Click Here

How to create test cases

There’s no one-size-fits-all for writing a good test case. There’s a lot to consider — not only between projects, but among each test within the same project. Because while they may seem like glorified to-do lists, great test cases are far more than that.Here we’ll address common struggles related to test cases and why they happen. We’ll also provide a step-by-step guide to writing a great test case, including an example. Back to top Common Problems with Test CasesSimply knowing how to write a test case only gets you so far. Poorly-written test cases can waste time and, depending on the product, potentially put users in danger.Writing great test cases means that not only are all the right tests performed, but also that you maximize your ROI. To do that, we need to look at what’s going wrong, and why.If you struggle with any of the following, you’re likely missing something:Accuracy of test cases — they may include ambiguous language or wrong information.Producing reproducible results — will two testers get the same result on the same test?The time it takes to write tests.The time it takes to train your testers — to be productive and know what to do next on their own.Knowing when you’re done writing tests.Keeping tests up to date — will they stay accurate as requirements change?The Core of the ProblemIn all of these instances, you’re probably missing clear answers to one or more of these three fundamental questions:Where is your source of truth for requirements/user stories? Depending on your product and team, requirements can look very different. Some will have formal product requirements that have been documented, reviewed, and approved. Some teams use a rather informal process to document requirements. Which sometimes means they aren’t documented somewhere that’s easily accessible to testers. Or that testers may not know if they’re working with the most recent version of requirements. The bottom line is that requirements tell you what to test. If you know what you’re testing, you know when you’re done. (By the way, using an electronic repository will solve this. Here’s one you can try for free.) Who will be running the tests? You need to write your test cases to your audience. Testers will vary in things like domain expertise, experience, and language fluency. A tester with a good understanding of the industry in general, and what the product is supposed to do, won’t need certain things clarified within the test case. Experienced testers also won’t need certain steps clarified. For example, the step could simply say “create a test case,” instead of listing out the steps to create one. It’s important to know exactly how much information you have to include. And if you’re working with contract testers who don’t share your native language, you need very consistent vocabulary and clear, unambiguous steps. What are the risks of not being perfect? Answering this tells you whether you need to cover every scenario. For example, test criteria for a GI endoscopy system you write for the Mayo Clinic would be more demanding than that for an online coloring book. The ultimate question is: if the test isn’t perfect, could someone die or be otherwise harmed? Knowing the risks helps you determine the amount of time you spend on:AccuracyTest coverageReproducibilityBe sure to answer all of these questions before you begin writing your test cases. Back to top How to Write Test CasesThis checklist outlines the steps you need to take to write a test case, and the guidelines that help ensure it’s written well.Step 1. Find your requirements (or user stories.) Again, these tell you what to test. It’s critical to complete this step first. Begin each test case from a requirement.Step 2. Create a summary for each test case. This is a clear, simple description of what the test is. If you use a test case management solution, this will appear like a title in your overview of your tests. Make it easy to distinguish one from another.Step 3. Include a description of goals. Help the tester understand the big picture.Step 4. Determine the starting conditions and any setup or equipment your test relies on. Picture a recipe for baking a cake. “Pre-heat the oven to 350 degrees” is a pre-condition. Two greased 9” round cake pans are required equipment. Do this for yourself by identifying necessary setup:Do you need special equipment for the test?Does the test need a clean install?Does it need specific database contents?Do you handle sensitive data (like credit card processing)?What assumptions do you have about other tests that might have been run?Does the test require specific test data or parameters?Use this information to be clear and thorough in your steps.Step 5. Write clear, simple steps. If you can, attach an image of a screenshot or mockup to help clarify. Here are the criteria that will make the difference between a good test case and a poor one:Write with clarity, concision, and precise language.Ensure it meets the criteria of tester independence. Would any two testers get the same results? If you say enter a “large amount” into a field, one tester could enter 100 and another 1000. Can it be that ambiguous, or do you need to be clearer?Determine what assumptions have been made in the step. What steps