCM (e.g., GIT) or build tool (e.g., Maven) that exists.
- Multiple Build Triggers: Scheduled, Dependent on other projects or Build remotely.
- Distributed builds: Jenkins can distribute build/test loads to multiple computers with different OS.
- Test Results: ALM Integration to push the Test results and Email the report as well.

How DevOps is changing the world of Software Testing?

DevOps – a methodology to integrate all functions from Dev to Ops into one cycle, so that the software is deployable at any point in time.

What does it mean for QA? As per me, it just means that we too start early, test continuously in between, and monitor till the end.

Note: DevOps will NOT replace Software Testing ☺

- Obviously, it calls for “Continuous Testing enabled by Automation”. Test-first | Fail-fast-Fail-often. Continuous testing is feasible only with increased automation, i.e., your test suites are triggered with every deployment.
- And to achieve that, automation should shift from UI to API level instead.
- Once your API tests pass, testers can then quickly verify the UI functionally.

All this enables quicker delivery of quality software.

For Testers, it simply means upgrading your skill set to have both manual & automation expertise.

What is Jenkins Master-Slave architecture & why do we need it?

If you have only a few jobs to run – that too on the same platform (e.g., Windows) - having only a Jenkins master should be okay.

But,

If you have hundreds of jobs - running on different platforms (Linux, Windows, etc.) - a standalone Jenkins instance won't be enough – depends on the disk memory & CPU capabilities.

The way out?

Scale your Jenkins by implementing a master-slaves architecture.

- **Master:** base installation of Jenkins. Schedule build jobs. Dispatch & monitor builds to slaves.
- **Slave:** a device (can be a VM, Docker image, etc.) configured (a Java executable) to act as an automation executor on behalf of the master. Execute build jobs dispatched by the Master.
- Master and Slave communicate through TCP/IP protocol.

That's how you manage distributed builds via Jenkins. It allows Jenkins Master to remain responsive to users, while offloading automation execution to the connected slave(s).

Which tool is better to automate AngularJS application? Selenium OR Protractor?

Both Selenium and Protractor can be used to automate both traditional as well as AngularJS applications. But WebDriver tests are Angular-agnostic, they know nothing about Angular and the complicated machinery on the page it uses to manipulate DOM.

It might have sync issues with the AngularJS application, which can be fixed by introducing custom waits, catching known exceptions, soft assertions and retrying failed test steps a couple of extra times to ensure sync.

On the other hand, Protractor is a NodeJS program which is written in JavaScript, integrating technologies like WebDriverJS & Jasmine to facilitate automation for AngularJS applications. It understands Angular and this allows you to write tests a bit easier.

If you are comfortable with Javascript, go for Protractor. Else, code well using Selenium itself ;-)

Any idea about 'Page Object Model'? Or POM?

One of the most popular Design pattern now-a-days. As the name suggests,

- **Page:** Any web page you want to interact with. E.g. Registration page, Login page, Homepage, Search page, Checkout page, etc.
- **Object:** Object-oriented programming jargon. Create a class containing properties (variables) & behavior (functions). Objects are instances of a class.

Combining these – ‘Page Object’ is nothing but creating ‘Objects’ (classes) of different Web pages which the script needs to interact with. Create a page object containing all the page elements and associated methods which are specific to a page. The benefit?

- Separation between Test scripts & Page Classes.
- Provides a single Object Repository along with associated functions.
- You need not identify the web elements in each & every script.
- If locator value changes for an element (UI changes) – change it in the respective page object, which is then used in different scripts. One change = reflected in all scripts.
- Enhanced readability for Test scripts.

Create Page Object for all Web pages >> Based on Test, navigate to the required page & access Page specific methods. When navigating to other page, return that particular page object. Else return the current page object.

Note: We can have multiple Page Objects even for a single web page – say, Header, Footer, Main page, etc.

What are the different formats available to store/access Test Data?

It is advisable to keep Test Data separate from Test scripts using parameterized tests. But what's your Test data source plays an important role in overall Test performance.

Excel: Language library to parse Excel data is generally larger, slower and complicated. Excel consumes more memory while importing data. It's difficult to programmatically manipulate Excel files since the Excel is proprietary.

CSV File: Stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas.

XML: More structured file format with a lot of repeated data (tags).

JSON: Data is in name-value pairs separated by commas. It is less verbose than XML and faster - parsing an XML file is slow and cumbersome.

A Database? Plain Text files helps in version control along with the tests, not so simple with Binary data files like Excel.

Which Tests do you think should be automated?

Using tools to test helps in building a re-usable framework for any repetitive tasks. Or some tasks which are way too tough when done manually.

- Automate whatever possible (not a worthy solution if it is required to be run just once).
- Undoubtedly the regression suite (no explanation required I guess)
- Using tools for performance & security tests
- Unit tests for build confirmation.
- Only the tests which needs to be run repeatedly, i.e. ‘multiple’ times.

Which OOP concepts have you used in your Test Automation Framework?

All.

- **Class-Object:** The heart of OOP. E.g., you create Page Objects in the Test classes to perform relevant functions.
- **Encapsulation:** Access web elements via Page Object methods. No direct access.
- **Abstraction:** From the Test classes, just call the Page class methods to achieve the required functional flows.
- **Inheritance:** Most of the frameworks have a Base Page class and a Base Test class which is extended in all other page & test classes. Generally used for setup & initialization.
- **Polymorphism:** There are times when we need methods differing in the arguments to handle different flows.

What are the challenges you have faced in Automation testing?

Yeah! That's a good practical question to probe into candidate's experience.

- Unrealistic expectations from automation testing. Its automation, not auto-magic.
- Script maintenance is an ongoing challenge (well thought-of framework design can help here).
- Maintaining automation for different products + environments + customers.
- Bridging the divide between automation and manual testers, i.e., team coordination & collaboration.
- Refactoring the legacy code when your suite grows really big.

Which one do you use for your automation locator strategy? XPath or CSS?

CSS is a bit faster than XPath and is more readable. But yes there are a few cases where it is not the right tool, example - searching for a parent element or searching element by its text.

XPath engines are different in each browser, hence make them inconsistent. Coming to IE, it does not have a native XPath engine. In that sense, XPath won't be a good choice for cross-browser scripting. XPath tend to become complex over time.

Also, do you use a mix of locators in your automation scripts? Or follow a single & consistent locator strategy? As for us – we use only CSS locators across the framework.

#Tip: Prepare for Maximum Bandwidth

You don't want your computer or internet connection to give out in the middle of a video interview, or a telephone interview, for that matter (if you're on a VoIP call). Close all your other programs. Sit next to your Wi-Fi connection. Block all other sounds in your environment by locking your cats in a closet, turning off the house phone ringer, and dosing your children with sleeping aids. Well, not really, but you get what I mean.

#Tip: Speak Clearly

In a phone interview or a Skype interview, your voice is going to be weighed much more heavily, since your appearance and other aspects don't have to be. Be upbeat and enthusiastic and try to smile – it really does show in your voice. Have a glass of water handy to keep your throat from getting dry. Speak slowly & be clear, to be sure that your answers are clearly understood.

You must have heard about POM.xml?

Project Object Model: an XML file that contains information about the project and configuration details used by Maven to build the project.

In the normal project, you will add JAR files and libraries as required. In Maven-based project, those JAR files, libraries are added to the project using this pom.xml. In the POM context we call those JAR files & libraries as ‘dependencies’.

Some of the configuration that can be specified in the POM are the project dependencies, the plugins or goals that can be executed, the build profiles, and so on.

If there are enough entries in the pom.xml then that is all you need! Import it to Eclipse >> Maven will download your source code from CVS, download various dependency jars (like log4j, Apache Commons, json, etc.), run the tests, build the jar/war, deploy to your app server, generate a report, etc. Each task is mentioned as a goal.

The most popular use: when there is no Maven, it needs to add all the library JAR files one by one to the project. But when there is Pom.xml there is no need to add library JAR files one by one. Simply add the dependency to the Pom.xml, and it will automatically add the library JAR files to the project.

In short the pom.xml will have all information to build your project.

We are running our suites from Jenkins jobs in different VMs (slaves). What's different with Selenium Grid then?

- Jenkins is just a CI-tool, not tied-up with Selenium or Testing in particular. It can be integrated with any other tool – development, QA, DevOps, etc. to trigger the build execution. Selenium Grid however is specific to Selenium automation.
- Jenkins master-slave configuration is a way to run different jobs in parallel on different machines. I.e., Job-1 in VM-1, Job-2 in VM-2, etc. Selenium Grid however is more flexible in running different jobs OR TCs within a Job in parallel on different configs (OS+browser).
- Selenium Grid is specifically built for parallel execution hence handles most of the management tasks out-of-the-box. For Jenkins – you need to embed it in the code, or use plug-ins or configure/manage accordingly.

Generally, for QA/Testing it will be like Jenkins (CI) >> Selenium Grid (Parallel) >> Node-1 | Node-2 | Node-3 | ...

What's the difference between Git and GitHub? Or is it same?

- **Git:** an open-source distributed version control system, a command-line tool to manage your source code history.

- **GitHub:** a web-based hosting service for Git repositories.

Hub as in the center or the core. To work individually, you maintain a copy of your project code in your local Git repository >> do the code changes >> and then add-commit-push-merge these changes to the hub (remote git repository), i.e. GitHub so that it is now available for all in the team.

In simple terms, GitHub is a website where you can upload a copy of your local Git repository.

Note: You do not need GitHub to use Git. A remote git repository can be created in many other web servers like BitBucket, webfaction or heroku.

What's the purpose of Maven settings.xml?

One, we have pom.xml – which is part of the project itself. It is accessible to all. It contains “project information” that's necessary to build it - mainly all the dependencies.

So, what is settings.xml?

Usually saved in \${user.home}/.m2/settings.xml, settings.xml is your system and/or user configuration – your local system configs and your user info.

And what is system/user configuration?

The path to your local repository, some maven preferences (e.g. interactive mode, offline, use Plugin Registry), configuration for different proxy profiles, server-specific settings (remote repository servers and the authentication info), download mirrors for repositories, build profiles and plugin Groups (group Ids to search for a plugin).

Note: Since it contains personalized configs, the settings.xml file should not be bundled to any specific project, or distributed to an audience. Simply stated - settings.xml file is user-specific (private) whereas pom.xml is project-specific (public).

What are the common Git commands that you use frequently?

- **git clone:** clone a remote repository to your local.
- **git pull:** pull all branches with latest changes to your local.
- **git status:** to see what files have been created, modified or deleted.
- **git add foo.js:** to stage or simply add files.
- **git reset HEAD:** remove all staged files.
- **git commit -m "comments":** commit the staged changes.
- **git reset --soft HEAD~1:** undo your most recent commit and put those changes back into staging.

- **git push:** push your committed changes to a remote repository.
- **git stash:** save changes that you don't want to commit immediately.
- **git branch:** list out all the branches.
- **git checkout <branch>:** switch to different branches.

Test Automation is NOT Auto-magic.

- Test automation is seen as a silver bullet. Click a button and you should quickly get a happy report.
- Automation takes time. It's not a short-term test cycle. It needs thought and preparation to be successful.
- You cannot (& should not) automate ALL cases.
- Putting pressure to automate quickly will only confirm failure at the end.
- Don't plan for short-term automation goals/ROI. Plan for Test automation keeping in mind your long-term goals/ROI.
- Inter & intra collaboration during the initiation phase helps avoid redundancy/rework.
- It doesn't work - once automated, can be run any time in future. Scripts need periodic maintenance to align with current functionality. Account for the maintenance efforts as well.

Let's say you have scheduled nightly test execution. But the application was down from 2-4AM. How will you handle such exception in your automation framework?

We do have Jenkins plugins to handle different post-build actions. In this case the post-build action is "Retry build after failure".

For example – the “Naginator” plugin which allows you to automatically reschedule a failed job with option to specify the delay and max number of successive failed builds (since we don't want an infinite loop here ;-))

What are different types of testing framework techniques?

The most popular frameworks,

- **Data-driven:** separate test scripts logic and the test data which allows to execute scripts by passing different sets of test data. I.e., same script executed multiple times with different test data values.
- **Keyword-driven:** Define ‘keywords’ or ‘action words’ for each function or method. Then, a test script is a sequence of specified keywords. Once keywords & their implementation (actual methods) is defined - Testers with less programming knowledge can also create test scripts by sequencing the keywords.
- **Hybrid:** Combine two or more frameworks and leverage the strengths and benefits of each, as per the environment-application-etc.

- **BDD:** Tests are written in plain descriptive English type grammar and are explained as behavior of application and are more user focused. BDD frameworks such as Cucumber or JBehave are an enabler.
- **TDD:** Repetition of a very short development cycle: requirements are turned into very specific test cases, then the software is improved to pass the new tests, only.

What are the important modules of an Automation Testing framework?

- **AUT:** The application under test
- Test Automation **tool** such as HPE UFT or Selenium
- **Programming language** – Java, Python, Javascript, C#, etc.
- **Type of Framework:** whether it's a keyword-driven framework, data-driven or a hybrid framework. Behavior-driven-development using Cucumber and continuous integration using Jenkins is also popular.
- **Configuration file:** where you maintain the static info - environment variables, login credentials, browser details, file paths, etc.
- **Test Data:** How the Test data is maintained, i.e. in database, excel files, XML, etc. & where it is saved.
- **Utility Classes:** Common actions such as waits, actions, capturing screenshots, accessing excels, sending email etc.
- **Page Classes** - Application Objects/Web Elements: Inbuilt 'Object Repository' OR Page Object Model design pattern.
- **Test Scripts:** Test classes including base class and individual test scripts.
- **Test Suites:** Using excel containing a list of all test scripts with Yes/No parameter? Or TestNG suites?
- **Test Environment:** A stable environment on which application is deployed for test purpose.
- **Test Execution:** Using Jenkins to trigger the automation jobs? Or run locally? In a virtual machine?
- **Results:** This includes the error logs, execution status, screenshots, formal reports, etc.
- **Test Management Tool:** where the test cases & screenshots (optional) are stored.
- **Code Repo:** Where is the code hosted? A git repo? Or SVN?

What all collections have you used in your framework and why?

- **Hashmap:** to store test data OR configuration properties in key-value pairs.
- **List:** to store a list of web elements to be iterated OR to store arguments/parameters.
- **Set:** to manage window handles.

#Tip | ‘Story-Telling’ is one of the differentiators

Story-telling is an art which is equally rewarding. If you can answer an interview question from your personal experience – it’s a great way to earn brownie points. Yeah, once there was a project which had too many defects or a project which missed the deadlines or different third-party co-ordination.

When you narrate a personal experience, the interview is not just about Q-and-As, it is then more a sort of discussion. Sharing your ‘actual’ hands-on experience of a situation is much better than ‘perfect’ theoretical answer.

What are the different Test design techniques used in Automation testing?

Test Design techniques, i.e., techniques used to identify effective test scenarios exploring different product features. As you mentioned BVA, State transitions, Equivalence, etc. are the common test design techniques across industry.

But wait! The second part of the question is a little confusing – “used in Automation testing?”

For automation, ‘Design patterns’ is the term which I know – the best practices and already-tested solutions to general problems. It can be a creational, structural or a behavioral design pattern. Using a design pattern helps in building effective & efficient framework.

Hope the interviewer is asking about ‘design patterns in automation testing’ and not about the ‘Test design techniques in automation’.

How and when automation is started in agile?

A good question indeed.

Agile automation OR automation in agile is very important to aim for quicker delivery cycles. The general practice is to ‘automate N-1 sprint’, i.e. develop automation scripts of the previous sprint in the current sprint. Why? Since N-1 sprint is already tested and verified.

On the other hand, agile pushes the automation from UI level to testing the APIs instead. Since UI change frequently, it makes much more sense to automate your API tests which can be used to test the functionality quickly.

You can also explore the usage of TDD/BDD frameworks while designing the automation framework.

“Page Object Model” is a framework?

No, it’s a design pattern. You implement design patterns within frameworks.

POM is framework-agnostic because it can be used with many different frameworks. We often call POM a framework but mostly it’s a design pattern/class style to create object repository.

POM has become popular in test automation for enhancing test maintenance and reducing code duplication. How? Instead of maintaining locators in every script, build a Page object for each webpage containing its locators and related methods. So, in future if any locator changes - just make that change at one place (Page Object) instead of multiple scripts.

Which is the best open-source tool for Desktop automation?

It depends on your product, OS and what exactly you want to automate. One approach can be to download a bunch of trial versions for tools and try them out to see what works for you.

To list some,

- [SikuliX](#): Automates using image recognition and OCR.
- [WinAppDriver](#): Built on top of WebDriver protocol.
- [Winium.Desktop](#)
- [TestArchitect](#): A Keyword-Driven testing tool.
- [AutoIt](#): To automate Windows GUI.
- [AirTest](#): aimed at hard-to-automate applications like games.
- [Appium](#): bundles WinAppDriver as well as Appium4Mac which can be used to automate Mac desktop applications.

Can we automate ‘exploratory testing’?

Explore: travel through (an unfamiliar area) in order to learn about it.

By definition, it makes sense NOT to automate exploratory tests. How can one automate that is unknown or unfamiliar? It is inherently human – a moving target based on continuous exploration of knowledge about the product.

But then what are these tools that claim to do exploratory testing? Guess these are just monkey testing tools trying different ‘random’ paths (& not based on the gained knowledge of the product) within the application?

What can be the top reasons for automation failure?

- The first – no planning, feasibility and POC research before starting.
- Test automation requires continuous maintenance. If not maintained, the backlog keeps growing.
- Not deploying skilled personnel.
- No standardization across modules-projects-business units-etc.
- Trying to automate everything without a thought/discussion.
- Lack of interest from the management.

Is it important to know Manual testing before starting Test automation?

I would rather rephrase it as “Yes, it is important to know Testing concepts before starting Test automation.”

Test automation is a part of testing activity, so obviously you need to be aware of the testing concepts and have that quality mindset.

The catch here is – First, industry created this divide between manual testers and automation testers and now industry itself has bridged that gap to expect just a ‘tester’ who knows both.

I.e., If you want a successful QA career – get rid of manual/automation tester tag and acquire both the skills equally well.

We run test script and most of them fail due to environment issues and we consider it as PASS and go ahead. We regularly have environment issues and offshore team cannot do anything to resolve as it is client network. What do you suggest doing in this case?

A good real-time scenario. Environment issues are a big bottleneck to test automation success. In fact, to the whole test process itself.

Ideally the test environment should mimic the production environment with smaller infra. But we often see environment issues hindering the test progress.

