**Nowacka, Hanna** [**hanna.nowacka@nordea.com**](mailto:hanna.nowacka@nordea.com)

Dominika Wojto

**Interview questions**

1. How is good test case prepared?

Good test cases are prepared by understanding requirements, identifying test scenarios, prioritizing cases, defining objectives, writing clear steps with pre/post conditions, specifying input data and expected results, keeping cases atomic, ensuring reusability, reviewing, and executing systematically.

1. What would you tell about defect...? What does a good defect include.? How you should write defect to make it clear for developers

A defect is an abnormal behavior in software. A good defect includes clear description, steps to reproduce, environment details, severity/priority, screenshots/attachments, expected/actual results, and additional context to make it clear for developers.

1. What tool you use to support testing process

Testing processes are supported by various tools such as test management tools (e.g., Jira, TestRail), automation tools (e.g., Selenium, Appium), performance testing tools (e.g., JMeter, LoadRunner), and defect tracking tools (e.g., Bugzilla, GitHub Issues).

1. Which test level you used in testing (Unit testing. System testing, integration, regression testing, UAT..likewise)

Test levels used include unit testing, system testing, integration testing, regression testing, and user acceptance testing (UAT).

1. What type of test techniques you know or have used so far?

Test techniques include equivalence partitioning, boundary value analysis, decision table testing, state transition testing, pairwise testing, and exploratory testing.

1. What requirements you now other than functional requirements. Non-Functional. What you know of NF requirements

Non-functional requirements include performance, usability, reliability, scalability, security, and compatibility requirements.

1. Let’s imagine a situation you are only tester in the team. you test web application with 5 browsers, 4 languages having a total 2000 testcases altogether. you have time to do only 100 tests. What will you do in the best? when you cannot perform all test. What prioritization will you do and What will be the basis.

In a situation where I have limited time to execute only 100 test cases out of a total of 2000 for a web application that supports multiple browsers and languages, I would prioritize the test cases based on the following factors:

1. **Risk-based Prioritization**: Identify high-risk areas of the application where defects are more likely to occur or where failures would have the most significant impact on users or business objectives. Focus on testing functionalities critical to the core functionality of the application.
2. **Browsers and Languages**: Consider the most commonly used browsers and languages by the target audience. Prioritize testing on browsers and languages that have the highest user base or market share. Additionally, prioritize testing on browsers and languages known to have compatibility issues or user experience concerns.
3. **Functional Coverage**: Ensure that the selected test cases cover a wide range of functionalities across the application. Prioritize test cases that validate core features, critical user workflows, and frequently used functionalities.
   1. **Regression Testing**: Prioritize regression test cases to verify recent changes.
   2. **Localization and Browser Compatibility**: Test languages and browsers with the widest usage and known issues.
   3. **User Experience**: Emphasize test cases that evaluate overall user satisfaction.

By prioritizing based on these factors, I'd select a subset of test cases that offer the most coverage and risk mitigation within the limited testing time.

1. When you have a bug but developer claim that this bug does not appear in his local environment, but bug is critical to fix. What will you do resolve this?

Investigate further by analyzing differences in environments, providing detailed steps to reproduce, collaborating with the developer to resolve the issue, and considering environmental factors such as configurations, dependencies, or data.

1. What roles you cooperated in your project in your experience.

Cooperated in roles such as requirements analysis, test planning, test execution, defect tracking, test automation, and collaboration with developers, product owners, and stakeholders.

1. What personal skills you should have while dealing with teams
2. What position you would like to be in 1 / 2 years. how would you imagine

In 1/2 years, envision taking on a leadership role such as a test lead or test manager, leading testing efforts, mentoring team members, and contributing to organizational growth.

1. What you know about BDD

Behavior-Driven Development (BDD) is a software development approach that emphasizes collaboration among stakeholders to define behavior through examples in natural language, focusing on business value and acceptance criteria.

1. What is content of a good defect for developer too understand?

A good defect for a developer includes clear description, reproducible steps, environment details, severity/priority, screenshots/attachments, expected/actual results, and additional context for understanding and resolving the issue.

1. What is most important for you while working in teams?

Collaboration, communication, teamwork, trust, respect, empathy, and shared goals are most important while working in teams.

1. What are the test levels according to ISTQB. What you used/performed

Test levels according to ISTQB include component testing, integration testing, system testing, and acceptance testing. Have experience in performing all these test levels.

1. how you prepared for acceptance test

Preparation for acceptance testing involves understanding user requirements, defining acceptance criteria, creating test cases, coordinating with stakeholders, executing tests, and obtaining user approval.

1. Black box testing or White box testing. What you deal more and why

Both black box and white box testing are important. More focused on black box testing as it verifies functionality without knowing internal code structure, ensuring user perspective and requirements are met.