have already been taken? What’s in the database? Is this the first time a user has logged in? Write enough steps for the kind of tester who will be running the test. Step 6. With those same criteria, also include expected results — this is how the tester knows whether the test passed or failed. To continue the recipe analogy, for the step “Remove the cake from the oven after 45 minutes”, an expected result would be, “The top of the cake is brown, and the sides have pulled away from the pan.”This outlines the basics of writing a test case well. You can include other parameters, as will be shown in the following example.BLOG: GET MORE TIPS FOR WRITING TEST CASES IN SOFTWARE TESTING >>Back to top Test Case Example for Manual TestingHere is a screenshot of a test case created in Perforce TCM. You can see all the elements of the steps outlined above.Additionally, it provides other information you may choose to include.Leverage your test case tool’s ability to support custom fields, requirement traceability, and configurable workflow to help you organize and manage your testing efforts.You can use this additional test meta-data to help:Determine the best person to run the test – functional area, type of test.Plan to test around limited shared resources – hardware, database instance.Be more efficient by grouping tests with similar setup requirements – platforms, user personas.Analyze risks – complexity, risk of harm.Provide status updates – test coverage, productivity.Identify out-of-date or stale tests – suspect flag, workflow status.What additional information you include with each test case will depend on your project template and test management process.BLOG: HOW TO WRITE (BETTER) TEST CASES IN JIRA >>Back to top The Easier Way to Write Test CasesAll of this information will help you craft better test cases. But they can be even easier to create when you switch from managing tests with spreadsheets and Word docs to a proper test case management solution.Perforce TCM lets you write, execute, and track tests with ease. Use it on its own for test management or as part of the Perforce ALM (formerly Helix ALM) suite. Even pair it with your existing tools, including Jira.You can try it out for free. Use the buttons below to download your 30-day trial or watch a demo of Perforce ALM to get a complete overview of the solution.Manage your test cases for free. See Perforce ALM in action. Test cases are designed to verify that your application is operating as expected. Test case writers design test cases so testers can determine whether an app or software system's feature is working correctly. Applications must be tested thoroughly to find out how the system behaves under all possible input conditions. A clear understanding of software functions and the testing process can make writing tests that identify defects easier. You can use the following walkthrough to learn more about writing and formatting different types of test cases. Afterward, strengthen your testing skills by earning the Google IT Automation with Python Professional Certificate. Learn more about test automation, troubleshooting, and debugging.What is a QA test case?Test cases are instructions for testers to follow to ensure programs are functioning properly. They describe how the software should work in normal, abnormal, or error operating conditions. Test case writing converts user requirements into a set of test conditions and descriptions that indicate how a system is functioning. In an automated test script, more than one test case can combine to form a test suite.If you're practicing your test case writing skills to advance your career, you might consider earning a credential to demonstrate your efforts. For example, you can earn a certificate in Software Testing and Validation from the University of Leeds in just five hours. Or, put some practical experience on your resume by creating your first test automation script using Selenium and Java. You'll be guided through the process and complete a project for your portfolio in just two hours with the following Guided Project:Test case vs. test scenario When writing a test case, you typically think about every detail of "how" something should behave. For example, if you’re developing a login system, a test case might be that an error is displayed if you enter an incorrect email address. Then you may have tests for:Not entering any email addressAdding a space at the end of an email addressUse all caps for the email addressCapitalizing the first letter of the email addressTest scenarios, also called test conditions or test possibilities, represent typical tasks users might want to accomplish with the software. A simple test scenario may require multiple test cases to cover the different outcomes. In our login system, some scenarios could be "I can successfully log in" or "I can't log in without entering my email address correctly."TIP: Sometimes, the terms test case and test scenario seem interchangeable—but it's helpful to remember that "scenario" refers to the bigger picture of what you're trying to accomplish with your tests, and a "test case" focuses on the minute details of the test scenario. Types of test cases Test cases can be categorized based on the purpose they serve in testing. As a quality assurance