One suggestion would be to publish ‘environment stability report’ measuring the downtime within a week and how it has impacted the test progress. Once people see frequent downtime in the reports – they tend to take notice and then brainstorm on how to resolve them. But the first task is to showcase the frequent downtime and its impact.

I want to automate a form which consists of OTP verification. Can I automate OTP process?

- a. Connect the phone to the COM Port via USB >> Fetching SMS (smslib.jar) >> Parse SMS to get the OTP.
- b. Connect the phone >> Automate the SMS App (e.g. Appium) >> Parse it to get the OTP.
- c. Register for the HTTP SMS Gateway >> Call the method to fetch SMS >> Parse the message to get the OTP.
- d. Get the OTP from DB, if its in-house application or if it can be accessed.

Difference between CI-CD and DevOps?

DevOps - a collaborative culture with a set of practices, tools, tech, and processes that streamline the product development - with emphasis on effective communication, integration, and collaboration for delivering quality products with continuous rapid deployment. E.g. code ownership, automation, incremental changes, and continuous improvement.

CI/CD - principles and practices that help teams deliver frequent code changes, i.e. ongoing automation and monitoring throughout the application life cycle - from integration and testing to delivery and deployment. “CI/CD pipeline” is one of the best workflows that a DevOps team can follow.

- **CI** usually refers to integrating, building, and testing code within the dev environment. I.e. automatically building and performing unit tests upon making changes to the source code.
- **CD** builds on this, dealing with the final stages for production deployment. I.e. a staging area where code changes can be verified and reviewed manually before releasing to production.

The benefit - CI/CD puts together all code changes into a single repository and runs automated tests. As such, the product is fully developed and ready for deployment at all phases.

What actions do you take for Test Automation stabilization?

Yeah! Script maintenance is an ongoing activity. There is no point in just automating the cases unless your test suite is useful, i.e., well stabilized scripts.

Test automation failures generally fall into below categories,

- **Environment issues:** The application went down, server not responding, data propagation, etc. We need to highlight & follow-up on such re-occurring issues in order to get stable environment for test automation.
- **Script issues:** Constant fixing of sanity/regression scripts as & when failure happens. Scheduled daily/weekly build jobs help here.
- **Config (if any):** Make sure to have the pre-requisites aligned before you trigger test automation. Else, a quick failure analysis might help to update the configs.
- **UI/Backend changes:** Should be part of regular test automation refactoring.

Note: A well-structured report & proper logs help a lot to zero-in the root-cause and subsequent quick stabilization.

What are the different Cucumber Options?

In your Test Runner, `@CucumberOptions` are used to set some specific properties for the Cucumber test.

- **features:** path of your feature files, i.e. where scenarios are described in BDD format using Gherkin language (Given-When-Then).
- **glue:** path of the step definition files, i.e. where implementation of above BDD statements is available.
- **dryRun:** to check if all steps in a scenario (feature file) have a corresponding step definition implemented.

- **tags**: instruct to execute specific tags in the feature files.
- **monochrome**: to enable more readable console output.
- **format**: what all report formatters to use.
- **strict**: fail the execution in case there are undefined/pending step definition.

How are automation tasks prioritized in your project? And who prioritizes it?

First, Test automation should align with your project goals, i.e., the test and delivery cycles.

To do this, we have the yearly goals broken down into quarterly goals – documented. Yes, it's documented in a Confluence page and the goals are revisited every quarter. Goals are defined w.r.t project/product activities (scripting + execution targets), technical enhancements and personal learning. These are team goals & can be re-aligned based on project priorities.

A weekly report on Test automation progress helps to track these quarterly goals.

Daily stand-ups act as a quick check on the current test automation tasks, and where we are heading in order to achieve the quarterly goals.

Regarding, who prioritizes it – Quarterly goals are set as a result of discussion between the Manager & Automation Engineers. Weekly reports are sent by Test Automation lead. And Daily stand-ups has everybody's participation, with Manager checking on the priorities.

How to automate a Captcha image with dynamic values?

Full form of CAPTCHA: Completely Automated Public Turing test to tell Computers and Humans Apart".

One liner answer is: No, we can't automate this. Captcha is made to prevent Bots and Automation.

Do we have any third-party library? Yes, we do have but not at all reliable to automate different types of captchas, and should not be used. Need to use OCR to read the images text, which is actually a complex algorithm and not generic for all types of Images and Bots Captcha.

Then what to do?

- Ask your dev team to disable it in your lower environments like: QA/STG or DEV.
- Ask your dev team to hard code this Captcha value to a specific value only (but only in test env.)
- Ask your dev team to create an API to get the Captcha value in test env.

Hence, very simply, if there are any other ways around to handle CAPTCHA in Automation, then the CAPTCHA is broken and pretty much useless! Means -- Security Breach Alert!

#Tip: Take a pause to think

Don't hurry. I repeat, don't hurry! Yes, do not be in a hurry to answer the question. It doesn't give you any brownie points.

ALWAYS, take a pause to reflect on the question once interviewer is done asking. If possible, ask clarifying questions to better understand the question and also to get some time to think. Take a pause, gather your thoughts, frame it and then answer. The interviewer will respect your desire to offer a thoughtful answer.

It is not a rapid-fire round. A good answer is always better than quick answer!

How do you reduce the browser speed from 10MBPS to 2MBPS?

Go to browser (say chrome) >> Developer tools >> Network tab >> On the right, you should see a label called ‘No Throttling’. If you click on that, you’ll get a dropdown list of a pre-configured speeds that you can use to simulate a slow connection.

The choices range from Offline to Wi-Fi and the numbers are shown as Latency, Download, Upload. The slowest is GPRS followed by Regular 2G, then Good 2G, then Regular 3G, Good 3G, Regular 4G, DSL and then Wi-Fi. Pick one of the options and then reload the page you are on or type in another URL in the address bar. Just make sure you are in the same tab where the developer tools are being displayed. The throttling only works for the tab you have it enabled for.

If you want to use your own specific values, you can click the Add button under Custom. Click on the Add Custom Profile button to add a new profile.

How will you maintain a DevOps pipeline?

Step 1: CI/CD framework

The first thing you need is a CI/CD tool. Jenkins, an open source, Java-based CI/CD tool that popularized the DevOps movement and has become the de facto standard. Jenkins is just one of many open-source CI/CD tools that you can leverage to build a DevOps pipeline.

Step 2: Source control management

The best (and probably the easiest) way to verify that your CI/CD tool can perform some magic is by integrating with a source control management (SCM) tool.

- Avoid merge conflicts while effectively sharing the code modifications.
- Recover a previous version—and the process of making a backup.

A SCM tool helps by storing your code in repositories, versioning your code, and coordinating among project members. Although there are many SCM tools out there, Git is the standard and rightly so.

Result: The CI/CD tool can automate the tasks of checking in and checking out source code and collaborating across members.

Step 3: Build automation tool

You can check out the code and commit your changes to the source control, and you can invite your friends to collaborate on the source control development. But you haven't yet built an application. To make it a web application, it has to be compiled and put into a deployable package format or run as an executable.

Enter the build automation tool. No matter which build tool you decide to use, all build automation tools have a shared goal: to build the source code into some desired format and to automate the task of cleaning, compiling, testing, and deploying to a certain location.

Result: You can put your build automation tool configuration files into your source control management and let your CI/CD tool build it.

Step 4: Web application server

So far, you have a packaged file that might be executable or deployable. For any application to be truly useful, it has to provide some kind of a service or an interface, but you need a vessel to host your application. For a web application, a web application server is that vessel.

You need an HTTP server as well as some other environment (like a virtual machine) to install your application server. There are a number of open-source web application servers available, e.g. Tomcat Apache.

Result: Now the DevOps pipeline is almost usable.

Step 5: Code testing coverage

There are many open-source tools available to test your code and suggest ways to improve its quality. Even better, most CI/CD tools can plug into these tools and automate the process. There are two parts to code testing: code testing frameworks that help write and run the tests, and code quality suggestion tools that help improve code quality.

Result: This is just the tip of the iceberg for what a complete DevOps pipeline can look like. Start with a CI/CD tool and explore what else you can automate to make your team's job easier.

Jenkins: What's is difference between build, job and project?

- **Job/Project:** Jenkins seems to use these terms interchangeably. They all refer to runnable tasks that are controlled / monitored by Jenkins. Go to Jenkins top page, select "New Job", then choose "Build a free-style software project". Actually, when you "create a new job", you will see four types of jobs, and three of them are described as "project".

Jenkins supports four types of projects: free-style, maven, multi-configuration and external job. The free-style project is the central feature of Jenkins. It can be combined in SCM with any build system.

- **Build:** Result of one run of a project.
- **Slaves/Node:** Computers/VMs that are set up to build projects for a master. Jenkins runs a separate program called "slave agent" on slaves. When slaves are registered to a master, a master starts distributing loads to slaves. Term Node is used to refer to all machine that are part of Jenkins grid.

Maven: What are goals in maven & phases of maven?

When Maven starts building a project, it steps through a defined sequence of phases and executes goals, which are registered with each phase. A goal represents a specific task which contributes to the building and managing of a project.

Lifecycle: The Maven build follows a specific life cycle to deploy and distribute the target project.

- default: the main life cycle as it's responsible for project deployment
- clean: to clean the project and remove all files generated by the previous build
- site: to create the project's site documentation

Each life cycle consists of a sequence of phases. The default build life cycle consists of 23 phases, clean life cycle consists of 3 phases, while the site lifecycle is made up of 4 phases.

Phases: Each phase within a life-cycle is responsible for a specific task,

1. pre-clean: execute processes needed prior to the actual project cleaning.
2. clean: remove all files generated by the previous build.
3. post-clean: execute processes needed to finalize the project cleaning.
4. validate: check if all information necessary for the build is available
5. compile: compile the source code
6. test-compile: compile the test source code
7. test: run unit tests
8. package: package compiled source code into the distributable format (jar, war, ...)
9. integration-test: process and deploy the package if needed to run integration tests
10. install: install the package to a local repository
11. deploy: copy the package to the remote repository

Goals: Each phase is a sequence of goals, and each goal is responsible for a specific task,

1. compiler:compile – the compile goal from the compiler plugin is bound to the compile phase.
2. compiler:testCompile is bound to the test-compile phase.
3. surefire:test is bound to test phase.
4. install:install is bound to install phase.
5. jar:jar and war:war is bound to package phase.

Plugin: A Maven plugin is a group of goals. However, these goals aren't necessarily all bound to the same phase. E.g., Maven Failsafe plugin is responsible for running integration tests with two main goals - integration-test and verify.

Maven: Failsafe plugin

Maven Failsafe plugin is responsible for running integration tests. The Failsafe plugin has two main goals:

- integration-test: run integration tests
- verify: verify all integration tests passed

Advantages of Maven

Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

- Build automation tool used primarily for Java projects, can also be used to build and manage projects written in C#, Ruby, Scala, and other languages.
- Resolves Dependencies: define dependencies in POM file instead of manual adding it to your Build Path. Maven will take care of all the dependencies or jar which are defined in pom.xml.

Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories and stores them in a local cache. In the normal project, you will add JAR files and libraries as required. In Maven-based project, those JAR files, libraries are added to the project using this pom.xml. In the POM context we call those JAR files & libraries as 'dependencies'. Some of the configuration that can be specified in the POM are the project dependencies, the plugins or goals that can be executed, the build profiles, and so on.

Git: What is difference between fetch and pull?

In the simplest terms, git pull does a git fetch followed by a git merge. You can do a git fetch at any time to update your remote-tracking branches under `refs/remotes/<remote>/`. It will never change any of your own local branches under `refs/heads`, and is safe without changing your working copy.

- **Git pull:** will merge any pulled commits into the branch you are currently working in.
- **Git fetch:** gathers any commits from the target branch that do not exist in your current branch and stores them in your local repository. However, it does not merge them with your current branch. This is particularly useful if you need to keep your repository up to date, but are working on something that might break if you update your files.

#Tip: Take care to answer the questions

When interviewers ask for an example of a time when you did something, they are asking behavioural interview questions, which are designed to elicit a sample of your past behaviour. If you fail to relate a specific example, you not only don't answer the question, but you also miss an opportunity to prove your ability and talk about your skills.

There is no hurry to give immediate answers. Think it through, keep speaking about what experiences you are thinking about. If not the one in Q, the interviewer might pick up from the one you are speaking about.

#Tip | Thank You & Please.

Thank You and Please are strong words but used only occasionally. When sending a new mail or a reply – always start with a Thank You note. E.g., "Thank You for letting us know about the QA opportunity." And etiquette say to end it with a Please note. E.g., "Please consider my candidature for this opportunity."

Use these in your written interactions before and after the interview!

What are the tools and frameworks required to complete an automation project?

Web Browser Automation | Selenium Webdriver

Selenium Webdriver offers a rich suite of testing functions specifically geared to the needs of testing of web applications of all types. Selenium Webdriver is probably the best option for automated testing of Websites today. It is becoming increasingly popular and it is the first choice of automation testers as well as organizations for automating the testing of Web-based applications for both the GUI as well as the functionality.

Programming Language setup | Java, Python, Ruby

Since Selenium supports multiple programming languages like C#, Java, PHP, Python, Ruby etc., you can pick the language of your choice. It doesn't matter in which language your application under test (AUT) is built. Selenium with Java is recommended because JAVA itself is a very powerful & commonly used language. Before anything else you need the required programming language setup on your machine!

Integrated Development Environment Tools | Eclipse, IntelliJ

What is the first thing that you need to write a code? Yeah, a text editor. But text editor doesn't offer any other advantage like auto-correction, intelligent code completion or form designers. Here comes an IDE – Integrated Development Environment – a text editor with additional support for developing, compiling and debugging applications. E.g., Eclipse, Visual Studio, NetBeans OR IntelliJ.

Testing Framework | TestNG

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use”

Build Automation Tools | Maven, ANT or Gradle

Build automation is the process of automating the creation of a software build and the associated processes. Considered to be the first step in moving toward implementing a culture of Continuous Delivery and DevOps, Build automation combined with Continuous Integration, deployment, application release automation, and many other processes help move an organization forward in establishing software delivery best practices. It helps to improve product quality, accelerate the compile and link processing, eliminate redundant tasks, minimize “bad builds”, have history of builds and releases in order to investigate issues and save time & money.

Behavior Driven Development | Cucumber

Behavior Driven Development (BDD) is a rising methodology to test your code. Behavior Driven Development gives us an opportunity to create test scripts from both the tester's and the customer's perspective as well. At the start, brainstorm about the acceptance test scenarios which should be passed in order to call the software successful. These Test scripts are in simple English language, so it serves the purpose of documentation also. By the time the code is ready, test scripts are ready too. The code has to pass the test scripts defined in BDD. If it does not happen, code refactoring will be needed. Code gets freeze only after successful execution of defined test scripts. In BDD, whatever you write must go into Given-When-Then steps.

Continuous Integration Tools | Jenkins

Continuous Integration is a development practice that requires automation testers to integrate code into a shared repository at regular intervals. This concept was meant to remove the problem of finding later occurrence of issues in the build life-cycle. Continuous integration requires the testers to have frequent builds. The common practice is that whenever a code commit occurs, a build should be triggered. Jenkins is an open source automation server written in Java that allows continuous integration. It is a server-based system that runs in servlet containers such as Apache Tomcat. Builds can be triggered by various means – commit in a version control system, scheduling via a cron-like mechanism or requesting a specific build URL. The basic functionality of Jenkins is to execute a predefined list of steps, e.g. to compile Java source code and build a JAR from the resulting classes. Jenkins monitors the execution of the steps and allows to stop the process, if one of the steps fails. Jenkins can also send out notification in case of a build success or failure.

Other Tools:

- **AutoIT**: to work on windows-based popups (since it's not web-based)
- **Apache POI**: API for Selenium Data driven tests. Apache POI is an open source java library to create and manipulate various file formats based on Microsoft Office like DOC, DOCX, PPT, PPTX, XLS, XLSX. It is mostly used for reading and writing excel documents. Like Apache POI, there are other libraries for performing operations on Excel files.
- **ExtentReports**: To generate dynamic HTML logs, step-by-step test case summary, execution history, filter reports based on test status, captures details like OS-Memory-Java version and attach error screenshots within the report.
- **ReportNG**: A TestNG plug-in, used to create better reports than the default reports generated by TestNG. Once you execute the tests, TestNG generates a test output folder at the root of the project containing detailed report (errors, test groups, execution time, step-by-step logs) in the <index.html> file and a Summary report (test pass/fail/skip count) in the <emailable-report.html> file. Summary report is an email friendly report which you can embed and share with the stakeholders.

What were the challenges you faced in designing Test Automation framework?

Some of the challenges,

Sync issue or Timeout

Sync issue/Timeout issue is one of the most challenging tasks in any test automation tool. Because applications load times differ based on multiple factors. Say, you are performing some action on an element but that element hasn't loaded yet.

Solution: Smart use of explicit/implicit waits or writing a custom method which can handle these sync issues. For example, if you use explicit waits, you can make the Selenium WebDriver freeze the execution and wait until certain conditions are met.

Integration with different tools

Since we are using many open-source tools like Selenium, Maven, Jenkins, Auto IT, etc. so integration between these tools is sometimes a very challenging task.

Locator strategy

Locators are the core part of any scripting and we need to keep on enhancing our locator strategy for script stability.

Tip: Always write dynamic or custom locators.

Dynamic Elements

Many web apps or websites often have web elements that are dynamic in nature, which are not visible when you visit the site for the first time. The dynamic elements can be handled with dynamic XPath or dynamic CSS selectors. You can also use functions, such as contains, ends with, starts-with, etc. to handle dynamic objects.

Cross browser testing

While designing script we always focus on one browser but when it comes to execution - few locators will work in one browser but not the other.

Solution: Once the script is developed, we need to run them on the different browsers and analyze the results for any change in the locator strategy.

Framework Enhancement & Maintenance

Framework enhancement and maintenance is not a one-day activity, we need to keep adding new features or libraries so that we can minimize execution time and maintenance task.

Pop up handling

In many applications, you will find random pop-ups and their behavior is not consistent, so need to handle that as well. Moreover, windows-based alerts are beyond Selenium's capabilities as they are part of the operating system, not the browser.

Explain which priority Test Suites you run in Automation, do you run all P1, P2, P3 suites?

- First, all P1-P2-P3 suites are scheduled to run every weekend.
- Next P1-suite is triggered when we have some fixes deployed. Post which, if the P1-execution is okay and we have some time – P2-P3 suites are executed.
- Sanity suite is scheduled to be run daily morning before we start the work and post every build deployment.

Maven: What are transitive dependencies.

