



User acceptance test form

User acceptance testing (UAT) templates are essential tools for quality assurance teams to plan and execute user acceptance tests. These documents help ensure that end users face minimal issues during testing and that software functions as expected in real-world business scenarios. having the right processes and workflows in place is crucial. To achieve successful UAT, you'll need two key templates: a UAT test plan template and test case templates. The UAT test plan template and test case templates and the testing process strategy to be used during execution. It also covers scope, objectives, milestones, deliverables, environmental requirements, and features to be tested. The test case template provides real-world scenarios and step-by-step instructions for testing specific software or website app features. To find a suitable template for your organization, consider factors such as the tools you use (e.g., Jira, GitHub, Trello, GitLab, Asana) or how you manage backlogs (Excel spreadsheet, Word document, email). By utilizing these templates, you can ensure that your UAT testing is well-planned UAT testing is well-planned use companies valuable development hours and resources. 1. To successfully execute UAT, ensure that your test cases include the following essential elements: ID or Name. Description. Test Steps (keep it brief but clear). Expected Result. 2. For teams using GitHub, you can use a template like this: - As you can use a template li Marker.io automatically collects additional data such as Reporter Name Source URL Screenshot Console logs and session replay Environment Info (device type, browser, screen size, OS...) 4. For teams using Jira, a template like this can be used: - As you can see, this issue type contains all the necessary fields for UAT, including ID, description, test steps, expected result, and actual result. 5. Similarly, Trello users can utilize a template with these same essential fields: - As you can see, this card contains the basic elements required for UAT testing, including ID, description, test steps, expected result. and actual result. 5. Similarly, Trello users can utilize a template with these same essential fields: - As you can see, this card contains the basic elements required for UAT testing, including ID, description, test steps, expected result. non-technical users participate in User Acceptance Testing (UAT), it's essential to collect accurate data. Marker.io automates this process, ensuring that valuable insights aren't lost or misreported. Key Data Points Collected: * Reporter name * Source URL * Screenshot * Console logs and session replay * Environment info (device type, browser, screen size, OS...) With Marker.io, end users can easily capture issues and add annotations to convey their concerns. This data is then converted into a new card on popular collaboration platforms like Trello, GitLab, or Asana. Benefits of Using Marker.io: * Collects valuable insights from non-technical users * Automates reporting, reducing errors and inaccuracies * Integrates seamlessly with various platforms, including Trello, GitLab, and Asana Consider signing up for a free Marker.io, adding notes to highlight key points. With just one click, the tool converts this into a new Teamwork task. If your team uses Teamwork, sign up for a Marker.io trial to explore its features. For Shortcut users, a markdown format is available for test scenarios: create a custom Status field with 'Pass' and 'Fail' values. This template includes basic fields like ID, description, test steps, expected result, actual result, and status. If you integrate Marker.io with Shortcut, additional information such as reporter name, source URL, screenshot, console logs, session replay, environment details will be automatically included. Similarly, for Monday users, a format is provided for test cases: this task contains basic fields like ID, description, test steps, expected result, and status. If you use Marker.io with Monday, data such as end user name, source URL, screenshot, console logs, session replay, environment info will be added automatically. For ClickUp users, a template is available for test cases: this task includes basic fields like ID, description, test steps, expected result, actual result, and status. If you integrate Marker is with ClickUp, data such as user name, source URL, screenshot, console logs, session replay, environment info will be automatically included. In each case, users can capture a screenshot using Marker.io, add annotations to highlight key points, and convert this into a new task for the respective tool with just one click. Sign up for a free Marker.io, add annotations to highlight key points, and convert this into a new task for the respective tool with just one click. following info: * User name * Source URL * Screenshot * Console logs and session replay * Environment info (device type, browser, screen size, OS...) You can take a screenshot with Marker.io and add annotations to report issues. Download: Wrike Template (.xlsx) Google Spreadsheets Format: ``` ID | Description | Test steps | Expected result | Actual result | Status ``` Download: Google Sheets Template Microsoft Word Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` Download: Google Docs Test Case Template: ``` ID | Description | Test steps | Expected result | Status ``` ID | Description | Test steps | Expected result | Status ``` ID | Description | Test steps | Expected result | Status ``` ID | Description | Test steps | Expected result | Status ``` ID | Description | Test steps | Expected result | Status ``` ID | Description | Test steps | Expected result | Status ``` ID | Description | Test steps | Expected result | Status ``` ID | Description | Test steps | Expected result | Status ``` ID | Description | Test steps | Expected result | Status ``` ID | Descripti Template PDF Test Case Template (for organizations requiring PDF files): Download: PDF Template (.pdf) Email Test Case Template: ``` ID | Description | Test steps | Expected result | Actual result | Actual result | Status ``` Download: Email Template (.txt) ``` User Acceptance Testing (UAT) is a crucial stage in the software development process that helps determine whether the final version meets the client's business needs. It involves checking the system's functionality, usability, and overall performance to ensure it meets user requirements. This testing type of work is essential because no developer can fully understand the business specifics of each new client. UAT