professional, knowing the difference between them helps focus your efforts and choose the right test format. Functionality test cases: These are the most basic and obvious test cases to write. They ensure that each feature of your system works correctly. Performance test case: This test ensures that the system runs fast enough. It makes sure that all system requirements work as expected regarding speed, scalability, or stability. Unit test cases: Software developers usually write unit tests for their code to check individual units, for example, modules, procedures, or functions, to determine if they work as expected. User interface (UI) test cases: It’s important to remember that the user interface is part of the overall system and not just a shell where functionality appears. UI test cases check that each UI element works correctly, displays, and is easy to use. Security test cases: Security test cases help ensure that a product or system functions properly under all conditions, including when malicious users attempt to gain unauthorized access or damage the system. These test cases safeguard the security, privacy, and confidentiality of data.Integration test cases: These ensure that the application components work together as expected. These test cases check whether modules or components integrate seamlessly to form a complete product.Database test cases: These test cases ensure that the database meets its functional and non-functional requirements. They make sure database management systems (DBMS) support all business requirements.Usability test cases: Usability test cases help check if users can use the application successfully. These determine whether users can easily use the system without difficulty or confusion. They also verify if users can navigate the system using common procedures and functions.User acceptance test cases: User acceptance test cases verify that an application satisfies its business requirements before users accept it. These determine whether users accept or reject the output produced by a particular system before release to the live environment.Regression testing: Regression test cases verify that changes made during development don’t cause any existing functionality to stop working. Regression testing happens after changes have been made to existing code to test that all existing or legacy functionality continues to work as expected after making the changes.Who writes test cases? Writing test cases is normally the responsibility of someone on the software development team, testing team, or quality assurance team. It’s typically preferred that a professional who was not involved with writing the code should write the test cases since they have a fresh perspective. Different software development frameworks like Agile and Scrum may vary in their QA testing approaches. Read more: Agile vs. Scrum: How to Choose the Best MethodHow to write test cases: A step-by-step guideTest cases are the blueprints that testers will follow, so they must be clear, thorough, and accurate. Below, we've outlined 10 steps you can take whether you're writing new test cases or revisiting and evaluating existing test cases.Define the area you want to cover from the test scenario.Ensure the test case is easy for testers to understand and execute.Understand and apply relevant test designs.Use a unique test case ID.Use the requirements traceability matrix in testing for visibility.Include a clear description in each test.Add proper preconditions and postconditions.Specify the exact expected result. Utilize suitable testing techniques. Get your test plan peer-reviewed before moving forward.Best practices for well-written test casesYou can use the tips in this section with the list above to create an efficient but thorough workflow. Let's start with two quick tips:Make test cases reusable and maintainable wherever possible. Your needs will vary depending on the software, application, or specific features you're testing. However, you can save time and energy by consciously creating test cases that are reusable and easy to maintain. Create test cases with the end user's perspective in mind. Remember throughout the test case writing process that you're trying to step into the user's place. Aligning your exploratory testing methods with the user's perspective will help you create efficient and relevant software application test cases.Consider creating a reusable test case template.A test case template provides a flexible but basic structure that you can customize on an as-needed basis. It can also serve as a checklist to ensure all essential elements have been included. Many testers use spreadsheets with one test per row and the test elements in columns. Here are a few elements you can add to your test case template:Test Case IDTest Case DescriptionPre-ConditionsTest StepsTest DataExpected ResultPost ConditionsActual ResultStatusBenefits of writing high-quality test casesWriting test cases allows you think through every aspect of your software and makes it easier to identify any software gaps as it develops. Several of benefits emerge when you write formal test cases. Documentation means you can guarantee the coverage of your tests.You can reduce software maintenance and bug fixes and support future costs. Test