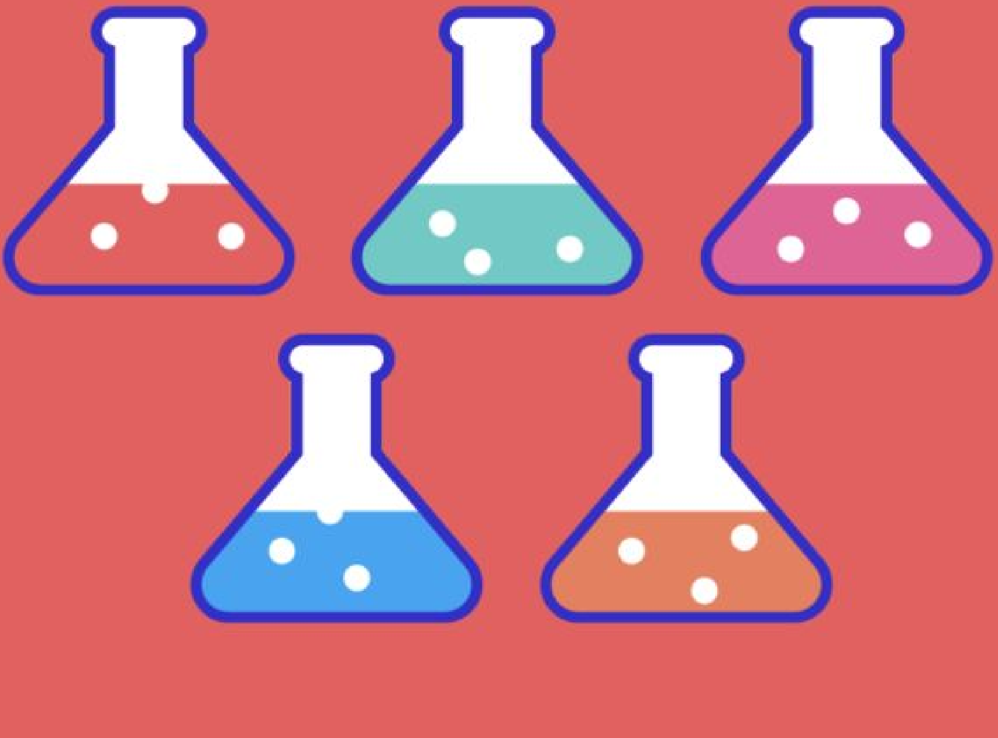
How to rock UAT

How to double the deployment speed of your product or acceptance of your project with User Acceptance Testing



usersnap

usersnap

What you will learn?

This ebook provided by Usersnap will give you guidance

- What User Acceptance Testing (UAT) is?
- When you should start with UAT?
- Who should be involved in UAT?
- User Acceptance Testing frameworks
- Is there a UAT tool?
- 5 steps for your best-practise UAT workflow
- How to rock UAT :) Enough?

What is User Acceptance Testing?

User Acceptance Testing (UAT) is the last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to the market.

User Acceptance Testing (UAT), also including Beta-, Application-, or End-User Testing, is often considered the last phase in the web development process, the one before final release or installation of the website or software for the client, or final distribution of it.

UAT is the usage of the software by people from the intended audience and recording and correcting of any defects which are discovered. It's the closest thing to a "_real world_" test available. It gives users the chance to interact with the software and find out if everything works as it should if features have been overlooked, miscommunicated, not communicated, and so on.

DevelopMentor puts it most succinctly when they describe user acceptance testing (UAT) as:

> ***"The goal of User Acceptance Testing is to assess if the system can support day-to-day business and user scenarios and ensure the system is sufficient and correct for business usage."***

When Should You Start User Acceptance Testing

Whilst User Acceptance Testing is essential, typically, it's not able to be undertaken until the application is largely feature-complete.