1. API testing – Services you know, platforms worked upon, please elaborate your experience

Experienced in API testing using tools like Postman, platforms like RESTful APIs, SOAP APIs, and GraphQL, testing endpoints, payloads, authentication, and response validation.

1. Test Data preparation – How you prepare Test Matrix

Test data preparation involves identifying data requirements, creating test scenarios, generating test data, defining input/output values, and organizing data in a test matrix for systematic testing.

1. What is Equivalence partitioning testing

Equivalence partitioning testing is a test design technique where input data is divided into partitions, treating each partition as equivalent and selecting representative test cases from each partition to achieve test coverage efficiently.

1. What the different types of test coverage techniques are?

<https://www.careerride.com/view/basic-concepts-of-software-testing-mcqs-18823.aspx>

**Manual Testing Questionnaire**

1. Why is Software Testing required?

Quality of the product, meeting the requirements, verify as per the requirements, quality and performance

Software testing is imperative, as it identifies any issues and defects with the written code so they can be fixed before the software product is delivered. Improves product quality. When it comes to customer appeal, delivering a quality product is an important metric to consider

1. When do you start the testing process?

Right from requirement gathering till the software is deployed, Better to involve Test team from requirement gathering. Will know the exact functionality. Test team will come up with Test cases

Software Testing should start very early in the project life-cycle, as soon as there's a Functional Requirements Document (FRD). The STLC consists of a series of phases carried out methodically to help certify the Application Under Test.

1. What is the difference between QA and QC?

QA ->Starts from Requirement gathering. Test Lead will give sign off, QC - > Quality check - > Test Manager gives sign off

| **Quality Assurance** | **Quality Control** |
| --- | --- |
| It is a process which deliberates on providing assurance that quality request will be achieved. | QC is a process which deliberates on fulfilling the quality request. |
| A QA aim is to prevent the defect. | A QC aim is to identify and improve the defects. |
| QA is the technique of managing quality. | QC is a method to verify quality. |
| QA does not involve executing the program. | QC always involves executing the program. |
| All team members are responsible for QA. | Testing team is responsible for QC. |
| QA Example: Verification | QC Example: Validation. |
| QA means Planning for doing a process. | QC Means Action for executing the planned process. |
| Statistical Technique used on QA is known as Statistical Process Control (SPC.) | Statistical Technique used on QC is known as Statistical Quality Control (SPC.) |
| QA makes sure you are doing the right things. | QC makes sure the results of what you've done are what you expected. |
| QA Defines standards and methodologies to followed in order to meet the customer requirements. | QC ensures that the standards are followed while working on the product. |
| QA is the process to create the deliverables. | QC is the process to verify that deliverables. |
| QA is responsible for full software development life cycle. | QC is responsible for [software testing life cycle.](https://www.softwaretestinghelp.com/what-is-software-testing-life-cycle-stlc/) |

1. What is static testing?

Will check documents, requirement documents and find errors. It involves review. A functional test and happy path testing.

Static testing is a software testing method that examines a program -- along with any associated documents -- but does not require the program to be executed

1. What is dynamic testing?

Running test cases.end to end testing

Dynamic testing refers to analyzing code's dynamic behavior in the software. In this type of testing, you have to give input and get output as per the expectation through executing a test case. You can run the test cases manually or through an automation process, and the software code must be compiled and run for this.

1. Test deliverables?

Test case design, Test plan, Test cases, Test summary reports, Release summary, Test results, sign-off on Test cases. Linking the test defect to a test case.

**Some of the important test deliverables are enlisted below for your reference:**

* Test strategy
* Test plan and estimation
* Test scenario
* Test cases and test data
* RTM
* Test summary report
* Test closure report
* Incident report

1. What is positive test approach and negative test approach?

Positive : whether application behaves appropriately as per the expectation. Negative: provide negative test data, invalid data

Positive testing determines that your application works as expected. If an error is encountered during positive testing, the test fails. Negative testing ensures that your application can gracefully handle invalid input or unexpected user behavior.

1. What is a Test plan and what it includes?

Test plan includes scope, risk related details, stake holders till sign off activities.

Planning for Timelines on Deployments, Regression is required or not.

Sharepoint, Test management folders

A Test Plan is a detailed document that catalogs the test strategies, objectives, schedule, estimations, deadlines, and resources required to complete that project. Think of it as a blueprint for running the tests needed to ensure the software is working correctly – controlled by test managers.

1. What is test strategy?

It is set of guidelines to tell how testing is done and the approach. Carried out by Test lead.

A test strategy is an outline that describes the testing approach of the software development cycle. The purpose of a test strategy is to provide a rational deduction from organizational, high-level objectives to actual test activities to meet those objectives from a quality assurance perspective.