There are two types of Maven dependencies:

- **Direct**: These are dependencies defined in your pom.xml file under the <dependencies/> section. Say, dependency A.
- **Transitive**: These are dependencies that are dependencies of your direct dependencies. Say, A above is dependent on B. Maven avoids the need to discover and specify the libraries that your own dependencies require by including transitive dependencies automatically.

Maven: What is dependency, group id and artifact id?

Dependency: a library, framework or otherwise JAR file that is stored on a remote server and upon the proper detailed information placed in your maven pom file, will be downloaded and referenced in your project's classpath either upon compilation, testing, or during runtime.

Maven uses a set of identifiers, also called coordinates, to uniquely identify a project and specify how the project artifact should be packaged:

- **groupId**: a unique base name of the company or group that created the project.
- **artifactId**: a unique name of the project. The artifact ID is also used as part of the name of the JAR file produced when building the project. The output of the build process, the build result that is, is called an artifact in Maven.
- **Version**: a version of the project.

#Tip: The Alternatives

Industry is now looking for multi-skilled QA-Testers who are good at testing and can also code for test automation. Only manual testing skill-set won't suffice. Only test automation won't suffice. The suggestion would be,

- *Multi-skill Testing: Highlight your testing skills of different types, levels & techniques. Say web testing, API testing, database testing, UAT testing, exploratory testing, etc.*
- *Product Testing: If you have any specific product testing experience, say T24, AEM, Salesforce, etc.*
- *Domain SME: Become a Domain SME. Few domains still have opportunities which might discount test automation only if you have strong domain expertise, say capital markets, P&C Insurance, Healthcare, etc. And find a way to highlight it via your profile/resume!*

Test automation anyways is one of the ways to upskill and find relevant opportunities.

Maven: How to validate whether a particular jar is added properly or not in Maven?

Run `mvn dependency:analyze`. It should do the work.

OR

You can directly check in your project path under 'Maven Dependencies' package.

What is a snapshot version?

A snapshot version in Maven is one that has not been released. The idea is that before a 1.0 release (or any other release) is done, there exists a 1.0-SNAPSHOT. That version is what might become 1.0. It's basically "1.0 under development".

The difference between a "real" version and a snapshot version is that snapshots might get updates. That means downloading 1.0-SNAPSHOT today might give a different file than downloading it yesterday or tomorrow.

Note: Usually, snapshot dependencies should only exist during development and no released version (i.e., no non-snapshot) should have a dependency on a snapshot version.

How occasionally your build is run and different trigger scenarios like as soon as the dev build comes or nightly execution.

- Sanity suite is scheduled to be run daily morning before we start the work and post every build deployment.
- All P1-P2-P3 suites are scheduled to run every weekend.
- P1-suite is triggered when we have some fixes deployed. Post which, if the P1-execution is okay and we have some time – P2-P3 suites are executed.
- All P1-P2-P3 suites are executed during the regression cycle.

Test Automation is both under-rated & over-rated.

Automation Testing, the go-to word today for almost every QA. Automation Testing is both under-rated and over-rated.

As a Tester, we under-estimate the power & applications of Automation. The default understanding is – Automation is only for regression + automation is only confirmatory, not exploratory to find bugs + it can be done only on stable application + waste of efforts due to high maintenance.

As a Management, we over-estimate the power & applications of Automation. We talk about on-the-fly scripting, 100% functional coverage, unrealistic timelines, introduce AI+ML algorithms, everybody automate, all this when your team is struggling even with the regression stabilization.

Why do you think Test Automation is getting popular?

'Automation' is the buzzword today. Testing using tools has become an integrated skill. Some tools like Selenium have become a basic skill.

But why is 'Automation' gaining so much limelight?

- There is no agile without test automation. Agile brings in the concept of quick delivery – which is only possible if test automation is utilized intelligently.
- Automation saves time & effort.
- After being under-rated, finally automation is catching up.
- It started from Regression but automation is now capturing different avenues – like Test-driven-development.
- Testing without tools, though effective - was/is not efficient. We give too much effort estimations.

Bloated Automation.

When you try to automate everything under the hood without any fore-thought, only to realize – Oops! This automation has grown too big to handle/maintain.

Why? Just because leaders love to fit the word 'automation' in about everything. It's a marketing tool first.

But it doesn't work in the long-run. It's a selfie 😊 :-P If you try to automate everything – one day it will be really tough to manage. Add to it – the changing people working on it over time.

Automation Best Practice

Two small yet beneficial test automation scripting practices.

1. **Writing Generic methods.** Generic as in broad or standard methods which cover one aspect and not just particular field. Ex. – instead of multiple methods to update different fields under a tab, it is always better to write a generic method for the tab with particular field as an argument.
Adv: You need not write multiple duplicate methods doing the exact same thing – just on different fields of the common UI.
Catch: You need to identify generic locators which can traverse through the fields one by one.

2. **Modular programming.** Small reusable methods that can then be used to form bigger methods as & when required. Ex. – login, navigate, search, update, save, etc. should be individual methods that can then be reused to form bigger methods to validate the functional flow.
Adv: Reusability. You need not write the same steps time & again for different flows. Say, login. Just call the smaller login method in other required methods.
Catch: Identifying the baby-steps that can be clubbed to form a reusable method.

Automation mis-understandings

Some of the pointers (whatever you call them – misunderstandings/myths/facts/etc.) regarding Test automation.

- Automation takes time. You cannot just jump-start and finish it quickly.
- Automation is for the long-run. To actually yield ROI, keep an eye on the long-term benefits of test automation. Strategize.
- You should not automate everything. It's not like you keep on adding tests to your suite and one fine day – it is too heavy to carry!
- A successful test automation needs some design thinking. It is not just about taking the test cases and then scripting it one by one.
- Automation is NOT to replace human-doing-testing. Rather, it is to complement the manual tests for better quality.

Automation, a boon or a bane?

Just like any other technology, if you use it wisely – there are some major benefits. And vice versa ;)

Utilize automation wherever it's logical. Not because every other Project & Org is driving crazy on automation.

Nah! Not automating (or not trying to automate) everything doesn't mean that you are from the stone-age :P Utilize automation where it makes sense – repeated tests, continuous testing, parallel execution, device fragmentation, etc.

If this is not the case – you are burning the Org's revenues just to prove that you are technically ahead. It won't last for long.

#Tip: The Mail to Recruiter

The first mail you send to a recruiter is important. Hence it should portray all the important criteria.

Subject: "Position | Experience | Location"

Body: "Hi ABC,

"Got to know that you are hiring for <role> position at <Location> location. Please find attached my latest resume for your consideration.

- *Total Experience:*
- *Relevant Experience:*
- *CCTC: xxx Lacs*
- *ECTC: Open for Discussion*
- *Notice Period: xxx Days (negotiable)*
- *Reason for Change: Career Growth*

Do let me know if you need any other information from my end.

*Thank You,
<Full Name>
+91-xxxxx-xxxxx"*

Some important pointers related to Test Automation,

- Aim for high Test Coverage
- Understand the Code You Test
- Never trust the User: User entered data is important.
- Fix your Flaky Tests: Inconsistent failures should be fixed as it can reduce the trust.
- Identify Opportunities: Don't just automate the test case as it is – find opportunities to expand the scope.
- Not just "Test cases": Expand the automation beyond test-case-tool-test-script to help humans do their jobs better.
- Treat automation as Software development: account for the same activities - Design, Implementation, Storage, Testing, Bug fixing, Logs, etc.
- Follow appropriate coding standards and idioms
- Consider maintenance and upkeep: There will be application changes and/or bugs. Develop in a way to reduce the maintenance; we also must allocate time for these activities.
- Logs and reports: Appropriate logs, results, and error messages are critical to maintaining automation.
- Influence testability and automatability: We must let the devs know what's required to perform testing and to create automation.
- Avoid Hard coding: Make your design/code generic instead.

Is learning 100% automation possible?

What is '100% Automation'?

Does it mean **learning all available tools in the market?** No, it is not possible. With different application types & different types of testing – there are numerous tools. Be it functional, web, performance, desktop, API, mobile, security, embedded, etc. The list goes on and on...

Or to actually **master a particular Automation tool set**, say Selenium? No, it is not possible. An Automation Tool set consists of different (& evolving) individual tools & languages and concepts. Even the experts learn something new now & then.

To achieve 100% Automation? No, it is not possible. At the end applications are used by humans and it needs a round of testing by humans. AI has not reached that pinnacle yet...

So, what should we do? Learning any technology is a moving target. There is just 'Learning', and no '100% Learning', literally. Start with a popular tool and keep learning...There is no top, only further heights to reach 😊

Do you find the bugs through automation testing? If Yes, which type and what bug has been detected?

Yes, we do find defects via well-implemented Test automation.

Most of the projects have regression suites automated – so generally we get **regression defects** once you execute the regression tests.

With agile and DevOps practices, many projects have implemented CI-CD pipelines which are effective to quickly identify any **build and sanity issues**.

Note: It can be functional UI issues OR API defects OR database errors OR performance issues – depends on which level you have automated and are running the tests.

Going one step ahead, projects that adopt TDD model are able to identify [**or rather prevent**] any functional defects reaching the test phase itself.

What is Object repository?

Object: In test automation, any field that you want to interact with is also known as object/element. E.g., textbox, dropdown, hyperlink, radio button, alert, checkbox, etc. anything!

Repository: a central location in which data is stored and managed.

Combining these two, an object repository is a centralized storage of element locators in the form of objects.

It is important to separate element locators and test scripts. Why? Because if there are any changes in locator, you don't need to update that in all the scripts – just update it in the central object repository – and it will be automatically reflected in all test scripts.

Note: Object Repository is a general test automation concept, but more closely tied with HPE UFT.

How do you build object repository? In HPE UFT – you have dedicated support. In Selenium – either you follow the Page Object Model to store page locators in separate page classes OR use a file as object repository. In any case – locators should be separated from test scripts.

Jenkins – How do you schedule a job? What is the format & what is it called? How do you configure a Jenkins job to run every 15 mins?

In Configure option > under 'Build Triggers' > select 'Build Periodically'.

To mention the schedule, Jenkins uses a cron expression, and the different fields are:

- MINUTES: Minutes in one hour (0-59)
- HOURS: Hours in one day (0-23)
- DAYMONTH: Day in a month (1-31)
- MONTH: Month in a year (1-12)
- DAYWEEK: Day of the week (0-7) where 0 and 7 are Sunday

Note: A CRON expression is a string comprising five or six fields separated by white space that represents a set of times, normally as a schedule to execute some routine.

Example,

- Schedule build every day at 8h00 = 0 8 * * *
- Every 15-minutes = */15 * * * *
- Every day at 8:30 = 30 8 * * 0-6

How do you debug Test automation failure?

Debugging, an important activity in test automation. Say you triggered a test automation job – once completed – how would you analyse the failures?

The first step would be to check **Test execution report**. In mostly all test automation frameworks – we capture screenshots [for all steps or at least for failures]. Check the report > go through the failed cases to verify the screenshots. What are we looking for? It can be an environment issue, or a defect – anything which can be caught clearly via screenshots.

Once we are done with test report analysis, the next step is to **verify logs** for remaining failed cases. If logging has been correctly implemented in the test framework – it might tell you a few things about the failure.

If still we don't have any idea about the failure, the next step would be to **run the script locally** and verify where is it failing. It might need a script fix OR script stability issue (to be rectified) OR a requirement changes OR a defect, etc. Also, we might need to run it in debug mode to verify the steps & associated test data.

Basically, in debugging we are trying to find (& possibly fix) the root-cause for script failure – environmental, requirement changes, script issue, script stability issue, test data, OR a defect.

How many test cases you have automated per day?

There is no one answer that fits all projects.

It depends on the test case steps and complexity. Sometimes we can automate 3-4 Test cases per day which have limited steps and few validations. On the other hand, few complex test scenarios may take 1 day or more.

Always mention both best and worst count for your project experience and then conclude with the average cases automated per day. I.e., Numbers should justify your experience.

Note: Also, sometimes you get stuck in some technical challenge which eats up a day just to stabilize the script.

How to add Screenshots in extent report?

Extent Report is a library that can be used to build a customized detailed report. ... This report can be built in JAVA, DotNet and it provides a detailed summary about each testcases and its each test steps too in graphical manner.

In simple terms, you just call the method **addScreenCaptureFromPath(screenshot path)**; to add screenshots to the extent report. We can add this only with OnTestFailure, or with any ITestListener method.

```
public void onTestFailure(ITestResult arg0)
{
    try
    {
        //Log failed test case name
        test.log(Status.FAIL, "Failed Case is: " + arg0.getName());
        //Add screenshot to report. takeScreenShot method [in TestUtilities] returns the path.
        test.addScreenCaptureFromPath(TestUtilities.takeScreenShot(arg0.getName()));
        //Additional error info
        test.log(Status.FAIL, arg0.getName()+" FAIL with error " + arg0.getThrowable());
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
    report.flush();
}
```

How do you run Selenium tests from Jenkins? Or how to integrate Jenkins with Selenium?

Before we move to creating Jenkins job, let's understand the basic components test automation framework.

- **IDE:** editor where you write your tests, e.g., Eclipse, IntelliJ, etc.
- **Selenium:** a library which is used while writing test scripts for web UI automation.

- **Build tool:** It could be Maven/Gradle/etc. Used to manage project dependencies and to build a project. E.g., Maven – your pom.xml file has all the dependency information which your project needs [selenium, extent report, log4j, apache poi, etc.] to build successfully.
- **Repository:** centralized version control system where you save your code. E.g., Git repository.

Now that basics are covered, what do you think Jenkins need the most to execute the tests?

Yes, the code 😊

1. When configuring a new Jenkins job, under “Source Management”, select the appropriate repository for the location of your project and pass the URL and credentials. E.g., Git repository URL and Username/Password. And also, the branch to be used under “Branches to build”. That’s it.
2. In the “Pre-steps” build section, specify the Root pom and Maven targets [“Goals and options”] that need to be executed in order to run the tests, in this case “clean test”. “Goals and options” field can also accept any other command line options to Maven, such as “-DskipTests=true”.
3. Last, in the post-build section “Archive the artifacts” we can mention the path where we need to archive the results (screenshots, reports, etc.).

Run the job > code is checked out from Git repo > build using Maven pom.xml and goals > execution started.

Can you explain your Test Automation framework?

One of the most popular interview Q, and also one of the important one. We will try to explain a Selenium hybrid framework. The important point to note is that you should mention all different parts of your framework.

*“We are using a hybrid framework built on **Selenium with Java**. We are using ‘**Page Object Model**’ design pattern where we have Page classes containing the locators & methods corresponding to each of our application’s webpage and then we have Test classes where these methods are called. Environment variables such as URLs/Username/Password/Database parameters/etc. are stored in a **Config properties file**. For Test data, we are using csv/excel which is being read using **Apache POI library**. Logging is implemented via **log4j library**. We are using **TestNG** as our test framework hence have testng.xml files to manage test suites. To segregate cases, we are using TestNG grouping feature. For reporting, we are using **ExtentReports**. **Maven** as our build tool to manage dependencies and **Jenkins** for our test execution. After every build execution in Jenkins, the results are pushed to Test Management tool **ALM** via Microfocus Jenkins-plugin”.*

Once done, be ready for a follow-up Q like how do you read property file? OR How is ExtentReport implemented? OR What all different TestNG annotations have you used? OR Why Page Object Model? OR How frequently do you run Jenkins jobs? Etc.

What measures will you take to reduce Test automation execution time?

Few of the measures could be,

- Code Optimization/Refactoring to remove unnecessary waits (if any) and instead use explicit waits wherever possible so that the execution continues as soon as the condition is met.
- Utilize 'Parallel test execution' via TestNG or Selenium Grid or multiple Jenkins jobs etc. Running tests in parallel reduced the execution time drastically.
- If classes are run parallel, we can include multiple @Test in single class so that login-logout-setup-teardown methods are executed only once at class start and not at every @Test.
- We can also try test execution in headless browser. Not a drastic time saving, but just an idea.

Can you send email through Jenkins?

Yes, Jenkins provides with an email notification service through which you can report the build status and testing results to the team.

How? This can be done with the help of default Email notifier [has a default message consisting of a build number and status] OR an Email plugin [configure every aspect of email notification such as when to send, who receives it, and what the email says]. Plugins are the primary means of enhancing the functionality of a Jenkins environment to suit organization or user-specific needs.

1. Click Manage Jenkins > Manage Plugins > Search & Install plugin.
2. Click Manage Jenkins > Configure System > Section Email notification > provide SMTP server details/username/password/etc.
3. Configure Job > Add a post-build step 'E-mail Notification' > Enter the details like recipient, trigger event, etc.

When? Say, if the build is not successful then the team is notified about the status of the build. Or after the Test execution, a report is sent to team members.

#Tip: Think out loud

Interviewers want to see not only what you know, but also “how you think”. Your problem-solving ability is equally important to arriving at a solution. Sometimes you might not get a solution but what different approaches you took might impress the interviewer.

How? Be prepared to share your thought process and explain the rationale behind your decisions. When asked a question to solve a problem or develop code - ask clarifying questions, state your assumptions, and be prepared to share your opinions or explain your choices.

Don't just sit there & keep thinking – instead keep speaking what you are thinking about how to solve it, do you have similar experience, what different approaches are you thinking about, the logic you are thinking, etc. That way, the interviewer knows that you are trying 😊

How do you pass environment variables via Jenkins Job?

Global Properties: Manage Jenkins > Configure System > Global properties option > Check the ‘Environment variables’ checkbox and then add the variables and their respective values. E.g., JMETER_HOME = ~/jmeter

Jenkinsfile: Set environment variables globally by declaring them in the environment directive of your Jenkinsfile.

Jenkinsfile (Declarative Pipeline)

pipeline

{

//Setting the environment variables DISABLE_AUTH and DB_ENGINE
environment

{

 DISABLE_AUTH = 'true'
 DB_ENGINE = 'mysql'

}

}

‘EnvInject’ plugin: Install and use the ‘EnvInject’ plugin to inject environment variables during the build Start-up. Job Configuration > Add Build step > Inject environment variables > add the required environment variables.

Any idea about Virtualization – Virtual Machines – Containers – Docker – Kubernetes?

Virtualization: Create an abstraction layer over computer hardware that allows the hardware elements of a single computer to be divided into multiple virtual computers, i.e., multiple operating systems to run alongside each other, sharing the same physical computing resources.

Virtual Machine: In virtualization, we are dividing a single hardware into several independent ‘virtual machines’. VMs enable teams to run what appear to be multiple machines, with multiple operating systems, on a single computer. How? Using ‘Hypervisors’, that can separate VMs from one another and allocate processors, memory, and storage among them.