Guru99 lists 10 prerequisites, which must be met before UAT can begin. These are:

- Business requirements must be available
- Application code should be fully developed
- Unit testing, integration testing & system testing should be completed
- No show stoppers, or high or medium defects in the system integration test phase
- Only cosmetic errors are acceptable before UAT
- Regression testing should be completed with no major defects
- All the reported defects should be fixed and tested
- Traceability matrix for all testing should be completed
- UAT environment must be ready
- Sign off mail or communication from system testing team that the system is ready for UAT execution

Recommended Reading:

- 6 Tips For a Successful User Acceptance Testing Plan
- How to create a product roadmap with Usersnap Labels
- 4 reasons why you should integrate visual communication into your workflow

Who should be involved in User Acceptance Testing?

The most important peer group to include in UAT are “real” end users of your software. Every role and stakeholder group should be included which means that people from each group should be selected to join the UAT team.

How To Get Started With User Acceptance Testing

Now that we’ve laid the groundwork for what UAT is and why it’s essential, let’s finish up by seeing how to get started. Normally, UAT consists of four steps. But it can vary, based on whether the application is being delivered to a single customer, or whether it’s intended to be off-the-shelf software, available for purchase by anyone.

Firstly, the criteria which the software is considered to be “working” needs to be assembled. These are likely to be collated from the system requirements, and user stories. Next, a set of UAT test cases must be created. Centric defines a UAT test case as:

A set of test steps, execution conditions and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement

Each case covers a specific usage scenario of the software. It is normally a set of actions which the user can carry out and be able to verify if the software’s worked as intended.

With these in place, the tests then have to be run and the results recorded. Were the tests successful, or did defects result? Any bugs then need to be corrected and re-tested.
Finally, assuming that everything is working as expected, an orderly sign-off needs to be completed. This is done with your client or the team they have assembled for the project, where they state that what they've received works as expected and meets their criteria.

User Acceptance Testing framework

User Acceptance Tests might not be called User Acceptance Tests in your organizations. There are various buzzwords – such as alpha or beta testing – out there. And sometimes we also get asked about the differences between UAT and functional testing.

Therefore we decided to collect all our thoughts and knowledge on the different types of UAT in this article.

To help you out, we've prepared a UAT testing template which you can download. This covers everything from preparing your team, the test environment, to test execution and evaluation. Feel free to grab a copy to help you out with your UAT testing efforts.

Is there a UAT tool?

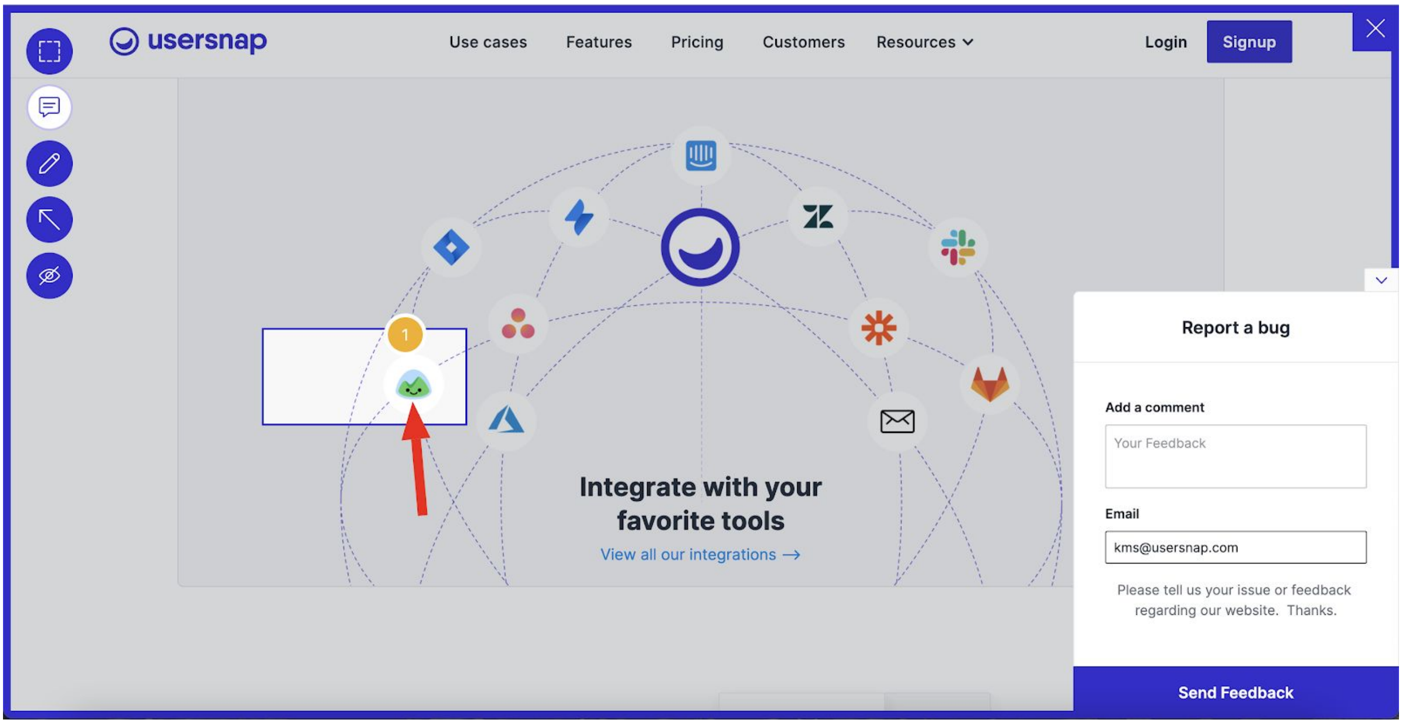
When considering User Acceptance Testing, you're probably not thinking of some "old fashioned" methods – like email, attachments, powerpoint presentations, spreadsheets, etc. You may consider compiling the client's feedback and testing results in your project management tools, such as Basecamp, Asana and Jira. But that's not ideal either.