1. Difference between test strategy and Plan.

Test Strategy cannot be changed during the project execution whereas a Test Plan can be changed. Smaller projects may not have a Test strategy.

Test strategy can be used for multiple projects

Test plan is specific to a single project

| **Test Plan** | **Test Strategy** |
| --- | --- |
| Test plan can be defined as a document for a software project which defines the approach, scope, and intensity on the effort of software testing. | The test strategy is a set of instructions or protocols which explain the test design and determine how the test should be performed. |
| A test plan can be changed. | While test strategy can’t be changed. |
| Test plan happens independently. | While test strategy is often found as a part of a test plan. |
| Test plan describes about the details. | While test strategy describes about the general methodologies. |
| A test plan is done by the test administrator or test manager. | While It is done by the project manager. |
| A test plan is utilized at the project level. | While test strategy is utilized at the association level. |
| A test plan has the essential objective of how to test, when to test and who will confirm it. | While test strategy has the essential objectives of what approach to pursue and which module to check. |
| Based upon the testing strategies. | Based upon pre-defined standards. |
| Different types of test plans are level-specific , type-specific , and master test plans. | Different types of test strategies are model-based , analytical, methodical , reactive , standard-compliant , consultative , and regression averse strategy. |
| Only a single project is affected at a time. | Multiple projects can be affected at a time. |
| It describes the general and common specifications in testing of a specific object. | It describes the approaches in testing. |

1. What is the difference between master test plan and a Test plan?

Master test plan is something which is required for bigger projects. Contains scenarios etc. Test plan is available for smaller projects. Master Test Artifacts

1. What is a test scenario?

Before writing test cases, Test scenario needs to be identified and then go for test cases.(Ex: Various options with positive and negative). Test scenario is at a high level. Test cases will be at a lower level.

A Test Scenario is a statement describing the functionality of the application to be tested. It is used for end-to-end testing of a feature and is generally derived from the use cases. Test scenarios can serve as the basis for lower-level test case creation. A single test scenario can cover one or more test cases.

1. Test design techniques.

Test driven technics, error based testing, dynamic testing, testing with boundary level testing. (Ex: different scenarios for login)

1. What do you mean test coverage?

It’s a metric which is used to measure how much testing is done.

Test coverage is a metric that specifies the quantity and quality of testing performed. Simply said — it's what is tested and how much code, platforms, and different user scenarios are checked. So, it's a significant indicator in software testing in terms of effectiveness and quality

1. What is a top down approach and bottom up approach?

Top Down : Homepage(Main module), Use Stubs

Stakeholder meetings, Scratch meetings, Requirement walkthrough, Development, Testing. Unit Testing, Component Testing

Bottom Up : Part of Integration testing. (Child Modules and go Up), Use Drivers

1. What do you mean smoke test, Exploratory testing?

Smoke : Perform after Software is built. A high level testing of all the critical functionalities. Making sure the main functionality is working…(End to End checking). Will perform this testing to make sure Build is working fine or not. When it is a new code deployment, one needs to check the basic functionality

Exploratory : Test execution is done in parallel. Run time scenarios are tested. Tester needs to think real time and perform testing. Without seeing a requirement, a senior tester can test it with his experience. Does testing to check whether he can break the application

Sanity Testing : In a new build, Firstly they will recreate the scenario and do testing. Any new functionality is added to the existing, testing to be done on the new functionality in detail

Regression : Only when we fix a bug and test the functionality thoroughly

1. How do you handle a non-reproducible bug?

Configurations to be checked, Take screenshots and maintain the recording, Inform to the Test manager to take it to the business users, This defect to be discussed in Defect track meeting, Bring this up in Daily stand-up call

If you can't reproduce a bug, you first document the steps and repeat them under different environment to find it again. And if the bug is unable to reappear, then you should ask for more information from customer/client regarding the bug reproduciblity.

1. What is Requirement Traceability Matrix?

A new requirement : Design the scenarios, No. of testcases per scenarios, then Test case to be drafted. This is handled by Test Lead and all the test cases to be properly mapped to test cases.We have covered all the test scenarios or not as per the BRD’s. It will help to make sure all the scenarios are covered. It can be considered as a Check List in simple terms

A requirements traceability matrix is a document that demonstrates the relationship between requirements and other artifacts. It's used to prove that requirements have been fulfilled. And it typically documents requirements, tests, test results, and issues.

1. What are the different types of functional testing?

Unit testing, System testing, Smoke Testing, Regression Testing, Acceptance testing, Integration Testing.

1. What do you mean non-functional testing?

It is defined as testing to check the reliability, performance, no of users can use this application. How system behaves if lots of users login at the same time, how secure the system is, volume testing, whether it can be installed easily.