Containers: a light-weight, more agile way of handling virtualization. Rather than spinning up an entire virtual machine, a container packages together everything needed to run a small piece of software - including all the code, its dependencies, and even the operating system itself. This enables applications to run almost anywhere - a desktop computer, a traditional IT infrastructure, or the cloud.

Virtual Machine vs. Containers

- **VMs** - a hypervisor virtualizes physical hardware, containing a guest OS, a virtual copy of the hardware that the OS requires to run, and an application and its associated libraries and dependencies.
- **Containers** - instead of virtualizing the underlying hardware, containers virtualize the operating system (typically Linux) so each individual container contains only the application and its libraries and dependencies. Containers do not need to include a guest OS in every instance and can, instead, simply leverage the features and resources of the host OS.

Docker: Docker is a set of “Platform As A Service” products that use OS-level virtualization to deliver software in packages called containers.

Problem: Large enterprise applications can include a massive number of containers. Managing these containers presents some serious issues for teams. What is running and where? handle crucial issues such as security and compliance? Etc.

Kubernetes: an open-source container-orchestration system [automated configuration, coordination, and management of computer systems and software] meant to coordinate clusters of nodes at scale in an efficient manner. It keeps track of your container applications that are deployed into the cloud. It restarts orphaned containers, shuts down containers when they're not being used, and automatically provisions resources like memory, storage, and CPU when necessary.

What are some measures that you will take on improving Test Automation KPI?

The first and foremost would be to follow the **Test Pyramid** for building test automation. I.e., more tests at unit level, then API and then UI. When you focus more on Unit and API layer, the automation is relatively stable requiring less maintenance and also issues are caught early in the cycle.

Target for **In-sprint Automation**. Generally, test automation is done for N-1 sprint but the goal should be to achieve in-sprint automation.

When writing scripts/framework, focus should also be on maintenance. Test automation eats up a lot of efforts in maintaining the test suites. Only when you reduce the **maintenance efforts**, that you can achieve greater ROI.

Build a **CI-CD pipeline** so as to trigger test automation with every build. It helps in utilizing test automation to the fullest and also catching any issues early.

These are just a few pointers, but there are many improvement areas in every project you work on 😊

#Tip | Though Manual, still learn the basics of Test automation.

Fine, you a Manual Tester. You have the domain expertise. You are a master in product testing. Or team management as a Test lead. Or planning? Whatever!

You still need to know the basics of programming and test automation tool. You need not be a framework-developer, but every organization now wants a Software Tester who knows both functional test + automation scripting.

So don't be ignorant about test automation. Have some basic knowledge and practice about programming and test automation tool of your choice. Something is always better than nothing!

Any idea about Microservices architecture?

Past: A single monolith would contain all the code for all the business activities an application performed. As the application's requirements grew, of course, so did the monolith.

"A microservices architecture consists of a collection of small, autonomous services. Each service is self-contained and should implement a single business capability".

How? Isolate software functionality into multiple independent modules that are individually responsible for performing precisely defined, standalone tasks. These modules communicate with each other through simple, universally accessible application programming interfaces (APIs).

- Fragmented into multiple modular, loosely coupled components, each of which performs a discrete function.
- Individual functions align to business capabilities.
- Can be distributed across clouds and data centers.
- Treat each function as an independent service that can be changed, updated, or deleted without disrupting the rest of the application.

Benefit? Applications are simpler to build and maintain when broken down into smaller pieces that work seamlessly together. More importantly—much easier to expand and scale.

What are some ways to reduce Test-Automation script maintenance efforts?

Script maintenance is important. It doesn't make sense to keep automating test cases which eventually won't yield result on a recurring basis. But there are few tips & tricks to reduce the maintenance efforts,

- First & foremost – **Think twice, code once**. A well thought-off test framework helps a lot in reducing the efforts later on. E.g., smart locator strategy, segregating config-locators-script-test data-assertions-etc. Additionally, focus should be on writing a stable script using proper waits and locators, not a flaky test that would need maintenance later on.
- **Independent tests:** Avoid too big tests which have many steps with different validations. Tests should be short & independent to avoid flakiness.
- **No Duplication:** Avoid writing duplicate code for similar actions. Instead focus on writing generic methods which are easy to maintain (since only one method would need a change).
- **Wait and Locator strategy:** Build a wait and locator strategy so that everybody writes consistent code. Example – only use explicit waits (no sleep or implicit wait) and follow the locator strategy ID > Name > Class > CSS > Xpath.

- **Review:** Don't let anybody check-in the flaky tests. Have a review process defined which checks for execution result as well.
- **Test Reports** play a vital role in minimizing the maintenance efforts. Take your time and build a reporting that can quickly give you enough information about the test failure – screenshots, logs, error stack trace, etc.
- **Execute-Analyse-Fix-Execute:** Have a process to continuously keep stabilizing your tests as & when new failures occur. It's should be an ongoing activity – fix your flaky tests.
- As a QA team, we should **push developers** to keep test automation in mind during their development. For example, defining a naming convention for UI elements (say ID) so that elements are located even after the UI changes.
- **Beyond the UI:** UI automated tests are hard to stabilize since there are frequent changes. Instead think of test automation at integration layer [API testing] OR unit level.

Does Test automation require some specific skill-set or everybody in the team can contribute?

Test Automation is just like a development project. As Testing requires a specific skill-set, so does test automation.

It requires an understanding of different tools that come together to enable test automation. At the foundation, programming knowledge and practise is a must. But it can be learned & practised, no rocket science there 😊

Everybody in the team can contribute? Yes and No. It all depends on the willingness & planning. Willingness to learn and contribute and planning for training/up-skilling the team members. People need some training and hands-on time before they can start contributing.

The approach should be to develop that willingness & competency within the team members before they start contributing. With execution of proper training plan, may be a month or two – and we should be good to go!

How do you solve merge conflicts in your source code using GIT?

Merge Conflict: Conflicts occur when the same file was changed in contradictory ways by two or more people. Git can most likely figure things out on its own but what if the exact same lines were edited in that file. In that case, Git has no way of knowing what's correct - you'll have to look at the changes and decide how you want the file to finally look.

Solution:

- We can simply open the file in an editor [say Eclipse], search for the conflict markers and make any necessary modifications. When done, add-commit-and-merge.

OR

- There are lots of dedicated "Merge Tool" applications that help with this process. We generally use 'KDiff' as our merge tool which shows different modified versions of the file content and we get to choose which changes to keep and discard > save the file > add-commit-and-merge!

How do you measure Test automation success?

In simple terms – total manual efforts saved.

How? We track Test Automation ROI on a quarterly basis.

I.e., How much manual efforts have been saved by utilizing Test automation over a period of every 3-Months.

- Test Automation effort: Scripting effort + Setup + Execution + Analysis + Maintenance.
- Equivalent manual effort if those cases would have been run manually.

The Test Automation ROI report is published to all the internal stakeholders.

Note: Test Automation yields ROI only if it is utilized multiple times for test execution. Say daily sanity | build sanity | weekly regression | etc.

How do you ensure test automation script is updated with corresponding update to test cases?

Problem: Test cases are written > Test automation scripts are prepared > Test cases are updated based on change requests/new requirements > Test scripts are still old hence result in false positives/failures.

Solution: There has to be a process to know which all test cases have been updated. How? We can use the test case management system.

1. Have a field 'automation status' at test case level in the tool.
2. Test cases are written in tool with automation status as 'New'
3. Pick all 'New' cases for test automation analysis, mark the automation status as 'Not feasible', 'Rejected' OR 'Candidate'.
4. Build test automation scripts for all 'Candidate' cases.
5. Once test script is ready-and-executed, mark automation status as 'Completed'
6. If there are any changes done to the existing test case, update the automation status as 'Modified'
7. Next time, pick all 'New' and 'Modified' cases for test automation analysis and scripting.

This way, test cases in the system and test automation scripts are always in sync.

Which all Jenkins plug-ins are you using?

- **Dashboard view:** monitor the status of all the jobs - duration taken by each job and entire time execution.
- **View Job Filters:** build different views for all your Jenkins jobs based on products, test levels, customers, etc.
- **Parameterized Trigger plugin:** get user input as a variable and use on run time. Used for dynamic environments where you have lots of options and user-defined values to be used in the build which may keep changing.
- **Git Plugin:** provides access to GitHub as an SCM which acts as a repository.
- **GitHub Integration Plugin:** integrating Jenkins with GitHub projects - schedule your build, pull code and data files from GitHub repositories to Jenkins, and automatically trigger each build as needed.
- **GitHub/GitLab Pull Request Builder:** automate code review. Upon initiating a pull request, the plugin will retrieve the source and build results so that automatic merging can be initiated.
- **JIRA Plugin:** integrate JIRA into Jenkins, view Jenkins through JIRA.
- **Maven Integration plugin:** build projects (maven projects) and produce the corresponding JAR or WAR build artifacts.
- **Pipeline Plugin:** automate continuous delivery pipeline and performs other complex tasks.
- **Build Pipeline Plugin:** provides a view of the jobs that make up the build pipeline, upstream and also downstream. Also define manual triggers for specific tasks that may need intervention before their execution.
- **Kubernetes plugin:** work with Kubernetes, setting up and tearing down Jenkins' agents.
- **Mailer Plugin:** configure email notifications for build results.

API TESTING

Any idea about SOAP UI basic flow?

Consider a FX project using a Web Service to fetch current currency conversion rates from FXmicropay. Front-end system is connected to FXmicropay via Web Service. SOAPUI is one of the popular Web Service testing tool in the market. The steps include,

1. **Connection:** using WSDL (Web Services Description Language) file. WSDL is an XML based file which is used to describe the services offered by a web service – method used to get the rate, currency abbreviations, response details, HTML get & post request details, etc. Enter the WSDL URL in SOAPUI to make a connection.
2. **Mapping:** Understand the request-response XML mapping as per business/system requirements.
3. **Request:** Prepare the SOAP request in an XML format. Request contained the From & To currency.
4. Send Request XML
5. **Validate** the Response XML received & displayed in SOAP UI.
6. Now that web service request-response is validated, smoke test it via front-end as well.

What are the common HTTP Request methods?

HTTP defines a set of request methods to indicate the desired action to be performed.

- **GET:** retrieve data.
- **HEAD:** identical to GET request, but without the response body.
- **POST:** submit an entity to the specified resource.
- **PUT:** modify data with the request payload.
- **DELETE:** delete the specified resource.
- **CONNECT:** establish a network connection to a web server.
- **OPTIONS:** find out the HTTP methods and other options supported.
- **TRACE:** performs a message loop-back test.
- **PATCH:** apply partial modifications to a resource.

What's the difference between API and a Web Services?

Yeah, API and Web Services are sometimes confusing. But always remember,

"All Web Services are APIs, but not all APIs are Web services"

How?

- **API:** Application interface, meaning that one application is able to interact with another application in a standard way – be it over the network or locally.
- **Web Service:** a type of API, which MUST BE accessed through a network connection.

Spot the difference? Yeah, the network!

APIs can be exposed through local files (such as a JAR file in a Java program, .H file in C/C++ programs, etc.) to allow two local applications to communicate with each other. This doesn't require a network as the two applications are communicating within a single device.

Note: REST APIs are a standardized architecture for building web APIs using HTTP methods, i.e. they are a type of Web Service since they use HTTP network.

HTTP Status Codes

It takes a lot of work to provide a reliable API - which consumers can trust to provide consistent, stable, meaningful, and expected behavior. In this line - 'HTTP Status codes' are standard responses that help identify the cause of the problem when the resource request is not processed properly.

HTTP response status codes are grouped in **five classes**:

- Informational responses (**100–199**),
- Successful responses (**200–299**),
- Redirects (**300–399**),
- Client errors (**400–499**),
- Server errors (**500–599**).

Some common status codes,

- **200 Ok** | Successful requests other than creations and deletions.
- **201 Created** | Successful creation.
- **204 No Content** | Successful deletion.
- **400 Bad Request** | The path info doesn't have the right format, or a parameter or request body value doesn't have the right format, or a required parameter is missing, or values have the right format but are invalid in some way.
- **403 Forbidden** | The invoker is not authorized to invoke the operation.
- **404 Not Found** | The object referenced by the path does not exist.
- **405 Method Not Allowed** | The method is not one of those allowed for the path.
- **409 Conflict** | An attempt was made to create an object that already exists.
- **500 Internal Server Error** | The execution of the service failed in some way.

HTTP Status Codes aren't just a nice to have thing when it comes to APIs, they are essential.

#Tip | Understand the market trends

Don't fret over 'There are no Manual testing jobs' OR 'Nobody is asking for Manual testers. The industry WILL change, always. Fretting about it won't help.

Instead understand the market trends and be ready for the future. Don't just think about this job switch – what about five years from now? Yeah! Understand the trends – work on it – be future ready.

It could be product testing specialization, becoming a domain expert QA, test automation, etc. Pick a trend & start learning!

#Tip | Understand the 'Interview pattern'

Interviews are different than your actual day-to-day work. The interviewer needs to gauge your skills & competencies within hours. How? By covering different topics. Generally, a company or project has a checklist requirement of different topics to be covered – say agile, testing concepts, test automation, API testing, SQL, programming language, attitude, flexibility, communication, etc.

So, it is important for us as professionals to understand the pattern and prepare accordingly. Only Test automation will not do. Only Testing concepts won't do. Only Lead/Manager attitude won't do either. Interviews are holistic in nature to gauge candidates on ALL required parameters – and all topics are knock-out rounds. So, understand the common trends (via interviews/Job desc.) and prepare accordingly! You can even refer the "Interview Experience" and "Interview Preparation" menu at Parikshram to be better prepared & crack your next opportunity. All the very best!

What are some common 4XX series HTTP status codes?

4XX series is corresponding to client errors, i.e., something wrong with the request itself.

- 400 - Bad Request – invalid syntax.
- 401 – Unauthorized – authentication required. Server need to know client's credentials/identity.
- 403 – Forbidden – client does not have access rights to the content. Server knows the client identity, but the client doesn't have required permissions.
- 404 - Not Found - requested resource doesn't exist. (common when hitting invalid URL).

400 & 404 are straight-forward. Whereas 401 [authentication failed] & 403 [permission failed] looks similar but have different meanings.

What's serialization & deserialization in JSON/API Testing context?

In general, when transmitting data or storing it in a file, it is required to be byte strings. Serialization converts objects into byte strings. And recovering original object from the byte string is known as deserialization.

Object > Serialize > Byte strings > Deserialize > Object.

Applying it in JSON/API context,

- **Serialization:** converting objects into Json to be sent in the request.
- **Deserialization:** converting response Json into Objects for storage/validation.

How?

For java, there are different libraries such as - google/gson, FasterXML/Jackson, alibaba/fastjson, etc.

What all things you verify in API testing?

One of the common interview Q. Before that, how does an API [request-response] work OR what components are involved?

- **Endpoint:** the URI + path/query parameters to send the request to.
- **Request-type:** different HTTP methods.
- **Request:** the payload in xml/json/etc. format.
- **Auth:** check for authorization and permission.
- **Response:** response from the server.

Considering these components, what all things you would verify?

- **Schema validation:** The most important. For both request & response, i.e., different fields [type/format/occurrence/etc.] as per the specifications [XSDs]. Different permutations & combinations.
- **Test Data:** Different P&C of Test data in the API request.
- **Authentication:** verifying the auth mechanism [happy-flow/time-out/invalid/etc.]
- **HTTP methods:** which all methods are supported.
- **HTTP Status Code:** Basic check before validating the response.
- **Error conditions:** how well the API handles the invalid inputs.
- Performance & Security aspects.

What are the common Rest authentication methods?

Before jumping to it, let's first understand authorization & authentication,

- **Authentication:** proving your identity. Say a company-issued I-card proves that you are an employee.
- **Authorization:** proving your right to access. With company I-card you can have access to common areas but NOT the security or server rooms. Basically, your permissions.

Some common rest authentication schemes are,

- **Basic:** The most straightforward and easiest method. Use of encoded 'username – password' in the request header.
- **Bearer:** also called 'token' authentication. Token - a cryptic string, usually generated by the server in response to a login request. Once the token is generated, client must send this token in the Authorization header when making requests to protected resources.
- **API Keys:** Generate a key [unique value assigned to each first-time user] >> then use this key the next time you try to access the system.
- **OAuth:** User signs-in, grants permission, and your app can authenticate each request with an access token. E.g., 'Login using Facebook/Gmail' option on many websites OR when you cross-post on any social network | App-A > Facebook Login popup > Enter credentials > Give permissions > App-A now has the token generated by Facebook > this token can now be used by App-A based on the permissions granted.

What all details can be included in a Rest API header?

Headers are mostly classified as request and response headers - property-value pairs that are separated by a colon. Set the request headers when you are sending the request > Set the assertion against the response headers. Some header types,

- **Authorization:** Carries credentials containing the authentication information of the client for the resource being requested.
- **Content-Type:** The MIME type of the request or response. E.g., text/html or text/JSON.
- **Date:** The date and time of the request or response.
- **Accept:** Informs the server about the types of data that can be sent back. E.g., Accept-Charset: which character sets are acceptable by the client.
- **WWW-Authenticate:** Sent by the server if it needs a form of authentication before it can respond with the actual resource being requested. Often sent along with a response code of 401, which means 'unauthorized'.
- **Cache-Control:** Cache policy defined by the server for this response, a cached response can be stored by the client and re-used till the time defined by the Cache-Control header.
- **Expires:** Response header indicating the time after which the response is considered stale.
- **Set-Cookie:** Send cookies from the server to the user-agent.

How to validate response schema?

Schema: specification for the format for defining the structure data, i.e., structure.

Schema validation is an important part of API Testing. It ensures that the response from the endpoint matches with the predefined set of rules. Rest Assured provides this capability using 'Rest Assured Schema Validator'.

- Add 'Rest Assured Schema Validator' maven dependency in your project POM, say for json <json-schema-validator>
- Expected schema: Can be provided by developers for testing purpose. OR just copy the JSON returned by any of the existing API and generate the schema using any online schema generator tool. Save this schema in a JSON file and add it into the classpath of the project.
- Test: As simple as calling a method <matchesJsonSchemaInClasspath> on your response with expected schema file as parameter.

```
RestAssured.given().when().get("endpoint").then().assertThat().body(matchesJsonSchemaInClasspath("JsonSchemaFile.json"));
```

Note: The json-schema-validator can perform fine-grained validation by defining custom configuration rules as well.

Code to include basic Authentication Header.