A UAT tool should minimize the effort in giving feedback and include screenshots and video screen recordings. As it's for real-world testing, different kinds of devices and browsers environments would come up, so it will also be important to be able to automatically capture such metadata.

We don't want to hide from you that we favor this cool feedback solution for your best-of-breed User Acceptance Testing process. It's … drumroll … Usersnap

Used by more than 17,000 companies world-wide up until now.

Or check out our collection of the best UAT tools in this article!



Report a bug

Add a comment

Your Feedback

Email

kms@usersnap.com

Please tell us your issue or feedback regarding our website. Thanks.

Send Feedback

5 steps for your best-practise UAT workflow

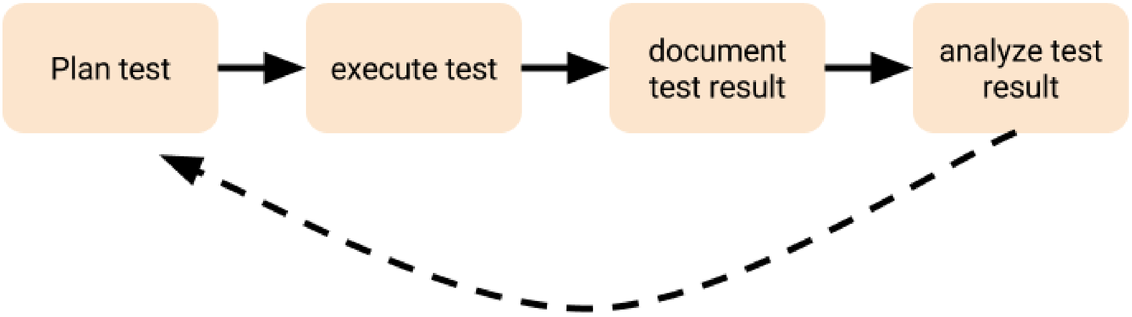
User Acceptance Testing is complicated. From creating a UAT plan, to executing your test cases and analyzing them. All kinds of different departments and people are involved.

And worst of all: UAT takes place at an awkward time: at the end of a project.

Your development team is done with their tasks (at least they should be) and your QA agents start testing the application. And they have just one main job: Testing if the application works for your users.