1. What is defect life cycle?

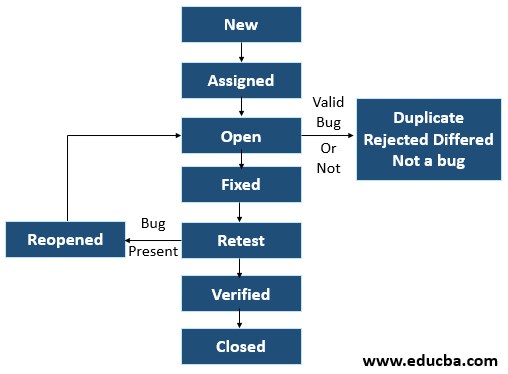
Status: New, Assign to Developer, Opened.

After fixing, assigns to tester to test,

Developer may Defer, Duplicate if the defect is repeated,

Developer can rejected

Tester will test the defect and close it with Closed status.



1. What do you mean defect leakage?

Whenever a defect is found in UAT by an end user/business user, it is considered as defect leakage.

1. Difference between priority and severity in software testing?

Priority is the importance which is given to a bug to fix

Severity is the impact on the functionality

1. What do you mean defect density in software testing?

No of Defects detected. Percentage completed. Calculated based on 1000 lines of code divided by No. of defects

1. Defect Detection Percentage mean in software testing?

It’s a metrix… No. of Defects found by Test team/Total No of defects(Test team+UAT teams+integration test total defects). Efficiency of the progress is gauged basing on this Defect Detection percentage

1. An example of high priority and low severity bug?

Say, a typo has high priority and low severity (Name of the bank or color of the company logo)

1. What is go and no go in Testing?

High severity or blockers, it is a No-Go. If it is a No-Go, it will be reverted to the previous stable version

1. What is regression testing? What is retesting? How they differ from each other?

Regression Testing: Thoroughly testing that bug and make other functionalities are not impacted. Regression pack is a standard set of test cases which is part of Automation. Standard functionality will not change. Additional functionality will be there, but no change in the existing functionality

Retesting : Developer says it is fixed. Doing testing once again

1. What is Boundary value analysis (BVA)?

It’s a kind of testing. (Text field : checking the validity… like min and max values, any invalid input

Boundary-value analysis is a software testing technique in which tests are designed to include representatives of boundary values in a range. The idea comes from the boundary. Given that we have a set of test vectors to test the system, a topology can be defined on that set.

1. What are the key challenges of software testing that you faced in your career?

Talk about challenges and also tell the solutions to resolve those challenges.

1. What is difference between Front End Testing and Back End testing?

Front end :Done on UI Part,

Back End : Performed on Backend, more of security and API testing. Fetch data from SQL Server or Mainframe

1. When we choose Informal Testing?

Informal reviews: Close the defect with comments. If justifiable comment is not considered, it can be considered as Informal testing, Small changes can be tested without capturing screenshots as it is not effecting any critical functionality

1. What is Equivalence partitioning testing?

Testing with boundary values, basically testing with specified values is called Equivalence partitioning testing. Break it into different values and testing. (Invalid Input(1-100), Valid Put(101-200), Invalid Input which is more than valid Input(Above 200))

1. What is black box testing? What are the different black box testing techniques?

Where tester is not related to code level testing. Tests only functionality. Widebox testing is done by developer. Equivalence and BVA Testing

BVA, Decision making

1. Mention what the different types of test coverage techniques are?

BVA

Equivalence Testing

Functionality testing

Test coverage tells the no. Of test cases which were covered for a particular requirement

Techniquies : Requirement coverage (Mapping the Test cases), Product Coverage, Risk Coverage,

Requirement Tracebility : Requirements drafted for different Modules, A checklist will be maintained in a Matrix format

Decision coverage Testing : Conditional Testing : Best example to give is Login Module

1. What is random/monkey testing? When is it used?

By giving random data so that we can break the system.

Load testing, Stress testing which comes under non-functionality, It is Timeboxed and Random testing, provide random values and check the response time,

1. When should testing be stopped?

If the entire functionality is blocked due to some issue, it can be deferred

All the Test cases are tested successfully and get a signoff from either Test Lead or Business lead

Whenever the Test project is over, whenever the Test deadlines are completed

We should make sure there are no open defects

When the code is defect free

QA Interview questions: ( Shankar ) – 11th May 2023

1. Can you explain the Test parameters ? what are the test parameters
2. Can you explain +Ve & -Ve test case ? why its required ?
3. Do know what kind of test levels?
4. Can you walk us through the process of debugging a failing automated test?
5. Have you integrated automated testing into a CI/CD pipeline before? If so, can you explain the process you used?
6. Can you share any tips or best practices for creating efficient and effective automated tests in C#?
7. What are the test levels/test pyramid and what are the main differences between levels?
8. What kind of test, except for functional ones, exist and do you have own experience with any of them
9. What is the difference between Severity and Priority?