The basic authentication scheme requires the consumer to send user id and a password encoded in Base64. REST Assured provides an easy way to configure the credentials,

```
RequestSpecification request = RestAssured.given().auth().basic("username", "password");
```

OR

```
RequestSpecification request = RestAssured.given().auth().preemptive().basic("username", "password");
```

Preemptive: Sometimes a server might use a challenge-response mechanism, i.e. wait for the server to challenge before sending the credentials. With preemptive, REST Assured will send the credentials without waiting for an Unauthorized response.

Note: Normally, we do add <preemptive> command to avoid complications and the overhead of making an additional request.

Handling dynamic JSON objects

We can use different options of Jackson library to handle dynamic JSON objects, which have unknown properties.

- **JsonNode:** add it as a field in the POJO class [JsonNode details;] followed by subsequent assert
[assertThat(product.getDetails().get("parametername").asText()).isEqualTo("value"));
]
- **Map:** add a field of type Map in the POJO class [Map<String, Object> details;] followed by subsequent assert
[assertThat(product.getDetails().get("audioConnector")).isEqualTo("none"));]
- **@JsonAnySetter:** use @JsonAnySetter to mark a method [with name and value arguments] for handling unknown properties.

```
Map<String, Object> details = new LinkedHashMap<>();  
@JsonAnySetter  
void setDetail(String key, Object value) {  
details.put(key, value); }
```

#Tip: Wear a Confident Smile

A friendly, confident smile is of the most important things to wear to an interview. While it won't get them to overlook your other expertise, it will work wonders in making you look like a perfect addition to their team. As you give more interviews, you will get used to it.

#Tip: Why are you changing Job so quickly?

If you are looking for a job within a Year – this question will be asked for sure. Be prepared. Be genuine in your reply – what is it that forced you to look out within a Year – it cannot be just your CTC. Unless it is too-low. Your reason should be something which won't impact your candidature at the prospective employer – like some particular policy, or being on bench since joining or too-low CTC, or opposite-skill-set-assignment, etc. Think it through and be prepared.

What is header in API?

Headers are the name or value pairs that are displayed in the API request and response messages.

- **Request Header:** Contains information such as the type, capabilities, and version of the browser that generates the request, the operating system used by the client, the page that was requested, the various types of outputs accepted by the browser, and so on.
- **Response Header:** Includes information such as the type, date, and size of the file sent back by the server, as well as information regarding the server.

Why they are important? Headers define the operating parameters for the process of data transfer from source to destination. They help to know about the source and destination, decide the data transmission mode and encoding type. Headers also speak aloud about the content, its type, its size, restricts amount and variety of data allowed at both ends of communication.

When you will get 422 http error with example?

422 Unprocessable Entity response: Indicates that the server understands the content type of the request entity (hence NOT a 415 Unsupported Media Type status code), and the syntax of the request entity is correct (thus NOT a 400 Bad Request status code), but it was unable to process the contained instructions.

The error occurs when your request data doesn't make logical sense. To fix it, try reviewing your data to verify whether or not you may have incorrectly defined a particular piece of data in your request.

Example: When you send an update request to the JSON API, say redmine/WikiTesting.json with Content-Type: application/octet-stream. Because of this, Redmine is unable to parse the payload since it doesn't know in what format the data is. To solve this, you should always make sure to set the correct content type when posting data. In this case, it should be set to application/json.

Give one example of purge method in API?

In simple terms, the unofficial PURGE method is used for purging caches (remove/delete forcibly).

Background: The goal of caching is never having to generate the same response twice. The benefit of doing this is that we gain speed and reduce server load. The best way to cache your API is to put a gateway cache (or reverse proxy) in front of it.

When a safe method is used on a resource URL, the reverse proxy should cache the response that is returned from your API. It will then use this cached response to answer all subsequent requests for the same resource before they hit your API. When an unsafe method is used on a resource URL, the cache ignores it and passes it to the API. The API is responsible for making sure that the cached resource is invalidated (or purged). I.e., When an API receives a call with an unsafe method on a resource, it should fire a PURGE request on that resource so that the reverse proxy knows that the cached resource should be expired.

What are Header types?

Every REST request must contain three HTTP header fields: Accept, Content-Type, and Cookie.

- **Content-Type:** describes the format, example - JSON (application/json) or XML (application/xml) or etc.
- **Accept header:** describes which format you want a response body. Example, responses can be delivered either as XML or JSON by modifying the Accept header.
- **Cookie header:** contains the authenticated session ID that you obtained after creating a REST API session. Having this header with the session ID allows your subsequent requests to be authenticated. Without it, you will receive a 401 Unauthorized on any request attempt made.

How you will perform put, post and get method in API?

- **GET:** used to request data from a specified resource.

Ex: GET /echo HTTP/1.1

Host: sts.com

- **POST:** used to send data to a server to create/update a resource.

Ex: POST /test/demo_form.php HTTP/1.1

Host: sts.com

- **PUT:** used to send data to a server to update a resource.

Ex: PUT /test/demo_form.php HTTP/1.1

Host: sts.com

#Tip | Why are you looking for a change?

One of the common interview questions – why are you looking for a change? And many people can't think of anything constructive apart from the financial gain. We would suggest a more constructive answer like,

"Changing organization is an important decision. There is no single reason for the Job switch. Professionally I have learnt a lot at my current organization but now the growth & learning has become somewhat stagnant. To move forward, I need to move out of the comfort zone. I am looking for a long-term commitment where there are opportunities for me to learn & grow professionally. On the personal front, (it's time that I marry/I recently got married/recently bought a house/recently became a parent/etc.) – so looking for some financial gain as well."

What is best in API - SOAP or REST and why?

REST allows a greater variety of data formats, whereas SOAP only allows XML. Coupled with JSON (which typically works better with data and offers faster parsing), REST is generally considered easier to work with. ... REST is generally faster and uses less bandwidth.

REST is always better than SOAP in situations that do not require you to fully map a set of objects to the client. Another factor is the simplicity of the REST protocol when compared to SOAP.

What are pre-requests in Postman?

Scripts are piece of code that you can write and let Postman execute it at specific points in your test Lifecycle.

- Pre-requests scripts, which will run before Request.
- Tests scripts: which will run after Response.

Scripts are used to enable dynamic behavior to request and collections. It allows you to write tests, change parameters and even pass data between the requests. A script can be added to request, collection, folder or an independent request.

Note: Scripts in Postman are written in Postman Sandbox.

How would you change the nested part of the JSON in API Automation?

Using JSON-Java library, JSONObject > put will replace any existing value. Simply call

```
js.getJSONObject("Level-1").getJSONObject("Level-2").put("Key", "Value");
```

Why we use put for login API instead of get?

- **GET**: used to request data from a specified resource.

Ex: GET /echo HTTP/1.1

Host: sts.com

- **PUT**: used to send data to a server to update a resource.

Ex: PUT /test/demo_form.php HTTP/1.1

Host: sts.com

Since login API needs username/password OR other authentication details, we are effectively sending the data to a server and NOT just reading/requesting the data.

How to pass query param with GET method in Rest Assured?

```
@Test  
public void validateQueryParamInGiven()  
{  
    Response resp = given().queryParam("page",  
"2").when().get("https://sts.com/api/users");  
    assertEquals(resp.getStatusCode(),200);  
    System.out.println(resp.getBody().asString());  
}
```

How to pass header for GET method in Rest Assured?

```
@Test  
public void validateGivenHeader()  
{  
    Response resp = given().header("Content-Type","application/json").  
when().get("https://sts.com/public-api/users");  
    assertEquals(resp.getStatusCode(),200);  
    System.out.println(resp.getBody().asString());  
}
```

What are the components of an HTTP request?

- **Action** showing HTTP method like GET, PUT, POST, DELETE.
- Uniform Resource Identifier (**URI**): URI is the identifier for the resource on the server.
- **HTTP version**: Indicate the HTTP version like- HTTP V1.1.
- **Request Header**: Request Header carries metadata for the HTTP request message. Metadata could be a client type, format supported by the client, format of a message body, cache setting etc.
- **Request Body**: Resource body indicates message content or resource representation.

Why is API testing considered better than UI testing?

API testing is now preferred over GUI testing and is considered as most suitable because:

- API testing provides access to the application without the user interface. The core functionality of the application will be tested before the GUI tests. This will help to detect the minor issue which can become bigger during the GUI testing.
- Easy Integration with GUI: API tests provide highly integrable tests which is useful to perform functional GUI tests after API tests. E.g., simple integration would allow new user accounts to be created within the application before GUI tests are started.
- API Test automation is more stable, easier to maintain and provides fast feedback.
- API test requires less code so it can provide better and faster test coverage compare to GUI test automation. This will reduce the cost for the testing project.

#Tip: Why should we hire you?

This can be a stumper, if not thought out before the interview. However, if you prepare, it can turn into a golden opportunity to give a winning elevator speech. Summarize your best points, and try not to feel uncomfortable by being overly modest.

Use Examples - “From the job description, I understand you are looking for someone who can take your automation team to the next level. During my time with ABC, I performed a similar task, with excellent results...”

What is payload in Restful Web services?

The “payload” is the data you are interested in transporting, i.e., the request/response body. This is differentiated from the things that wrap the data for transport like the HTTP/S Request/Response headers, authentication, etc.

What is the upper limit for a payload?

GET generally appends data to the service URL. But its size shouldn't exceed the maximum URL length. However, POST doesn't have any such limit. So, theoretically, a user can pass unlimited data as the payload to POST method.

But, if we consider a real use case, then sending POST with large payload will consume more bandwidth and will also be more complex to manage any changes. It'll take more time and present performance challenges to the server. Hence, a user should take action accordingly.

If child/target API is not available, how do you test your functionality?

Mocking: A mock API server or mock server API imitates a real API server by providing realistic mock API responses to requests. They can be on your local machine or the public Internet. Responses can be static or dynamic, and simulate the data the real API would return, matching the schema with data types, objects, and arrays.

There are many API mocking tools available - Nock, WireMock, MockServer, Mocky to mock API responses. Even Postman has a mock server if you want to use that.

Rest-Assured: Sample code to send a post request.

POST is used to send data to the Server. E.g., POST request to submit form data (HTML forms) on a web page. The data is sent in the body of the HTTP request.

Create a Request pointing to the Service Endpoint

```
RestAssured.baseURI = "https://restapidemo.com/customer";
RequestSpecification request = RestAssured.given();
```

Create a JSON request which contains all the fields

```
JSONObject requestParams = new JSONObject();
requestParams.put("FirstName", "STS"); requestParams.put("LastName", "Studio");
requestParams.put("UserName", "testingstudio"); requestParams.put("Password",
"password1");
requestParams.put("Email", "softwaretestingstudio@gmail.com");
```

JSONObject is a class that is present in org.json.simple package.

Add JSON body in the request and send the Request

```
request.header("Content-Type", "application/json");
request.body(requestParams.toJSONString());
Response response = request.post("/register");
```

Any idea about Apache HttpClient vs Rest Assured?

Overview: REST Assured is a high-level Java DSL for simplified testing of REST based services built over HTTP. On the other hand, Apache HttpClient is a low-level client for simplifying Http Communication, In-fact HttpClient is used by REST Assured under the hood for Http communication.

Rest Assured has always primarily been targeted towards testing hence offer many additional features that are simple to use when automation Rest API testing. Example,

- Validating REST API response using inbuilt Hemcrest Matchers
- JSON & XML serialization and deserialization
- Extracting JSON data using JsonPath and XML data using XmlPath
- Verifying response body, cookies, headers, content-type and http status
- Authentication using Basic Auth, Digest Auth, Form Authentication (CSRF support), OAuth (OAuth1 and OAuth2)
- Verifying multi-part form data
- Request and response logging for easy troubleshooting
- Session Filters

Note: REST-assured is a wrapper around Apache HTTP Client, focused towards API Test Automation.

What's the difference between API Request Header – Content vs Accept?

Accept and Content-type are both headers sent from a Client (say a browser) to a server.

- **Accept** header is a way for a client to specify the media type of the response content it is expecting. i.e., expected response media type.
- **Content-type** is a way to specify the media type of request being sent from the client to the server. i.e., request media type.

Any idea about Options method in API requests?

The OPTIONS method represents a request for information about the communication options available on the request/response chain identified by the Request-URI/URL/server.

In simple terms, if you want to know what all HTTP methods does an API support – send a Options request first.

OPTIONS /index.html HTTP/1.1

*OPTIONS * HTTP/1.1*

The response then contains an *Allow header* that holds the allowed methods:

HTTP/1.1 204 No Content

Allow: OPTIONS, GET, HEAD, POST

Cache-Control: max-age=604800

Date: Thu, 13 Oct 11:45:00 GMT

Server: EOS (lax004/2813)

Rest-Assured: How to read node values from Json?

There are many libraries available to work with Json parsing. Here we are using **Jackson API** for Java.

Use class **ObjectMapper** provided by Jackson API > method **readTree()** to deserialize JSON content as tree expressed using a set of JsonNode instances.

```
ObjectMapper objectMapper = new ObjectMapper();
JsonNode jsonTree = objectMapper.readTree(jsonObject);
```

Once done, we can easily get the value of any node using **get() and path() methods** of JsonNode class.

Normal JSON,

```
String firstName = jsonTree.get("firstName").asText();
```

OR

```
String firstName = jsonTree.path("firstName").asText();
```

Nested JSON,

```
String javaLevel = jsonTree.get("skills").get("Java").asText();
```

OR

```
String javaLevel2 = jsonTree.path("skills").path("Java").asText();
```

JSON Array,

```
String firstName = jsonTree.get(0).get("firstName").asText();
```

OR

```
String firstName = jsonTree.get(1).get("firstName").asText();
```

Note: Creating POJO classes for parsing a JSON to fetch values may not be easy all the time especially when you have lengthy nested JSON.

& This brings us to the end of your interview preparation.

Below are the Interview Experiences of 50+ QA-Testers across Start-ups & MNCs, which will give you a fair idea about the interview process, rounds, commonly asked Qs and some pro tips!

Hope the book was of some help in your preparation and answered few of every job seeker's queries.

Would love to know your feedback, or a list of Qs that you want to be included – just drop me an email at deepanshu@parikshram.com!

Join me in building the first dedicated platform for QA-Testers – <https://parikshram.com> – by spreading the word in your friend circle.

Enabling Success, together
Deepanshu Agarwal

Interview Experiences

Glossary

EPAM SYSTEMS | SDET I | Gurgaon | 3 rounds | 5 years

Two rounds of interview with initial codility programming test.

1st Round quite unique, very lengthy and detailed discussion lasts around 100-110 Mins involves almost all the phases and processes of Automation and testing.

2nd round is more of a managerial cum technical discussion if you are here, it means your 80% tasks is done, this round is more about your management skills, durability, troubleshooting and exploiting RCA's and processes.

1st Round,

- Explain about your framework, did you created it or simply worked upon.
- Explain how data, reporting and test suits are derived in your framework.
- Coding problem regarding frequency in string and output based on frequency.
- What's the use of static keyword?
- What are OOPS concept where have you implemented them in your framework
- Have you used Interfaces and encapsulation in your framework?
- How did you handle reports and logs in your framework?
- What is Static Keyword
- Explain POM.
- Write POM class for a login Page.
- How to handle frames and window-based pop ups in selenium.
- What is Authentication and Authorisation
- Explain HTTP protocols.
- Write automation script for login POST request using REST Assured. (Hardcoded)
- What are your achievements till now in your projects?
- Have you implemented solid principles in your codes?
- What tools have you used for reporting and tracking
- What is the bug life cycle?
- Explain STLC
- Write test data for a sample flight booking POST request.
- Explain your sprint cycle and activities.
- Have you worked on any CI/CD tool, how do you create Jobs and Nodes in Jenkins?
- Basic git commands

2nd round was more a managerial discussion but again it depends how your first round goes, based on that they may ask few more technical questions, it also depends on the person in front.

- Introduce yourself
- Explain your team structure and hierarchy
- What clients have you worked for
- Why do you want to change your company?
- How many env did you handled

- You get a mail in the morning your test env is down as a lead what will you do next.
- Explain some critical issues and challenges your projects have faced
- What are the things you would want to achieve with EPAM?

Few more questions were there related to project and skills, this is what I remember now.

Tips: EPAM has one of the best hiring processes, detailed interviews which are recorded and later on reviewed (Yes, they review it) and then your next rounds are decided, prepare well for your first round get a grip of not only theory but concepts and if you are appearing for a experienced role hands-on and real time troubleshooting does matter. Footnote: I was rejected in 3 Interviews back-to-back while starting, while ending I had around 5-6 offers just keep persevering and never give up, it's fun once you pass that thresh hold. All the Best :)

FIS | Automation testing | Bangalore | 2 rounds | 3.5 years

It was good but bit in details.

- Tell me about yourself
- Explain the framework
- Write the code for data driven
- Full form of XSSF
- Write without Fileinputstream in the data driven framework
- What code written inside the Chromedriver()
- Explain Selenium architecture
- Swap two string with and without using empty string
- We have two web tables and Xpath is same but in table one particular web element is there I want to extract that using existing Xpath write whole Xpath code
- Have you worked on database testing?
- Write code for Firefox driver invocation with set properties
- Write CSS selector for child to parent (it's not possible in Selenium)

Tips: Be prepared they are asking in detail questions.

CHETU | Software Tester | Noida | 3 rounds | 3 years

I was asked to appear for the written test which consists of 54 MCQs covering aptitude, reasoning, maths, testing technical questions to be answered in 40 minutes. I successfully passed that round then the second round was presented to me with 10 questions. All these were scenario-based questions and 25 minutes was given to me for this round. Then my third round was scheduled the next day and it was taken by test lead and they asked various questions on testing.

- Introduce yourself.
- What is manual testing?
- What is API testing?
- What is bug life cycle?
- What is the difference between priority and severity?
- What is the difference between verification and validation?
- What is black box testing?
- What is white box testing?
- What is compatibility testing?
- What are the test cases of a dropdown box?
- What is the difference between test case and test scenario?
- What is test planning?
- What is exhaustive testing?
- What is database testing?
- What is DML?
- What is adhoc testing?

EPAM SYSTEMS | Senior Quality Engineer | Bangalore | 1 rounds | 5 years

Technical round - 85 - 90 minutes

- Tell me about yourself
- What is the project you have worked on?
- What is the difference between smoke and sanity testing?
- What is the re-testing and Regression Testing?
- What is Defect density?
- Difference between Test Strategy and Test Plan?
- What is Usability and UI?
- What is Test Metrics and Test Estimation?
- What is Test Deliverable and Test Artifact?
- What are all the Test Design Techniques?
- What is Agile Process in your current company?
- How you will estimate story points in Sprint Planning?
- What is difference between Authentication and Authorization?
- What are the HTTP Methods in API?
- What is the difference between UI and API Testing?
- What are all challenges that you faced in your experience?

