UAT Initiation and design

Date	15November 2022
Team ID	PNT2022TMID17224
Project Name	Project - IoT based Industry -Specific Intelligent Fire Management System
TEAM LEADER	GAUTHAM K
TEAM MEMBERS	GOKULA KRISHNAN A GOWTHAM A GOWTHAM S

User Acceptance Testing (UAT) checks whether a product is the right one for the end users. It has other names, e.g., *end-user testing*, *operational*, *application*, *beta testing*, or *validation* but they describe the same thing. In quality assurance, it's important to distinguish between *validation* and *verification*.

Verification refers to general QA processes aimed at testing the technical aspects of a product to ensure it actually works. **Validation** (or user acceptance testing) is conducted to make sure that the product corresponds with business requirements and can be used by the end user.

Alpha testing is the initial stage of acceptance testing, typically performed by internal testers, to ensure that the product functions correctly and meets business requirements.

Beta testing, the second type of acceptance testing, aims at meeting user acceptance criteria. UAT can be performed by

- the actual users of an existing product,
- users of a previous version of a product,
- stakeholders involved in the development of the product, and/or business analysts as end-user specialists.

This enables the development team to fix most of the usability problems, bugs, and unexpected issues concerning functionality, system design, business requirements, etc.

Play

Ensure correspondence with business requirements. As we already mentioned, UAT is done to verify that the product operates in the real-world circumstances as required and allows end users to solve targeted problems. If you skip UAT, you

might miss out on some important flaws or system malfunctions that will inevitably cause user dissatisfaction.

Adjust initial requirements. Sometimes, as end users test the product, they can come up with some valuable thoughts on how to improve the tested software. Getting such feedback will allow you to adjust your requirements to get a result that will be more useful for your customers.

Avoid losses. First, it's cheaper to fix the product in the early stages of development, so finding flaws due to UAT will allow your development team to improve the product much more easily (that mostly concerns the Agile model though. Read on for more details). Second, we all know stories about product failures because of poor functionality and usability. UAT provides you with realworld user feedback and makes it far less likely to have losses caused by an unsuccessful product launch.

In any case, UAT requires organization and preparation work to make it effective. If you want to ensure your product's validity, consider the following steps in conducting user acceptance testing.

User acceptance testing deliverables

UAT test plan. Creating a UAT test plan will help you keep everybody aligned with the same objectives and vision. The main document, it includes all the information concerning what will be tested, by whom, and how. To cover all the organizational and processual aspects of UAT, you have to detail the testing strategy and entry/exit criteria.

End-user testing strategy. The strategy outlines the product you are testing, the purpose of user-acceptance testing, types of tests, and objectives. Your testing strategy should cover such information as

- product description,
- testing objectives,
- testing scope,
- standards,
- testing types,,
- testers/roles
- process curators (managers),
- reviewers,
- reporting standards, and outcomes.

Entry criteria. These are the conditions that establish that the software is ready to be tested. They are set at the earliest stage of planning by the development team, QA, business analysts, and stakeholders.

Exit or acceptance criteria. These are the conditions that dictate that the software is valid for the users. Matching acceptance criteria would be the final stage of your
UAT.

Test scenarios. Test scenarios are hypothetical situations that users may encounter when interacting with your product. Their aim is to guide your testers through possible system usage problems.

Basically, a test scenario should convey a simple idea of what will be tested. An example of a scenario is "check shopping cart functionality." Each user scenario is connected with one or two requirements or user stories. They are written to validate that the system is usable, checking the end-to-end operations with real data.
To write good test scenarios for user acceptance testing, consider involving end users in approval to include all the possible use cases, both common and uncommon. Also, consider writing them in plain language, avoiding complicated phrasing or overly techy explanations.

Test cases. A test case is a set of specific actions that are taken to test and verify a particular system behavior, feature, or functionality. Test cases are more detailed units that have to correspond with all the test scenarios. Most
often you will convert your user stories and business use cases to write efficient test cases. Examples of test cases are:

1. Check unregistered user adding the product into the shopping cart.
2. Check shopping cart filtering.
3. Check the "continue shopping" button.

Test cases are efficient when there is a clear purpose stated, and the user is able to understand what they should do to complete it. The user guide to a test case may look like this:

1. Open the application.
2. Add any product to a shopping cart.
3. Authentication is not needed.
4. Proceed to the shopping cart.

You may also include expected results in the test case, so that the user is aware of what is going to happen:

1. The product will appear in a shopping cart.
2. The system will ask you to authorize as a registered user.

Reporting standards. Define how a report should look and what information an end user should provide.

Test reports. These accumulate documented output data when the test is completed. Depending on the testing standards and testing scenario, various information can be included in reports. But typically in UAT, QA teams will require only a sign-off from the tester. A sign-off is just a confirmation that the test is successful and it corresponds to the user's criteria.

At the end of UAT, deliverables provided can be used by QA engineers or a UAT manager to extract valuable data and communicate results to the development team.

Traditionally, quality assurance engineers will be responsible for processing enduser feedback. The results of tests, bug reports, and fail/pass records are provided to developers to ensure constant communication between different parts of the team. Based on the end-user feedback, the QA team can also provide software quality metrics to measure progress in terms of UAT.

User acceptance testing templates

We've mentioned a few important documents that have to be created for proper UAT planning and execution. There are different ways to write them, but here are some templates that may come in handy.

- Test plan templates: Test Plan template by Coley Consulting, sfsu template (downloadable link), or iiba template (downloadable link)
- Test scenario template
- Test report template


Business Use Case
Customer Specifications
Requirements
Design
Coding
Release Testing
Acceptance Testing
System Testing
Integration Testing
Unit Testing
Product Development