imc

part of Scheer

User Acceptance Testing Guidelines

Project Guide

imc Learning Suite

May 07, 2024

im-c.com

Information

Testing, in whatever way you choose, is important for the success of the project. That is why we plan with a dedicated testing phase in every project.
Since you, the customer, are responsible for this phase of the project, we provide this guideline to help you plan and conduct the tests.
Because the better you are prepared for this phase, the more value you will get from it.

A popular approach to testing, is the User Acceptance Testing (UAT).
It is typically the final phase of testing before the system is officially rolled out for all users. It involves testing real scenarios by end-users or their representatives to make sure the con-figurations and workflows meet the expectations and needs of those users and allow to ad-just according to their feedback.

The following will give a general overview of all the steps surrounding UAT and an example on how to construct test-cases for you LMS-project, for imc will not provide detailed test plans or conduct the testing phase together with you.
Even just using parts of the entire process can hugely benefit your project.

Steps of setting up and performing a UAT

The following list describes the steps of setting up and performing a standard UAT.
Regarding your LMS project you may find that the scope needs to be adjusted to fit the timeframe of the project. On the other hand, the first three steps will come naturally whilst specifying the configuration of the system with your imc Business Consultant, but it is your responsibility to keep track of this information and use it for your preparation and execution of the testing phase.

1. **Define test objectives**
   Clearly outline the goals and objectives in your scenario, including what needs to be tested and what outcomes are expected.

2. **Define test groups**
   Define the separate roles you have in the system, as they usually have quite different tasks and workflows and therefore need separate test-cases.
   Depending on the complexity of your scenario and system you can use the main navigation roles (e.g., Learner, Admin, Supervisor, Tutor) as a guideline for those. But if you have groups within those roles, that differ in access, responsibilities, or workflow, it is essential to create dedicated test-cases for them.

3. **Identify test participants**
   Select participants to conduct the tests. They should ideally be or represent the actual end users. Since the time for testing during a project is limited, we recommend making the selection as early as possible and have them block time for the testing phase, so you are less likely to have timing issues.

4. **Create test cases**
   Develop test cases based on the requirements for each role/group. These test cases should cover various scenarios to ensure comprehensive testing.

5. **Execute test cases**
   Have the test participants execute the prepared test cases, documenting any issues or feedback encountered during testing.

6. **Monitor and support**
   Be available to assist test participants, answer questions, and address any issues that arise during testing. During the test-phase of a project your imc Business Consultant will assist with issues that might arise.

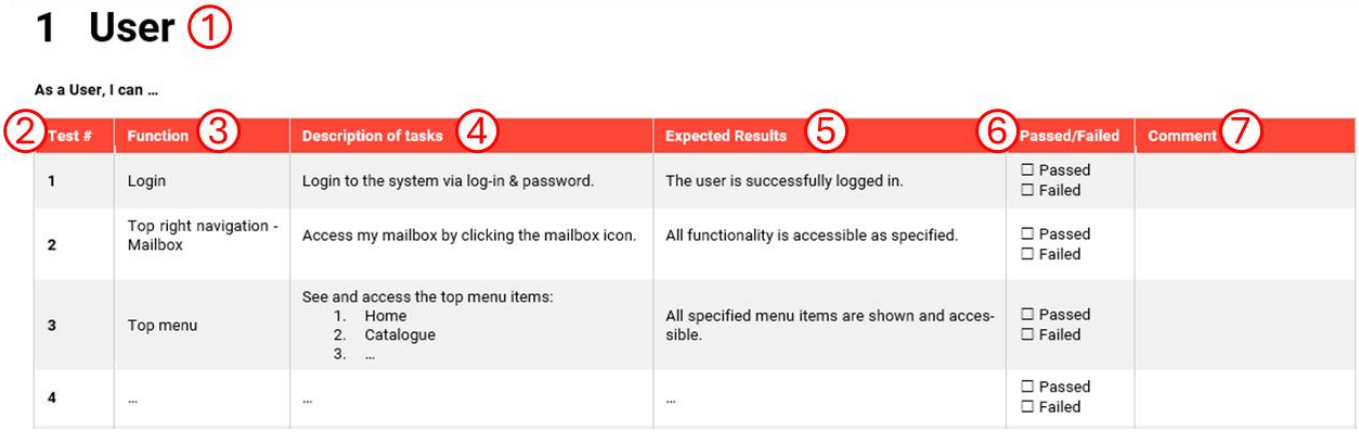
7. **Review and analyse results**
   Review the test results, analyse any issues or feedback provided by the participants, and determine whether the configuration meets their needs. Communicate key results with your imc Business Consultant to find a realistic approach.

8. **Iterate and retest (if necessary)**
   If issues are identified during testing, address them, make necessary adjustments, and conduct additional rounds of testing as needed until the system meets the acceptance criteria. Be aware of the limited time within a project and that a major delay may require additional budget.

How to construct test cases

There is no single right way to write test cases but the "User Acceptance Test – Template" is a viable way to do it and will be used as an example to explain the general construction of test cases and the relevance of all the components, to enable you, the customer, to create effective test cases for the testing phase on your own.

The template is made up of several tables in a docx-file, as pictured below, but the concept can also be applied to Excel or other similar tools.



1 User

As a User, I can …

Test #	Function	Description of tasks	Expected Results	Passed/Failed	Comment
1	Login	Login to the system via log-in & password.	The user is successfully logged in.	☐ Passed ☐ Failed	
2	Top right navigation - Mailbox	Access my mailbox by clicking the mailbox icon.	All functionality is accessible as specified.	☐ Passed ☐ Failed	
3	Top menu	See and access the top menu items: 1. Home 2. Catalogue 3. …	All specified menu items are shown and accessible.	☐ Passed ☐ Failed	
4	…	…	…	☐ Passed ☐ Failed	

1. **Role (User)**
   As the tasks and available functions in the system vary for the separate roles, the test cases must be specified and assigned accordingly.
   Some tasks may be the same for every role and can be collected in a general user role, as seen in the example above.
   The template differentiates only between the standard roles since it is a general example. If you have more roles or sub-roles (e.g., content administrator, system administrator etc.), the file needs to be adjusted accordingly.

2. **Test #**
   To number the test cases consecutively gives a good overview and makes it easier to reference specific test cases. In our template the cases are numbered per role.

3. **Function**
   Oftentimes you have more than one test case for the same general area in the system (e.g., course manager, management dashboard etc.). Defining them in the "Function" column, helps clustering those cases and gives a good overview.

4. **Description of tasks**
   These descriptions tell your participants what to do and should therefore be easy to understand.
   Depending on your scenarios you might need to cover more details with your descriptions:
   Create new media **vs.** Create new media of type WBT.
   Create new course **vs.** create new blended learning course.
   A more general description will give you natural results, as the users are not told how to exactly achieve the task, which can result in valuable feedback.
   Nevertheless, they should still have enough information to realistically bring the expected results (see step 5) and cover your key scenarios.

5. **Expected results**
   The expected results give participants a reference to decide if everything worked as intended in the next step. In our template the results are worded very broadly, but as the system has been specified according to your scenarios throughout the project, you will need to go into more detail here (e.g., which panels exactly should be visible on the dashboard).

6. **Passed/Failed**
   These fields let you quickly see, which test cases were causing issues.

7. **Comment**
   The comments give you an insight into what went wrong if a test case was marked as failed or if participants have additional feedback to a certain scenario, even if it was successful.