Tips: Please prepare well.

HCL | Test Lead | Bangalore | 4 rounds | 8 years

There were 4 rounds apart from normal HR discussion.

- 1st round was the shortlisting round so quick round of basic Qs on testing, java and selenium.
- 2nd & 3rd rounds were details technical rounds on automation, framework, API testing, situational, team management, estimations, agile, SQL, etc.
- 4th round was managerial, mostly related to project, roles, behavioral and management Qs.
- Last HR discussion

Questions,

- WAP to count duplicate character in a string.
- WAP to find 2 no with max product.
- SQL queries involving join
- SQL theoretical Qs
- Have you used Jenkins? How? What's the execution cycle?
- How to group cases in TestNG
- Challenges faced during automation and how did you handle it.
- Common exceptions that you have faced?
- What is your current role & responsibilities?
- Java exception hierarchy
- API methods, header, response validation, etc.
- Have you ever managed a team? What's the challenge?
- Any experience with performance? or security? or any other type of testing?
- Explain your framework
- How many test cases are automated, what the cycle?
- Tell me about your Sprint activities.
- What if you have to drive test automation efforts? and manage a team?
- Ever had short timeline? How did you handle
- What is the biggest challenge you faced till now?
- What all agile events do you follow?
- Why are you looking for a change?
- Any experience with Cucumber? What is step definition, feature file, etc.
- How do you handle stale element exception?
- What measures will you take to speed up test automation execution?
- Domain specific Qs
- What if you don't get a chance to work on test automation during initial month?
- What is your dream role? What are your expectations from this role?

Tips: Overall preparation. Practise java programs, and prepare other topics as well. All the best.

TRICON INFOTECH | Associate Test Engineer | Bangalore | 3 rounds | 4 years

There was a 3 round of interview, it was conducted thought Microsoft Team. Questions was asked based on Manual, SQL, Java and little bit Selenium.

- STLC, SDLC, Test case and Real Time Scenario based Questions in Manual Testing.
- Queries and about joins in SQL.
- OOPS concept in java.
- Basic questions in Selenium.

Tips: Confident and Hard work

UNITED HEALTH GROUP (OPTUM) | Sr. Quality Engineer | Noida | 3 rounds | 4 years

- 1st round is HackerRank test consisting of 2 programming questions in which you have to write a code and pass the test cases and MCQ questions on Selenium and Java basics.
- 2nd round: Core Technical round based on Java, Selenium, API.
- 3rd round: Manager Technical round.
- Last HR discussion

Questions,

- WAP to count number of characters in a string
- Interviewer will give some code and asked what the error in the code
- Interviewer will ask output for some code.
- WAP to count duplicate character in a string
- Sql queries - join, update
- How to setup a pipeline using Jenkins
- TestNG questions
- How to handle multiple windows using selenium
- Why we use try and catch
- What are different methods In API
- Difference between put and post
- Why we do API testing
- What are the different listeners in Jmeter for performance testing?

Round 2,

- How you resolve conflict if the developer not agree and says it's not a bug
- What you do when you don't have much time to test and in EOD the build is going to deliver to client
- What are the challenges you face in these 4 years in career and how you resolve them?

- What is agile, how it's different from waterfall
- Have you face any challenges in Agile?
- How you handle pressure?
- what are your strength and weakness?

Tips: Be prepared with Java and Selenium and should know each & everything written in resume. Question mostly asked on basis of that. Java is compulsory, should able to guess output of program. Should have knowledge about HASHMAP concept, able to write basic program in java . All the Best :)

EMMES INDIA | Senior SQA Engineer | Bangalore | 3 rounds | 5.7 years

Interview started with the question 'Tell me about yourself' and what is your roles and responsibility, then questions started coming from project related

- What to do you think why more issues is raised in UAT?
- Tools used in your project
- Defect life cycle
- What details you will put while raising bug?
- Scenario given for raising bug
- Which logs will be attached while raising bug?
- What is Selenium?
- What is WebDriver?
- Difference between explicit and implicit wait.
- How to locate WebElement?
- Difference between XPath and CSS selector.
- Text is used in which locator XPath or CSS.
- One scenario-based question was there to create possible testcases.
- What will you do if same issue is raised at offshore and by UAT team?
- Some questions were from banking domain
- What automation framework you use?
- Difference between find element and find elements.

Round 2,

- Defect life cycle
- Tools used in your project
- What details you will put while raising bug
- How much you will rate yourself in java from 1 to 5.
- What automation framework you use.

HR Round,

- Tell me about yourself, how your professional journey started in Testing.
- What motivates you in testing.
- Why do you want to join us?
- If I will ask to your manager then what weakness about you, he will tell.
- If I will ask to your manager then what strength about you, he will tell.
- How will you handle if frequent modification of testcase, if product is already implemented and release is very near?
- Any certification done in last one year?

Tips: Please prepare well as per job description and better know about your project.

RELIANCE JIO PLATFORM | Software Testing Engineer | Bengaluru | 4 rounds | 3 years

I got a call from the recruiter for Written Test / Coding round. All the process were quick and transparent. The interview was scheduled as per my availability and concern were addressed.

- Round 1 - Written Exam / Coding Round over HackerEarth.
- Round 2 - Technical Interview 1
- Round 3 - Technical and Managerial Interview
- Round 4 - HR Interview
- After I had my HR round and discussion, was offered by the company within a week.

Questions,

- Round 1: Coding round: There were 10 MCQs based on testing basic questions and 2 problems solving, where you can write the code in any programming language. It happened in HackerEarth platform.
- Round 2: Technical round with basic questions on software testing. Questions were from Selenium, API testing and TestNG framework. Be thorough with the interview questions, as they can ask any question.
- Round 3: Asked about my experience and few basic testing questions.

Tips: Be prepared for the interview.

DELOITTE | Salesforce Tester | Hyderabad | 2 rounds | 4 years

Overall, it was good. Most of the questions were scenario based and experience based. Two questions were situation-based.

Questions,

- Tell me about yourself
- How many years of experience in Salesforce testing?
- What is Profile and Role?
- Can two users have same Profile?
- Can two Profiles have same user?
- What is SSO and what is the use of it?
- How do you rate yourself in Salesforce out of 10 or 5 scale?
- Based on the above question's answer, they asked me where do you lack the remaining?
- On which cloud you have been worked?
- How many objects were there in SF and explain about them?
- What are Reports and how many Reports are there in SF?
- What are Permission Sets?
- Can we assign two roles for single user?
- Can we delete a user in SF?
- What custom objects you have used in your current project?
- What is SDLC and STLC?
- Which type of testing will be done first?
- How good are you in SQL?
- What are Joins and few examples of JOINS?
- Defect life cycle?
- If there are only 2 members in a team and the US is deployed just before 2 days of the end of the sprint and the resource is absent. How will you manage the work?
- Difference between Regression testing and Retesting?
- When you will perform smoke and when you will do Regression?
- Will you do UAT?
- Scrum Ceremonies?
- What is your role in Scrum Planning?
- What is Retrospective meeting?
- Apart from testing what kind of activities you will perform?
- What is RTM and what kind of metrics you will be tracking?
- Questions on Test management tools that i worked

2nd Round:

- Overall experience (exactly)
- Define Sales process
- Different kind of defect stages
- Did you handled team, if so, what challenges you had faced?
- Do you know automation? If we put you in automation testing, can you do well?
- We have different Clients working on different time zones. Will you be able to work late night?

Tips: In order to confuse, interviewer will repeat the question and answer which has been told. So be confident in what you say and stick to your answers. Know about your project thoroughly as the main questions will be on the current project.

COMMCE | Test Head | Bangalore | 1 rounds | 10 years

Interview was started with HR round. They recorded the video call while taking interview. Interview was simple.

- Introduce yourself.
- How much are you excited about the role?
- What's your aspiration?
- How many people worked under me?
- Which side would you prefer, people side or technical side?

Tips: Prepare well for taking Managerial responsibilities.

COGNIZANT | Test Engineer | Chennai | 2 rounds | 6 years

- Self-introduction
- Alpha and beta testing
- Agile working experience and related questions
- Jira and related questions
- Domain knowledge questions
- Advantages of TestNG
- What is POM
- TestNG annotation hierarchy
- How to use groups in TestNG and how to use
- Format of TestNG
- Xpath in Selenium
- Locators in Selenium
- Other basic Selenium questions
- Diff b/w RPA and Automation

Tips: Be confident and cross-question. Be strong in basics.

HCL | Scrum Master | Bangalore | 5 rounds | 7.5 years

There was total 5 rounds of interview conducted. 2 Technical rounds of Interview, 2 Technical rounds of interview conducted by Client followed by 1 managerial round. Interview process was very smooth and it took nearly 2 weeks to complete the process and get the offer letter.

Tips: Thorough understanding of the subject line.

ACCENTURE | Test Engineer Senior Analyst | Pune | 2 rounds | 5 years

Well to say general preparation and confidence is the key. While answering the questions make sure you answer about the usage of that method. E.g., If you're explaining Inheritance or polymorphism, try to answer what is the use of that in actual project. As well try to provide sufficient information how it is beneficial for project.

Questions: All general questions

Tips: Logic.

TECHBLUE SOFTWARE | QA Automation | Gurgaon | 2 rounds | 5.5 years

Interview started with the question 'Tell me about yourself' and day to day activity Later moved on with the questions after knowing my technical skills.

- What is verification and validation
- How to write testcases
- What is traceability matrix
- How to raise defect
- How to write testcases
- Have you written any testcases without document? If required then how will you write?
- What is interface in Java
- What is inheritance in Java
- How to implement interface
- How to extend class
- Is it possible to extend 2 classes together?
- Why we use interface
- What is Xpath
- Types of Xpath
- Explain both (relative and absolute) Xpath
- Explain assertion and verify.
- SQL joins
- How you will fetch data from excel.

- Have you worked on TestNG?
- What are the annotations?
- Explain the annotations.

Round 2,

- POM model
- Assert and verify
- Which locator you prefer – Xpath or CSS
- Explain both (relative and absolute) Xpath
- Have you worked on TestNG?
- Webdriver API
- How you are triggering your suite using Jenkins?
- Have you used javascript in your code - if yes, where used in code?

Tips: Please prepare well as per job description. Keep basics clear.

HARMAN | Lead Engineer | Bangalore | 1 rounds | 8 years

Interview started with the question 'Tell me about yourself'. Later, moved on with the questions after knowing my technical skills.

- Which is the latest version of Windows?
- Definition of Acceptance testing and Regression testing.
- What are Agile and Waterfall models?
- What does a Test case include?
- Explain Defect life cycle.
- Where do you raise bugs?
- What are the Debugging tools used?
- When a defect is raised, when is it a tester's responsibility?
- Important issues, how have you solved them?
- Roles and responsibilities on a day-to-day basis.

Tips: Please prepare well after reading through the job description. Very important.

ACUCERT LAB | QA Automation | Mumbai | 3 rounds | 4 years

- Round 1 - Introduce yourself. What activity you do in your day-to-day work. Are you comfortable with Python? I got one assessment for which I had to write unit test case for a program and found out bugs in the program. The original program shouldn't be affected and you have to write a script which will test the program.
- Round 2 - Assessment Test on program.
- Round 3 - Technical Round

Questions,

- How to take screenshot using selenium.
- What are the different locators used in Selenium?
- I was given link of redbud app and I had to write the automation test case till the user books the ticket this was very tedious task as it took around 1 hour for me.
- TestNG framework.
- Arguments used for driver.
- Jmeter.

Tips: practice well for programs and Selenium as on spot selenium programming will be very difficult to recall sometimes.

EPAM SYSTEMS | Software Test Automation Engineer | Hyderabad | 2 rounds | 3 years

The recruiter was so professional and told me that I will be having two rounds (Technical round and Techno-managerial Round) and If I get selected my offer will be rolled out within a day.

- Technical round - 85 - 90 minutes
- Techno-managerial Round - 55 - 60 minutes.

Questions,

- Tell me about yourself.
- What is the project you worked on?
- Project related question why did you opt for a particular licensed tool over various open-source tools in market?
- Explain oops concepts with examples.
- Difference between abstract classes and interface.
- Why multiple inheritance is not supported by java.
- How to overcome multiple inheritance problems in java.
- Gave a java program related to inheritance and asked me to the output.
- How to execute tests parallelly without the help of TestNG?
- What is Docker and difference between docker and selenium grid.
- Explain BDD framework and its significance.
- What are hooks in BDD and where to use it?
- Usage of data provider in BDD.
- How to run a testcase with multiple data in BDD.
- What is feature file and step definition file and how to integrate both.
- Difference between arraylist and LinkedList?
- Write a java program to remove duplicate values in a static array.
- Difference between hash map and hash table.
- What is API and what are different types of authentication methods used in API.

- How will you integrate Jenkins and git?
- How will you have multiple windows handles in a browser using Selenium?
- What are the different types of waits in Selenium?
- What are all locators present in selenium?
- Basic git commands.
- What is maven and why do we use it.
- Gave me a scenario to automate - There are three dropdowns Country dropdown, State dropdown, city dropdown with select tag. State dropdown appears into DOM once country is selected and city dropdown appears into DOM once state is selected. so, once city dropdown appears print out the values in the city dropdown which starts with A or a.
- Explain about defect life cycle.

Techno-managerial Round,

- Tell me about yourself.
- Why are looking out for job even though you have multiple offers.
- Who will decide which testcases to automate and which testcases not to automate?
- How testcases which needs to be automated are assigned to a resource.
- Explain STLC.
- Difference between driver.quit and driver.close?
- Disadvantages of Selenium.
- Gave me a java program and asked me to tell the output and also explain each and every line of code.
- Write a program to swap two numbers without a temporary variable.
- What are the fields which are necessary when raising a defect?
- Difference between priority and severity of the defects.
- Explain agile methodologies.
- How to automate captcha.
- Are there any scenarios where you are not able to automate in an application?

Tips: Never ever feel dejected if you don't get a call back. Getting a new job or switching to new job from existing job is a process in itself and it takes time. Please be confident and keep on updating yourself. It is important that you stay motivated in this process and don't ignore any job interview how bad the company profile is. Even though if you don't clear the interview the experience will be very helpful in the coming interviews. Each and every interviewer is asking for oops concepts and real time examples of oops concepts and expecting every hook and nook of selenium and other framework if you are an experienced candidate doesn't matter if you have relevant experience or not.

ORACLE FINANCIAL SERVICES | QA Automation | Bangalore | 2 rounds | 3 years

The total interview process will consist of 2 second (Technical Round and Managerial Round).

- Technical Round went on for 60 Mins.
- Managerial Round went on for 15mins.

Questions,

Technical Round:

- Tell me about yourself.
- Frameworks and tools used in your experience.
- Explain OOPS concept in detail with examples.
- Gave me a scenario to automate and asked to write the script in VBscript and Selenium.
- Difference between severity and priority of the defects.
- Gave me a java program and asked me the output.
- How will you establish a database connection in UFT and Selenium?
- Will you work in any other technology other than Selenium?
- How did you handle exceptions in your project?

Managerial Round:

- Tell me about yourself.
- Explain different frameworks you worked on.
- Given an opportunity are you willing to work in manual testing also.
- Willing to relocate and salary discussion.

Tips: Never ever feel dejected if you don't get a call back. Getting a new job or switching to new job from existing job is a process in itself and it takes time. Please be confident and keep on updating yourself. It is important that you stay motivated in this process don't ignore any job interview. Even though if you don't clear the interview experience will be very helpful in the coming interviews. Each and every interviewer is asking for oops concepts and real time examples of oops concepts and expecting every hook and nook of selenium and other framework if you are an experienced candidate doesn't matter if you have relevant experience or not.

INFOSYS | Test Analyst | Pune | 2 rounds | 3.7 years

Interview was good, first i have attended the technical round, in the panel there were 2 Technical persons and both are sounding good on terms of technical skills. Overall experience was awesome. We need to stick to our basics and we need to be sure on our daily activities.

Questions,

I have attended for Test Analyst role and all the questions were related to Testing, Testing Concepts, Testing Methodologies and main focus was on Agile Methodologies questions and they have asked me to derive scenarios for some examples.

Tips: Be clear on your basics, all questions will come from our daily activities itself.

PUBLICIS SAPIENT | QA Automation | Bangalore | 3 rounds | 3.8 years

There were three rounds, first technical followed by Manager round and HR round. All were very professional and interactive. My overall experience interviewing in Publicis Sapient was good.

Questions

Round 1: Technical

- Asked about final keyword and sent one program and asked output for same.
- Asked about constructors and types
- Asked to write program for reversing a string.
- Asked to write program to eliminate repeating character from string.
- Asked about static keyword, where we use and purpose
- OOPs concept and where I implemented in my framework
- In selenium, she asked me regarding windows handling
- Asked about assertion and types
- Asked how to handle drop-down
- Parallel execution of test cases in TestNG
- Asked questions on DDT framework (Data Providers)

Round 2: Manager Round,

- Asked about my project and roles
- Why I am looking for change
- Asked me any challenge (can be related to work or team member) I faced during my working span
- What are your career goals, where I wanted to be in next 5 Years?

- Asked my ratings in past 3 Years of my work
- Gave me hypothetical situation and asked me what will be my approach

Both rounds were approximately for 40-60 Minutes.

Round 3: HR,

As we already discussed about salary expectation on our first call, so HR round was more about explaining company culture, Compensation breakup explanation, benefits.

Tips: Be confident about your framework and explain as deep as you can. Focus more on basic topics that we leave in Java, Selenium and TestNG or whatever framework you are using.

FISERV | Specialist, Quality Assurance Engineering | Noida | 3 rounds | 5.5 years

3 rounds, 2 technical & 1 with client.

Technical rounds were hectic. There were 2 interviewers, one was asking questions related to UFT/QTP Automation tool and VBScript programming, Framework design, loading of function libraries & object repositories during run time, identification of dynamic objects, real project based problems, loading of Excel sheets, debatable operations , difference between selenium and UFT, some vb script programs like string reversal using loops and conditions, printing repeatable substring inside a string.

The other interviewer was asking questions completely related to my current project, roles and responsibilities. Difference between test strategy and test plan, advantages and disadvantages of agile and waterfall, which one is preferable, different scrum ceremonies, testing techniques, story point techniques, defect life cycle, how to handle a non performing team member, how to mitigate defect leakages, how to do root cause analysis, defect log etc.