is conducted after QA testing and is a collaborative effort between the development team, real users, and business stakeholders. UAT aims to verify that the system can perform its functions effectively in real-world conditions, ensuring it delivers a positive user experience and solves the intended problems. UAT helps avoid unsuccessful product launches due to functionality and usability issues, reducing the need for costly fixes during production. The benefits of UAT include increased customer satisfaction, improved profitability, and reduced costs associated with fixing defects after launch. By understanding the peculiarities of organizing and conducting UAT, developers can create software products that fully meet user requirements, making them more sought after and valuable in the eyes of clients. Throughout the development process, norms are crucial. Operational testing Alpha testing involves internal testers using real-world scenarios, while beta testing is performed by external users. Note that UAT uses black-box testing, except for alpha testing which can use white-box approach. Conducting PayPal payment systems) and business requirements (e.g., offering four payment options). Clear acceptance criteria ensure understanding of what needs to be tested. After completing UAT, analyze whether software fulfills its functions and helps achieve company goals. To plan the testing process, document project information before it begins, including testing strategy, test cases, scenarios, entry and exit criteria, etc. In this article, we will explore different types of documents used in UAT. UAT can take place at various stages in the software development life cycle, depending on the development life cycle, depending on the development methodology. For example, in Waterfall approach, acceptance testing is the final stage, while Agile teams conduct it after each iteration. The advantage of Agile is that test results can be used to continuously improve the product throughout its lifespan. The type of testing needed depends on the software solution's requirements. For instance, you may need to verify compliance with functional or non-functional requirements. As we previously mentioned, the composition of the test team may vary depending on the user acceptance stage. For alpha testing, internal company specialists can be involved, such as stakeholders, product owners, business analysts, and others. Real customers are best suited for beta testing and can be found on crowdsourcing platforms or attracted through landing pages and social network messages. The process of UAT testing has many tools to optimize the whole process. Specialized crowdsourcing platforms, test exploratory tools with screenshot capabilities and comment features, project management systems like Jira or Trello, and Test Management Systems (TMS) are available. Our example test management tool, testomat.io, offers limited free subscription, one-click reporting, quick creation of test cases, detailed reports, analytics for business requirements alignment, one-click import/export, easy automation, and stakeholder engagement. User training is necessary to perform tasks and fill out the User Acceptance Testing Checklist accurately. This process includes familiarizing participants with technology, tools, testing objectives, bug reports, and checklist principles. QA team managers are completed, UAT can be conducted directly. It's essential to control the execution process, explain requirements at each stage, and provide real data to users to understand the product accurately and obtain precise results. After completing UAT, developers must gather feedback, organize information, analyze data, and generate detailed testing reports manually or with specialized tools, which provides percentage passed/failed test graphs for easier understanding. The effective fixing of bugs by a team is vital in meeting user needs. To facilitate issue reporting, tools like Testomat.io integrate well with project management system (TMS), it's easy to link defects to user stories directly from test pages. After fixing the issues, re-validation of the software product is necessary to ensure that it meets the user acceptance criteria. A test plan should be created beforehand, including essential information about the upcoming test run and familiarizing oneself with user acceptance testing examples. For a smooth workflow, using a User Acceptance Testing Template and examples are crucial. The template typically includes: * Project Name/Release Version * Testing Objectives * Change Log * Introduction to the Document * Test Items * Detailed Action Plan * Environment * Test Items * Detailed Action Plan * Environment * Test Items * Detailed Action Plan * Environment * Test Items * Detailed Action Plan * Different Tools Used for UAT These documents can be fully customized to meet each team's needs. Writing test cases for User Acceptance Testing (UAT) is essential, providing instructions on how a specific function of the software product should be tested. Various templates and tools can assist in writing them, including those provided by popular project management systems like Jira, Gitlab, and xlsx tables. For creating reports on user acceptance testing (UAT), consider using templates in Microsoft Word or Google Docs. These documents should analyze test results and identify bugs for prompt resolution by the development team. Key components of such reports include: A comparison of test outcomes with predefined user acceptance criteria Information about identified defects, including severity and priority of resolution Action plans and deadlines for resolving issues These reports must be concise and clear to facilitate understanding among technically proficient team members, stakeholders, project managers, and product owners. Effective UAT requires thorough preparation. This involves creating a detailed testing plan, selecting user-friendly test management tools, and compiling comprehensive results reports. When choosing test management tools, consider their integration capabilities with bug-tracking systems, test case creation, and report generation. Our test management system offers advanced features in these areas and can optimize your UAT processes. For more information on utilizing our system, contact our manager for detailed guidance. This acceptance test approval form is used to document that the individual performing user acceptance testing has understood the importance of this task and acknowledged their responsibility as a tester.