cases can be used again in current and future projects.You can improve the quality of the software and the user experience.A higher quality product means more satisfied customers and higher profits.Test case management tools help you manage software and hardware development. These tools track your test cases, bugs, and other important information related to testing. Countless testing tools are available in the market. If you’re applying for test case writer jobs or a similar role, you might need prior experience with one or more of these tools: JIRAJunoOneKlaros-TestmanagementQACoverageQaseSPIRATEST by InflectraTestFLO for JIRATestpadXQualXrayZephyr ScaleZephyr SquadStrengthen your testing skills with Coursera PlusLearn job-ready skills from industry leaders like Google, Microsoft, and IBM with a Coursera Plus subscription—available in monthly and annual tiers. You’ll get a certificate for every program you finish, which you can add to further enhance your resume. A test case in software testing is document that consists set of conditions or steps designed to verify if a particular feature/functionality in a software application is working as expected. It defines the inputs, actions, and expected results for a specific test scenario to ensure the software behaves correctly under various conditions. A test case typically includes information like the test case ID, description, steps, preconditions, input data, execution steps, expected outcome, and any post-conditions or cleanup required.Test cases are essential for validating the functionality, performance, and reliability of software applications.It will lay out particular variables that QAs need to compare expected and actual results to conclude if the feature works. Test case components mention input, execution, and expected output/response. It tells engineers what to do, how to do it, and what results are acceptable.Read More: How to create Test Scenarios with ExamplesThe Objective of Writing Test Cases in Software TestingTo validate specific features and functions of the software.To guide testers through their day-to-day hands-on activity.To record a catalog of steps undertaken, which can be revisited in the event of a bug popping up.To provide a blueprint for future projects and testers so they don’t have to start work from scratch.To help detect usability issues and design gaps early on.To help new testers and devs quickly pick up testing, even if they join in the middle of an ongoing project.Standard Test Case Format Test Case IDTest ScenarioTest StepsPrerequisitesTest DataExpected/Intended ResultsActual ResultsTest Status - Pass/Fail While writing test cases, remember to include:A reasonable description of the requirementA description of the test processDetails related to testing setup: version of the software under test, data points, OS, hardware, security clearance, date, time, prerequisites, etc.Any related documents or attachments testers will requireAlternative to prerequisites, if they existTest Case Prioritization is vital while writing test cases in software testing. Running all the test cases in a test suite requires much time and effort. As the number of features increases, testing the entire suite for every build is practically impossible. Test case prioritization helps overcome these challenges.How to write Test Cases (Test Case Example)Let’s build a test case example based on a specific scenario. Here is a sample case.Test Case ID: #BST001Test Scenario: To authenticate a successful user login on Gmail.comTest Steps:The user navigates to Gmail.com.The user enters a registered email address in the ‘email’ field.The user clicks the ‘Next’ button.The user enters the registered password.The user clicks ‘Sign In.’Prerequisites: A registered Gmail ID with a unique username and password.Browser: Chrome v 86. Device: Samsung Galaxy Tab S7.Test Data: Legitimate username and password.Expected/Intended Results: Once username and password are entered, the web page redirects to the user’s inbox, displaying and highlighting new emails at the top.Actual Results: As ExpectedTest Status – Pass/Fail: PassOnce test cases have been shaped, corresponding tests must be run on real browsers, devices, and operating systems. Remember that device fragmentation is a significant concern for every developer and tester. Every website has to work seamlessly on multiple device-browser-OS combinations. With 9000+ distinct devices being used to access the internet globally, all software must be optimized for different configurations, viewports, and screen resolutions.Test on Real Device CloudHow to Write and Manage Test Cases Using BrowserStack Test Management?Creating and managing test cases using BrowserStack Test Management are simple and efficient methods. Here is a step-by-step guide to follow:Creating Test Cases Using Test Management ToolAccess BrowserStack Test Management: Register or log in to your BrowserStack Test Management account and go to your project.Navigate to Test Cases Section: Find and access the “Test Cases” section within your project.Create a New Test Case: Start by clicking on “New Test Case” to create one.Define Test Case Details: Enter the test case title, description, and tags.Add Test Steps: Outline your test case with clear, actionable steps for