The client was a US citizen and the coolest guy I have ever met in an interview. He started with a joke involving him and his wife and then made me feel really comfortable for discussion. Shared his travel story to India, discussed current pandemic situation. This went for 10 minutes. Then he asked me about my motivation and what makes me wake up every day, how do I solve problems, wats the important Qualities of a good leader, how would I lead a team towards a goal, what sorts of solutions have a I provided to a project problem using my skills and abilities. Then he explained about the project under him, the challenges and what are my responsibilities and expectations going to be. The interview went for 45 minutes. In the end he provided me a compliment that I have a right attitude and he would put on good words for Infront of the team.

Tips: Know well about your current project, roles and responsibilities, identify problem areas in your project and how you could have solved it. Be aware of Automation framework and tools. Have the positive attitude and be genuine with your answer.

TRUGLOBAL | SALESFORCE SR. QUALITY ENGINEER | PUNE | 4 rounds | 8 years

There was total 4 rounds: Technical, Another technical, Managerial, HR. The requirement was for salesforce testing. Below are the questions asked in technical rounds:

- Difference between Profile and Permission set
- Difference between Role and Profile
- Type of sharing settings in salesforce
- About sharing rules and public groups
- Types of sandboxes in salesforce
- Difference between Method override and method overloading
- Difference between Abstract class and Interface
- selenium code to login to a page
- How can you automate process in Salesforce - using workflow?
- Types of relationships in salesforce - Lookup, master detail and Junction
- Difference between lookup and master detail
- What is roll up summary field
- What are record types
- Types of reports in salesforce
- How can you handle dynamic elements in Selenium?
- Challenges in automation
- Challenges in salesforce testing
- Sales cloud flow in salesforce
- What is salesforce
- Is salesforce a public cloud or private cloud
- What is multitenant architecture in salesforce
- Where is data stored in salesforce
- How you test bulk data in salesforce
- What type of errors you get while loading test data in salesforce?
- Types of custom settings in salesforce
- Any experience in Integration testing
- Any experience in API testing
- How you test Approval process
- Types of Standard profiles in Salesforce
- What are Standard objects in Salesforce
- How to give access to reports in salesforce
- What is KPI for automation
- Types of testing - Unit testing, System testing, End to End, UAT, Black box, Functional, white box, sanity, regression, Integration testing
- Defect life cycle
- Explain Severity and priority with examples in defect cycle
- SDLC life cycle

- Difference between @beforetest and @beforemethod in TestNG
- Annotation flow in TestNG
- What is Entry and Exit criteria in testing
- How you make sure 100% test coverage

Managerial round questions,

- Are you a Team player or Team leader?
- What are the challenges you had faced?
- Your weakness
- How will you mange work when team member is on leave
- Do you have interaction with clients and how frequently?
- Tell me about yourself
- Why you want to join our company
- Why are you leaving your current company?

Tips: Study salesforce concepts thoroughly for Salesforce testing role.

MINDTREE | Senior Test Engineer | Bengaluru | 3 rounds | 3.2 years

Mindtree interview will be bit harder compared to other service-based companies.
Interview had 3 rounds: Technical, Managerial, HR

Questions

- Difference between final, finally and finalize
- Oops concepts and the multiple inheritance concept. How to achieve multiple inheritance?
- TestNG annotations, Priorities of TestNG tests, Order of execution of tests, Groups, Group of groups, Include and exclude concepts, Data provider, Invocation count, Parallel testing
- Cucumber: What is BDD and TDD, Types of frameworks, how is keyword stored in Keyword driven framework, Data handling code using Apache POI, Scenarios, Scenario outline, Background, Examples keyword, Testrunner file, Plugins in Testrunner, Monochrome in Testrunner, DryRun in Testrunner, Strict in Testrunner, Glue in Testrunner, what is Maven and how to skip the scenarios in Cucumber, Hooks concepts.

MANAGERIAL ROUND: It was some scenario-based questions, latest tools used, Collection concepts.

HR Round: General discussion about salary

Tips: Be prepared with your framework explanation what you do in your company.

NETAPP | QA lead | Bangalore | 5 rounds | 10 years

The interview was mostly related to file systems and automation. Interviewers were looking for in depth knowledge of file systems. Testing storage products, fundamentals of python.

- What do you know about file systems?
- Write a python program to give minimum number of denominations out of 10 5 and 20 for any required amount

Tips: Solve HackerRank problems.

VIRTUSA | QA Automation Selenium | Hyderabad | 2 rounds | 3 years

First of all, I would like to thank for this amazing Platform. It is the great Trusted platform.

I had 3 Years' experience on QA AUTOMATION SELENIUM at TCS HYDERABAD. And in my company 90 days was the Notice period and most of the company calls which I got asking for immediate joiners, max one month. So, I decided to keep resignation without any offers at hand. I decided to take the challenging mode. And then I started attending the interviews almost till now I attended around 40 companies and at first two months continuously I am in rejection modes at last level or at first level. But I didn't lose any hope and at 3rd month of my notice period I got the great confidence and I started attending the interviews again and two weeks back I got selected for two companies "VIRTUSA" and "VALUELABS". With great package almost above 100% hike i.e.; from 4.5 LPA from TCS to 10 LPA to VIRTUSA. Have a faith on you and start preparing the concepts of Selenium, Appium, Rest API and core Java. Start note down all the interview questions which were asked by them.

Selenium,

- Switching between the tabs (Get Window Handle and Get Window Handles and Return Type of both Handlers).
- Element Locators: Xpath, CSS Selector etc. (learn all locators)
- Wait times (Implicit, explicit, Fluent waits) and Differences and their Syntaxes.
- Challenges faced during automation.
- When will you decide the particular test case to be automated?
- Get, Navigate Methods.
- Action classes, Alert Classes, Auto It, How to Download and upload using Selenium.
- Selecting values from Dropdown.
- Differences between IDE and Webdriver.
- How to initialize the Webdriver and what is the Webdriver.
- Handling exceptions in Selenium.
- Entire about TestNG And their Annotations grouping, prioritizing. How to achieve parallel execution.

- Version control system like GIT and Commands like Rebase, commit, pushing, pulling and creating the branch name.
- CI/CD tools like Jenkins (learn this thoroughly)
- How to avoid Merge conflicts like pipeline is getting failed why and how.

APPIUM,

- How to find elements and asked about native, hybrid and web apps.
- Simulators, Emulators and Real devices,
- Native Frameworks like XCUI Automator for IOS and UI Automator for Android.
- ADB Commands.
- Rest API: Response codes.

Core Java,

- Oops concepts and their usage in selenium automation.
- Static, Final, Finally, Finalize, Try, catch, Throw and Throws and their differences.
- Abstract and Interfaces differences.
- Entire about collections like list set map and Hashmap, Hashtable differences.
- Why string is immutable and difference between string buffer and string builder.
- Constructor and in oops method overriding and over loading.
- Association, Aggregation and composition.

Coding,

- Reverse of a string.
- Removing the duplicate characters from a given string.
- Star programs.
- Factorial of a given number.

Tips: All the very good luck to all. Stay Safe and strong!

NTT DATA SERVICE | Senior Tester | Bangalore | 4 rounds | 3.2 years

The overall Interview experience was awesome.

- 1st round was technical interview where the questions were based on Java, Selenium waits, Iframes, Window handles and main comes the Page object model.
- 2nd round was Managerial round: The questions were based completely on the latest automation tools in the market, why automation only, how do you handle the conflicts and the task within the deadline.

- 3rd round was Client round: I got an exposure to talk to the client. It was a very beautiful experience as they were asking about the tools which I had worked on and the Selenium basics, API's. It was just like a discussion rather than an interview.
- 4th round was HR Round: The salary discussion and the negotiations etc.

Questions,

- 1st round: POM, oops concepts, Collection, Waits, TestNG annotations, Cucumber framework keywords, Groups, Window handles.
- 2nd round: Real time scenario questions.
- 3rd round: More in tools

Tips: Be thorough with your framework, Uses of OOPS and collection, Data handling.

MAVEN TECHNOLOGIES | Automation QA | Pune | 2 rounds | 3 years

In today's time just knowing the concept will not help you clear interviews. If you show holding experience then you should be ready for real time questions. Total 2 rounds and then final HR round was there. Medium process and not very hard or easy. Basic should be clear.

Questions: Oops, Exceptions, Collection, Framework TestNG

Tips: Real time knowledge

HCL | Senior Software Tester | Lucknow | 2 rounds | 4 years

Interview experience was excellent, always we should ready for the new challenges and always ready to learn new thing. The interview process consisted of Two rounds:

- Online F2F technical round
- Online F2F Managerial round

Questions,

- If you are ISTQB certified - you should know about seven principles of ISTQB.
- Explain your current running project with role and responsibility.
- How to conduct risk analysis.
- Type of Document which required for testing
- Some logical questions and test scenario.
- SQL

Tips: While giving interview try to explain your current project with proper explanation. if you are able to explain in proper way 50% chance to clear - next 50% chance how much you are confident while answering the question.

MAVERIC SYSTEMS | Test Lead | Chennai/Bangalore | 4 rounds | 7 years

It's a delight sharing my experience with you all. There conducted three rounds of interview, all were coordinated and challenging rounds. Technical round, Live code, Manager round. HR Talent acquisition team is so helpful. Key to Success:

- Explain about your experience in detail
- Questions will be mostly on your experience and tools which you have used - like HP ALM, UFT, SELENIUM, JIRA, BDD framework
- Use correct keywords while answering
- Research about the company and project which you are interviewed for.

Questions,

- They will ask our career profile
- Projects which we are into
- Technical questions on selenium, Cucumber, json mostly tools and frameworks you have used.
- Java string Manipulation
- Feature file creation (BDD framework) using cucumber
- Click, dropdown, selecting a text, screenshot functions in selenium
- Write a feature file for logging to Gmail
- XPath using CSS locator, object, class name
- Hashmap java program
- Reverse String java program
- Waits in selenium
- Selenium architecture
- Team Handling questions.
- Test data preparation
- RTM, client services questions

Tips: Research about the company and project which you are interviewed for, use correct keywords and logic for coding round.

JK TECH | Technical Lead | Bangalore | 3 rounds | 6.5 years

Had 2 rounds of technical round.

- 1st round - straight questions on selenium and java - with coding - for 45 min. I was able to answer most of the questions.
- 2nd round - tricky questions on selenium, java, process, API testing (with coding) for 1 hour. I was able to answer almost 70% correct and remaining I tried to tell related answers. I did not say no and gave up - I tried to answer the unknown questions.
- 3rd round was HR - communication check

Questions,

- Selenium program - button, radio button, scrolling down and up, multiple window handling
- Java programs - to swap 2 number, find factorial, find number of words in string, Fibonacci series
- Process of agile.

Tips: Be confident on what you answer - even though you know it is wrong.

INFOGAIN | Consultant Developer | Noida | 4 rounds | 5.5 years

I had gone through 4 rounds of interviews in this company, out of which 3 are technical rounds and 1 is managerial. The whole process took approx. 20 days, HR was good and she communicated well after every round.

- Java oops concept questions
- Basics programming question- reverse string, how to traverse in a table, split string.
- window handles, character count in a string.
- Array vs array list, list, map, set
- wait, glue in cucumber, runner class in cucumber, tags, TestNG, extent report.
- Define your framework (This que has been asked in every interview)
- Agile, Scenario-based ques related to project and many more....

Tips: If you are in Automation, prepare the basics of the language you are good in, like oops concepts in java. Try to give as many interviews as you can it will increase your chances of selection and make a habit of note down the question which was asked to you and do prepare it for the next interview. This way you will cover all the topics so easily. And don't give up if you not selected in first few interviews keep trying and prepare yourself you will certainly crack one.

OPTUM UHG | AUTOMATION TESTER | HYDERABAD | 2 rounds | 3 years

The interview process consisted of two rounds:

- Online F2F technical round (1 HOUR)
- Online F2F Managerial round (30 MIN)

For me, they wrapped up two rounds on the same day.

Questions,

- Tell me about yourself.
- Explain previous project frameworks and domain you worked in.
- Explain which type of selenium framework you worked in.
- Gave me a java program and asked me to provide the output.
- Asked me to write a program where I have to use add(), get(), remove(), size() methods on a dynamic array in the same program.
- How many types of TestNG annotations are available and when should we use it?
- What is the significance of actions class in selenium?
- Why hashCode is used in java.
- Asked me about the BDD framework like why the BDD framework is used, what type of language you will use in the feature file of the BDD framework, and how you will establish a connection between the feature file and step definition file.
- Asked me to write a scenario in the feature file format and the methods inside the feature file will represent the methods in step definition and types of keywords used in the feature file.
- Difference between scenario and scenario outline keyword in the feature file.
- How can we use multiple data in the feature file?
- Explain Defect lifecycle.
- Asked me how familiar I am with SQL. gave two tables and asked to explain what will be the output for different kinds of joins.
- What is the difference between union and union All?
- When will you use where clause, having clause and difference between them?
- Which comes first - Having or Where?
- Asked output for query - Select * from table.
- Asked about basic GIT commands.
- Asked about Jenkins integration with maven.
- Tell me about yourself.
- Why are you looking for a job change?
- Some scenario-based questions.

Tips: Update your profile in all the job portals regularly. Please know about your company and the profile you are applying to.

BANK OF AMERICA | QUALITY SPECIALIST | CHENNAI | 3 rounds | 3 years

The job role clearly mentioned about their requirement is on HP-UFT/QTP, Agile and Testing framework development. The interview process consisted of three rounds:

- Telephonic technical round
- Online F2F technical round
- Online F2F Managerial round

The interview experience is somewhat hectic because on the same day we have to go through 3 rounds which lasted around 40 mins - 70 mins.

Questions

- Tell me about yourself.
- Demonstration about Type of framework, Tools, Technologies, Domain that I have experience in.
- How to retrieve a file content from a particular location and print the number of times a particular string is repeated using UFT.
- Which Commands are used to print the total lines in a text file using UFT?
- Instr and Strcmp command in UFT.
- Which Commands are used for Retrieving Date and Time into the console?
- Now the interviewer started asking questions regarding the Banking domain.
- What is meant by wholesale banking and what is the mechanism used in swift messages?
- What is meant by the beneficiary and what are the details required to do the transaction and transfer the amount to a beneficiary?
- What is the mode of transactions and type of transactions available?

F2F Technical round,

- Tell me about yourself.
- Explain previous project frameworks and domain worked in.
- Asked why join, mid, instr and other inbuilt commands were used in UFT.
- Asked about defect tracking and also asked about how priority, the severity of the defects is decided
- Gave three scenarios 1) username and password are correct, but the password does not match the password criteria, after clicking the login button the page is still navigating to an unexpected page. 2) username and password both matching the default criteria, after clicking the login button but user logging into unexpected page 3) In the username field, the tester gave both correct username and password combined and left the password field blank, after clicking the login button but still the page is navigating to an unexpected page. Only one developer in the project working at the point of time define the priority, the severity of the defects, and justify your answer.

- What is the type of applications you worked on in your previous project?
- Explain how a bank transaction happens.
- In a text file, you have repetitive fields account number, transaction ID, transaction details how do you retrieve transaction ID for a particular account number.

Online F2F Managerial round: (This round happened with the Quality Manager)

- Tell me yourself.
- Why are looking for a job change.
- What are the roles and responsibilities in your previous project?
- What is the use of the QC center?
- How do you maintain your automation scripts and how often you do rework of automation scripts?
- If any script failure what are the actions you performed.
- Other than that, they asked some behavioural and scenario-based questions.

Tips: When you attend an interview, please make sure before the interview in which domain you are going to get interviewed. Prepare about the domain questions more.

ACCENTURE | Senior QA Analyst (Agile Tester) | Hyderabad | 4 rounds | 6 years

It took almost 1.5 months to complete 4 rounds of interview and release offer. Be patient and don't stick with one, keep searching/trying.

Thank you Parikshram, with your daily exclusive jobs list gave a tremendous amount of confidence about the opportunities available in the market and your tips really made to think and focus accordingly which helped in preparation. You guys are really doing a Great job. Keep up the Good work!

- Technical: Conceptual, Agile Methodology, Project, Roles & Responsibilities, Documentations, SQL basics, (Tell me about Webservices & mainframe testing n how did you used- I was not much aware of these but was mentioned in resume as I worked for couple of months - Questioned because I mentioned in resume)
- Managerial: How your day starts in the company/project? All situational with timelines-based questions, tested my approach to all situations and at times added few points also gave advice for my answers.
- HR: General intro, Sal discussion/Negotiations/Finalised CTC, explained benefits, Salary structure and all - After couple of days offer letter was sent.

Tips: Advice for Manual testers - Be proficient with the skills you have. For sure have knowledge on SQL like mandatory. Basic knowledge on automation would be good to you. - Use your LinkedIn well.

CAPGEMINI ENGINEERING | Technical Lead | Bangalore | 3 rounds | 8 years

Interviewed for API Automation Role. Good experience. Good communication by HR. They are very clear on the budget, roles and responsibilities of the vacancy. They were prompt in giving feedbacks.

- Explain Rest Assured framework in your current project
- What are the different methods used for Rest APIs?
- What is the HTTP Status codes and why do you get them?
- Explain serialization, deserialization
- How do you automate Auth token?
- What are the different types of Authorisation/Authentication?
- What is difference between scenarios and scenario outline in cucumber?
- How do you integrate Jenkins with cucumber test executions?
- What is the reporting structure for your test executions?
- What is a step definition file and why do you use it?
- Explain the architecture of REST API in your application
- How do you validate the response of your API in automation?
- How do you assert values in JSON response of your API?
- How do you provide test input for API in automation?
- What is query param and path params?
- When do you Base64 encoding?
- How do you integrate Jira with Rest Assured Cucumber?

Tips: Be thorough with concepts. Stay focused and answer to the point with sufficient explanation.

CAPGEMINI | Associate Consultant | Bangalore | 2 rounds | 3 years

Interviews process was good. 1st round technical and 2nd HR round.

- Role and Responsibility in project.
- About Framework
- Pattern program
- Reverse String program
- Selenium basic questions like Action class, about Webdriver, finding all links in page etc.

Tips: Prepare well before interview and stay focused and bold.

HCL | Senior Automation Engineer | Kolkata | 2 rounds | 5.3 years

- What is Smoke testing, sanity, regression
- When to raise a defect.
- Defect life cycle

- Oops concept
- Programs like: Reverse a string, Remove duplicate elements etc
- Collections in Java
- Basic selenium concepts: different type of waits, locators etc.

Tips: Have basic java programming skill. Prepare programs like: Reverse a string, Remove duplicate elements etc. Have selenium knowledge.

ACCE | Functional tester | Pune | 2 rounds | 3 years

First Assessment test was done. It was multiple-choice questions related to software testing. After clearing that test technical round done for 45 minutes.

- Write query for selecting second highest salary
- Write query for selecting employee names who have same salary
- Write query to select last record of the table
- Questions about RTM
- Negative scenarios for cab booking page similar to OLA
- If A, B, C modules are interconnected and you have access of only one module. How you will select testcases for regression testing
- What are the Entry criteria?
- What are the Exit criteria?
- Explain ECP with example
- Explain BVA

Tips: Prepare each detail about functionality related testing, be ready for everything you mentioned in your resume.

BAJAJ FINSERV | Manual Tester | Pune | 2 rounds | 1 years

It was quick & smooth process. One telephonic skill-based round done for about 45 minutes. Second round was for written skills. They gave me a scenario to write some testcases.

- Tell me about yourself
- Explain the projects you worked on
- One login-forget password scenario given and asked about negative testcases for it.
- Testcases for a tea cup.
- Writing testcases for 'Search' box for given link.

Tips: Basic manual testing.

PROGNEUR TECHNOLOGIES | Manual Tester | Pune | 1 rounds | 1 years

It was a quick & smooth process. Telephonic technical interview done for about 15-20 minutes.

- Tell me about your work experience
- Explain the project you worked on
- If you have a new project, what will be your approach.
- "The recent bug you found, which the developer didn't accept" scenario
- How you estimate your time to finish your testing. What will be your steps?
- Any experience in automation testing?
- Difference between smoke & sanity testing
- What will be your approach for regression testing?

Tips: Basic Manual software testing.

PRODUCTOSSEIR | Manual Tester | Pune | 1 rounds | 1 years

First HR called and asked about some basic information about my last experience, location etc. then scheduled 1 technical round of 30 minutes.

- Tell about yourself
- Tell about the project you have worked on
- Which tools you have used
- Tell me 4 Main test cases for IRCTC website
- Do you know about SAP?

Tips: Prepare basic software testing questions.

AMDOCS | Performance Test Engineer | Pune | 3 rounds | 2 years

My interview at Amdocs had 3 rounds - Technical, Managerial and HR. Technical round was 40 minutes approx. interview with basic questions in scripting (primarily Jmeter) since this requirement was for Jmeter. Managerial round was a 20 minutes interview, sort of technical only. Post which HR discussion was a telephonic round.

As heard, Amdocs usually also takes a scripting round which is of 3 hours in which you will have to do scripting for a scenario which they would share via screen share, which was not there in my case.

- Difference b/w vuser as thread and vuser as process
- Thinktime and Pacing difference
- How we calculate pacing for a scenario if we are only provided with TPS
- Any Performance bottleneck found. And how it was resolved
- How many servers are being used in your application?

- What are the backend technologies used for your application?
- How many types of thread groups have you used in jmeter?
- How many listeners are you aware of?
- Any custom plugins used?
- What is stability, scalability, stress and spike testing
- Ever came across memory leak during endurance test
- Which all GC algorithms have you used

ACCENTURE | Performance Tester | Gurgaon | 4 rounds | 2 years

The interview process was very smooth and quick. 1st there was an online assessment on LoadRunner on Mettl Secure Browser. Then 1st interview round was scheduled which was technical round for 1 hour, followed by a managerial and a HR round.

- Roles and responsibilities as a PT.
- What things do we collect during requirement gathering?
- If client doesn't provide with any TPS or SLA rt, how would we proceed?
- How do we identify a business-critical scenario?
- How do we set rules in automatic correlation?
- If we have dynamic boundaries of a dynamic value, how do we correlate it
- How many types of tests have you performed?
- Ever came across memory leak, how you resolved it?
- Ever monitoring garbage collector graph in AppDynamics?
- What are the different schedulers in PC?
- What is global scheduler

Tips: Focus on scripting and analysis part.

CAPGEMINI | Performance Tester | Pune | 2 rounds | 2 years

Interview consisted of 2 rounds - Technical and HR. Technical round was conducted on Elevator platform and was of approximately 40 minutes. Major questions were scenario based from scripting and few others from analysis. HR round was a telephonic round which majorly had salary discussion.

- How can we get the response times of each single request in a script: Using web diagnostics graph in LRA?
- Functions in LoadRunner
- If we have a scenario to upload and download any document. How will we proceed with the scripting for it?
- Correlation types

Tips: Focus on scripting part.

LTI | Specialist-QE Engineering | Pune | 2 rounds | 7 years

Had 2 rounds, 1st round was screening and little about skills, 2nd was technical round. It was based on manual as well as automation testing. They asked me more about situation-based questions.

- What is Regression testing.
- What is the difference between smoke and sanity testing?
- What is the difference between Equivalence class partition and Boundary Value Analysis, and what is its use?
- Java coding
- Explain automation framework which used in project
- Most of the question were situation based.
- They gave me scenarios and asked the logic for automation testing

Tips: 1st step is always shortlisting of resume; I have 3.5 years of gap. I used subject line which Parikshram guided me, and it was very beneficial, after gap also, I got many calls, as per my experience credit goes to subject line e. g. QA-Testing | 7.10 Years | Pune | Available immediately

COGNIZANT | PROGRAMMER ANALYST | HYDERABAD | 2 rounds | 3 years

The interviewer was very casual and told me about the requirement which they are looking for. The technical interview went on for 25-30 mins and at last, he signed off by saying he will speak with HR and get back to me. Got a call back from HR once I submitted all the requested documents regarding salary negotiation and other basic HR questions.

- Tools, domain, and Technologies which I worked in my experience.
- Frameworks which I worked on my previous project.
- Difference between HP-UFT and selenium.
- Defect life cycle process - asked for a detailed explanation (this question itself I explained for 5-7 mins)
- Selenium - what are actions, how will you handle select dropdown, which is faster CSS or Xpath.
- Why TestNG is used in the testing framework?
- Explain difference b/w regression, smoke, sanity testing
- Basic Java Questions.
- Why are you looking for a job change?

Tips: Please check your emails every now and then. If possible, create a separate mail ID for applying for jobs alone. If you have 90-day notice period please don't hesitate to apply and inform the recruiter while the initial screening round that if you get selected you can negotiate my notice with the current employer. Attend as many interviews as possible. Don't hesitate to attend interviews if you don't know about all the technologies mentioned in the Job description. Never ever think about whether you will get selected or not based upon your interview experience. It totally depends upon the recruiter's mindset and the role for which you applied.

ORACLE | QA Lead | Hyderabad | 5 rounds | 9 years

The interview was really smooth with clarity on each round. During discussion with HR, he has given me clarity on each round. I had 5 rounds - Core Java round, Automation Testing round 1, Automation Testing round 2, Managerial Round 1, Managerial Round 2.

Java,

- We have list of integers 1-99. We need to find a missing number. How to find and also should calculate the time complexity
- There is a string. Reverse the string in such a way that words should be reversed not complete string
- Difference between array and arraylist
- Difference between HashMap and Hashtable
- What is reflection in java
- Oops concepts overloading, overriding
- static keyword in java
- Programs on Map, arraylist

Selenium,

- How to handle Firefox download popup using selenium
- How to select a checkbox, radio button
- How to get options from a dropdown list
- How to navigate to iFrames and comeback to parent page
- How to test if a window is opened or not after clicking on a link
- Parallel execution
- Order of execution
- What are different ways to click on a button
- Actions class, JavascriptExecutor
- Project framework
- Some questions on runner class
- Given some sample programs with string methods, we need to tell the output
- Some statements were given on interfaces
- How to read a notepad file

- How to identify a span element
- Some scenario-based questions
- Reverse the array in the subset of N -> input: [1,3,5,7,9,11,15,17,19] order N =3 > output: [5,3,1,11,9,7,19,17,15].

Tips: Before attending any interview make sure that our basic concepts (Java, Selenium) are clear and thoroughly prepare with all the technologies and tools mentioned in resume.

DHARMA LIFE | SOFTWARE TESTER | SOUTH DELHI | 2 rounds | 2 years

Questions,

- Types of Testing
- Regression Testing with example
- Unit testing
- Drawback of selenium
- Limitations of selenium
- Tools used for automation testing
- How to do performance testing
- Difference between load and stress testing
- What is white box, black box and grey box testing
- Exceptions generated in selenium apart from webdriver exception

Tips: BE YOURSELF AND ENTHUSIASTIC IN THE INTERVIEW.

JOSH TECHNOLOGY GROUP | QA | Gurugram | 4 rounds | 2 years

- First-round: HR called up and asked about basic information
- Second round: QA Assessment Round
- Logical Objective questions
- Telephonic/Technical Round with QA manager: Grilled for about 1 hour
- Assessment Round: Given an assignment to make test cases and find out bugs on Zoom call, a girl (from QA team) was there. Around 1 hour 45 minutes I made test cases, for the application and found around 8 bugs, then I asked them to I need break or we you can judge me on the cases I made.

MAVERIC SYSTEMS | Senior Software Engineer | Chennai | 4 rounds | 3 years

It's a wonderful experience. Talent acquisition team is so helpful. We will be getting regular updates.

2 rounds from Maveric systems, 2 Rounds from client side.

- Oops concept
- Program to display first non-repetitive character in a string

- Program using TreeSet for sorting weekdays
- Window handling
- Frames
- Parent, ancestor Xpath live examples
- Live coding
- Scenario: Go to google and search Stop watch online, click on start, use wait and then stop
- print the value in the stop watch
- Jenkins
- Agile
- Selenium Grid

Tips: Be strong in Java collections and live practice is selenium Xpath.

CAPGEMINI | Associate Consultant | Bangalore | 3 rounds | 2.5 years

- 1st round - Basic Selenium, Agile process
- 2nd round - Key word driven automation
- 3rd round (client) - Jenkins, Test management tool integration and selenium grid
- How do you integrate automated test cases to test management tool?
- How will you run a project in Jenkins?
- How to set up hub and node

Tips: Be strong in frames and window handling concepts.

CAPGEMINI | Senior Consultant | Pune | 1 rounds | 13 years

Interview was for the Scrum Master position. It was easy, but not very easy. If you know your concepts, then you can clear it. Good communication is also important.

- Roles of Scrum Master
- What is Sprint Review?
- Time box of various scrum ceremonies
- Examples of Conflict resolution
- How to avoid scope creep?
- Can a person be both a Product Owner as well as Scrum Master?

Tips: Have good communication skills, understand the concepts.

APPSIAN SECURITY | Test Automation Engineer | Bangalore | 3 rounds | 3.7 years

It was nicely done. My HR coordinator was very friendly and frank and all 3 interviewers were very nice. Looking forward to join them.

- In Technical round, they gave me 2 tasks to automate.
- They gave me to automate Dynamic web table and to match entries with user entries.
- Next task to automate, pagination, if we are at page 1 so previous button should be disable and if we at last page so next button should be disable.
- Also, prepare your Automation framework.

Tips: Be prepared with all fundamentals. Round 1 was technical, where they gave me 2 tasks to automate. Round 2 was with Manager, he asked few questions on my roles and responsibilities and why I wanted to move out from my current organization. Round 3 was with HR, she asked me why are you moving out, are you open for Bangalore and we had Salary negotiation. Next thing, I received offer from Appsian Security.

NTT DATA SERVICE | Testing engineering specialist | Hyderabad | 3 rounds | 6 years

It was tough actually when I started giving was getting rejected most in the second round. What I noticed that round was very detailed one to the focus on particular concepts of framework, logic, and programming skills.

- How to use substring() method in java?
- How to find duplicates in Array using SORTING?
- Handling File Uploads Using AutoIT
- Advanced User Interactions
- TestNG
- Passing Parameters to Tests
- Configuring Test Suites
- Framework
- Page Object Design Pattern
- Data Driven Tests using TestNG and POI
- Introduction to Maven and ANT

Tips: Apart from all above Must learn Cucumber, Jenkins, Git Hub/ Bit Bucket/ Agile Process/ Jira (for project management and Defect Reporting) should have intermediate knowledge.

INDIUM SOFTWARE | Test Associate | Bangalore | 2 rounds | 0 years

First round was taken HR in order to filter candidates and second round was taken by Team Leads.

- Diff b/n smoke & sanity testing
- Explain low priority with high severity bug & high priority with low priority
- Explain compatibility, regression and exploratory testing
- Who will decide severity of bugs and who will set priority to a bug?

Tips: Be confident.

FREQUENCY SOFTWARE LLP | Associate QA | Pune | 2 rounds | 3 years

It was good experience. Interviewer was helpful & shared their knowledge, experience also.

- Associate QA Manual
- First Round: Intro, Project flow, Agile, SQL Queries, STLC, RTM.
- Second round: Google Scenarios, Story Points, SQL Queries.

Tips: Prepare SQL Well.

FREYR | Test Associate | Hyderabad | 2 rounds | 4.5 years

Over all it is good experience.

- Tell me about your self
- Projects and tools used
- Different types of testing smoke, sanity, functional, regression.
- Bug life-cycle
- Test scenarios, Test cases difference.
- Some basic SQL.
- What is Test Plan
- Second round: About project and difference between severity and priority and some practical questions like within limited time how do you execute large count of test cases how do you prioritise?

All the best each one of you looking for job.

Tips: If you are preparing for interview prepare daily maintain consistency try to learn new things.

SLK GROUP | Senior engineer testing | Bangalore | 3 rounds | 4 years

- Explain about yourself, your role.
- Roles and responsibility
- Explain types of testing you did
- Did you handle team
- Client interaction
- Which methods did you follow
- Tool you used
- Procedure you followed
- Daily routine
- Release duration
- Why looking for change
- Did you handle team

- Explain types of testing you did
- Priority Severity related questions
- Tough to test in your previous org
- Test plan explain
- Agile explain, types of agile
- Models diff waterfall and agile
- Agile meetings
- Test case design tech
- How you made coverage of scenario or TC is good
- Dev is not able to reproduce bug why?
- Inconsistent bug
- Environment tested
- Exploratory testing
- Management questions if you were lead what would you do if there are any bugs
- When to stop testing
- What test strategy did you follow
- Which all domain knowledge you have
- Realistic case study
- Calculation scenarios
- Write test case scenario, find bug and send us
- Test case and scenario template
- Difference smoke and sanity testing, verification and validation, retest and regression
- While raising bug what all parameters required?
- Charles proxy
- Types of mobile testing
- Difference between web and mobile testing
- Latest version, devices - mobile
- Types of mobile applications
- Types of mobile app testing
- Criteria while testing mobile application diff from web application
- SQL languages, sub queries, joins, grant revoke diff, write query for today's date, all types of queries
- What is API, why we need to do API testing
- API methods
- Error codes
- What are main things in API testing using postman
- Document where all API info will be stored
- Java oops
- Java Collection
- All basic concepts
- Write program for some basic program, reverse string, etc.
- Explain your framework, explain types of frameworks

- POM model
- TestNG
- Apache POI
- Implicit and explicit wait difference
- Locators
- What all will you automate

Tips: Be confident, explain your profile properly, tell straight forward what you know what you don't know, your future plans.

HEXAWARE | Senior Tester | Pune | 4 rounds | 4.9 years

It was a Good Experience.

- Testing Question
- API Status Code
- Put or POST Method
- Webservice vs API
- Monolithic vs Microservice
- Write Test Cases based on some scenario
- Capital Market Related
- Investment Banking
- SQL Queries (Joins, Sub-query, Update, DML, DDL)
- Test Scenario Real-Time
- How will you manage team?
- Team Related and Some Behavioral

Tips: Keep Basics Clear and Practice as much as You can.

RAZORTHINK | QA Lead | Bangalore | 6 rounds | 8 years

Round 1 and 2 were taken by junior testers. Very basic questions on Selenium, Java concepts.

Round 3 was managerial. Very realistic scenario-based questions. Round 4 and 5 were by senior management which went deeper into team management, team lead kind of questions. Round 6 was a project-based coding. Where I was asked to create a GIT repo, create a framework, automate a few scenarios on redbus.in site. Share my repo with them.

Tips: Be thorough with concepts. Before the interview process, discuss with the HR to know their budget for your role.

LNT INFOTECH | Sr. Automation QA | Bangalore | 2 rounds | 9 years

- First round was telephonic by some third-party vendor. He was quickly asking the questions to filter the candidature.
- Second round was via a zoom video call, taken by an SDET I guess.
- Overall - it was more technical interview, may be because of years of experience and role.

Questions,

- Questions ranged from Core Java - Testing - Agile - SQL - etc.
- Current role & responsibilities.
- Java program to find duplicate chars.
- Theoretical Qs around java - static, OOPs concepts, custom exceptions, lambda expressions, design patterns, etc.
- Selenium real-time Qs - dynamic element, exception handling, framework structure, reporting, listeners, run failed cases, read test data, config file, Jenkins, CI-CD.
- Testing - mostly situational like crunch timelines, team management, agile team/events/estimation, one or two SQL queries, test plan & approach, etc.
- API Testing basics like status codes, read JSON, authentication, etc.

Tips: Now-a-days the focus during initial rounds is on Java - programming - test automation. I feel keep giving interviews no matter what. I have learnt a lot during these multiple interviews itself.

CISCO | Software Engineer Test IV | Bangalore | 3 rounds | 2 years

I appeared for interview at Cisco a week back for testing engineer for DNAC team. The whole focus of interview was on automation testing via API's & also UI automation using selenium & other tools. Very technical rounds needing knowledge of Java, Selenium. There were also questions around issue tracking software like JIRA & Rally. Overall, it was a good learning experience for me.

- Introduction: My background & what do I do in my current role & related questions.
- What tools have you used for automation testing of RESTful API's. What all things do you verify in API Testing?
- What %age of total tests is at API level and at UI level?
- Sample code to send a post request using Rest Assured. What authentication mechanism do we use in our project.
- Selenium: What's your locator strategy? How do you handle dynamic elements?
- Say you got search results in a web table. How to read it using Selenium?
- Java programming Qs using Hashmap and Arrays.
- How many tests do you automate in a week? Some challenges that you faced while scripting/execution.

- Qs on basic knowledge about CI-CD, Jenkins, Git.
- Do you understand networking? I said yes, a little. so, it followed with
- What are subnets, VLANS.
- Difference between a router & a switch.

Tips: Knowing Java, Selenium & other tools useful for API automation is a must. Knowing Postman scripting & automation I thought would have been helpful.

& With this, you are ALL SET to face any Testing interview like a pro!

Hope the book was of some help in your preparation and answered few of every job seeker's queries.

Would love to know your feedback, or a list of Qs that you want to be included – just drop me an email at deepanshu@parikshram.com!

Join me in building the first dedicated platform for QA-Testers – <https://parikshram.com> – by spreading the word in your friend circle.

Enabling Success, together
Deepanshu Agarwal