thorough testing.Set Test Case Priority Level: Assign a priority level to manage the test execution effectively.Save and Review the Test Case: Ensure the test case is accurate and complete before finalizing.Talk to an ExpertManaging Test Cases Using Test Management ToolAccess Test Cases Section: Log into your BrowserStack Test Management account, navigate to your project, and enter the “Test Cases” section to start managing your test cases.View and Organize Test Cases: Look over the list of test cases in your project and arrange them by priority or other relevant criteria to enhance management.Edit Test Cases: Select a test case to see its details and edit information such as the title, description, test steps, and priority as necessary.Add or Remove Test Cases: Integrate new test cases by clicking on “New Test Case,” or delete outdated or unnecessary ones to keep your repository organized.Execute Test Cases: Run the test cases, monitor the progress of test executions, and track the results within the Test Management platform.Review Test Results: Post-test execution, analyze the outcomes, identify any issues or bugs, and classify them for fixing.Update Test Cases: Modify test cases based on the results or any changes in the application functionality. Why use BrowserStack Test Management Tool for Writing Test Cases?BrowserStack Test Management Tool offers several features that make it effective for creating and managing test cases for web and mobile applications:Efficient Test Case Creation: Use ready-made templates in text-based or steps-based formats for fast, consistent, and streamlined test case creation.Effortlessly Import from Zephyr Scale, Xray, TestRail, or CSV.Enhanced Test Case Authoring: Utilize features like quick additions, detailed forms, and a rich text editor for clear and comprehensive test cases, improving software quality.Customizable Test Cases: Adjust test details like priority, type, and status to meet project specifications, enhancing test accuracy.Efficient Test Case Management: Manage test cases effectively with bulk actions—edit, copy, move, and delete—boosting productivity.Advanced Search and Filtering: Find and organize test cases quickly with filters like status, Test Case Priority, ID, Title, and more.Integrated Workflows and Insights: Monitor test cases and runs with real-time insights into software releases, performance metrics, and trends.Robust Reporting: Utilize advanced reporting tools to track progress, pinpoint issues, and evaluate test outcomes, ensuring detailed quality assessments.Common Features of Test CasesLikely to be revised and updated regularly: Software requirements can change depending on business priorities or customer preferences. If requirements change, test cases will have to be altered accordingly. The detection of bugs and debugging steps may also require test cases to be changed.Likely to involve clustering: Test cases in a single test scenario usually have to be run in a specific sequence or in a group. In this case, particular prerequisites of one test case will apply to other cases in the same sequence.Likely to be interdependent: Often, test cases can depend on each other. This is especially true for layered applications with multi-tier business logic.Likely to be used by testers and developers: Test cases are helpful for developers and testers. For example, when devs fix bugs, test cases can be pretty valuable to replicate the said bug. In Test-Driven Development (TDD), devs create test cases to craft business logic, cover multiple test scenarios, and start writing code.Also Read: How to write Test Cases for Mobile ApplicationsBest Practices for Writing Test CasesHere are the best practices for Writing Test Cases:Prioritize clarity and transparency. Be clear, concise, and assertive in describing what the tester needs to do and what results they should ideally get.Focus on End-User requirements when writing sample test cases. Map test cases to reflect every aspect of the user journey. Use the Specifications Document and the Requirements Document to do so.Avoid repetition. If multiple tests can be executed with the same test case, use the Test Case ID to refer to the required test case.Keep Test Steps as minimal as possible. Ideally, keep it to 10-15 steps, if possible.Focus on achieving maximum test coverage. While 100% test coverage is rarely achievable, a high percentage can be attained with the right strategies.Create self-cleaning test cases. Test cases must revert the Test Environment to a pristine, pre-test state. Tests should not leave remnants of themselves in the environment when completed. This is an integral element of Configuration Management.Shape test cases for tests that return the same results no matter who runs them. Ensure that tests are self-standing.Access BrowserStack Test Case ManagementTo sum up, the foundation of successful testing lies in creating well-structured and result-oriented test cases. The BrowserStack Test Management Tool provides an intuitive platform and a robust framework that ensure comprehensive test coverage and offers a clear, detailed roadmap for QA teams to follow. By utilizing this tool, teams can enhance their testing efficiency and accuracy, leading to more reliable software and better end-user experiences.