I'll guide you through the following 5 steps:

- Planning
- Execution
- Documentation
- Evaluation
- Reporting & Lessons Learned



Planning

Planning your User Acceptance Testing efforts is an absolute must-have. Without proper planning, UAT won't get you anywhere. Planning your User Acceptance Tests must cover the following areas:

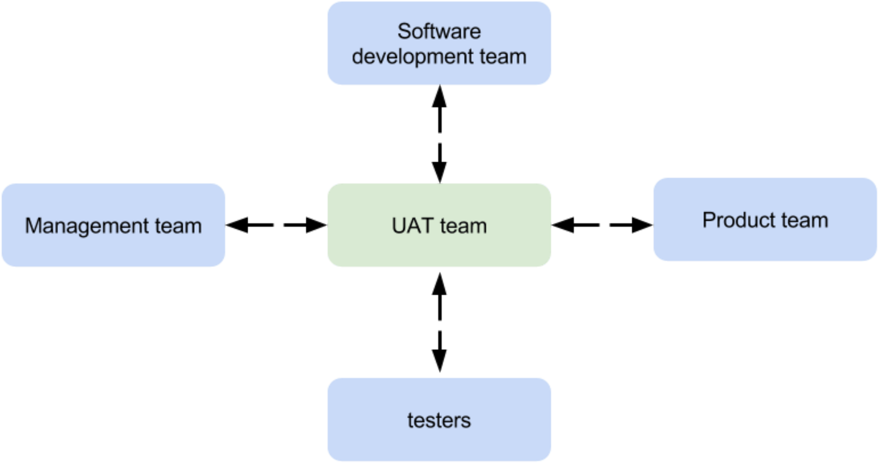
Planning & time management

In the first step, you need to clarify some basic questions. You need to collect information regarding your UAT schedule and your QA agents and testers.

Make sure to have all information in one place in order to set up a realistic UAT plan. And most importantly: Draft a concept of what your testing groups should look like.

Staff requirements

You need to plan who of your team is involved in the UAT yUAT execution as well as ensure that all responsibilities and tasks are known.



Every person involved in the UAT process should have a clear understanding of what his or her responsibility is.

When setting up your UAT team ensure to bring everyone on the same page. Establish clear communication guidelines and prepare your target audience for the test case.

Communication & Issue strategy

While executing the defined UAT test cases you need to make sure to have a workflow in place which deals with bugs, issues, and other problems.

UAT checklist

Before starting to execute test cases, I recommend making use of a User Acceptance Testing checklist. It will help you to stay focused and keep your efforts and to-dos in place.

Based on our UAT experience, we've put together this UAT workflow checklist template for you.

FREE DOWNLOAD

You can download and view the checklist for free.

Execution

You have a couple of options when executing your UAT test cases. As the name suggests your testers (= potential users) will now test and evaluate your application regarding certain test scenarios.

If you provide a global product on various geographical markets, chances are high that you can't meet these testers in person.

The test cases can be executed in one-on-one sessions via Skype or any other video calling software. During the sessions, you will gain a lot of quantitative and – even more important – qualitative data from your users.

Especially if the test cases are executed after the finished development process, you might end up with some new insights which you haven't taken into consideration. Your theory of how much know-how your users have will be tested.

Documentation

Executing and documenting your User Acceptance tests should happen at the same time. I'd recommend setting up a system which lets you document all relevant information without losing any data.

Documentation is one thing, clearly defined responsibilities for ensuring the implementation of your users' feedback is another key factor.

A lot of our customers have set up Usersnap for this step. Bugs, feedback, and other abnormalities can be documented on your user's browser screen. An easy-to-use project dashboard allows your colleagues to get a good overview of important issues. Making sure to delegate and assign priorities is another key thing here.

Evaluation

During this phase you need to evaluate if the defined criteria are tested and met. And most importantly if those criteria were successfully accomplished by your testers.

Has any test case failed? Which problems did occur? How can those problems be resolved and who is responsible for that?

The quantitative and qualitative data documented in the first place need to be analyzed. The following questions need to be answered:

- How many testers did complete the test cases?
- What was the overall rating of these test cases?
- What was the overall state of mind of each tester?

- Which emotions did occur during the test cases?

The evaluation phase is a pretty extensive one, since each and every single test case needs to be analyzed and put into context.

Reporting & lessons learned

You might think that the evaluation and reporting phase are the same. They are not.
During the evaluation phase you are collecting, aggregating and analyzing data. In contrast, the reporting phase deals with the bigger picture.

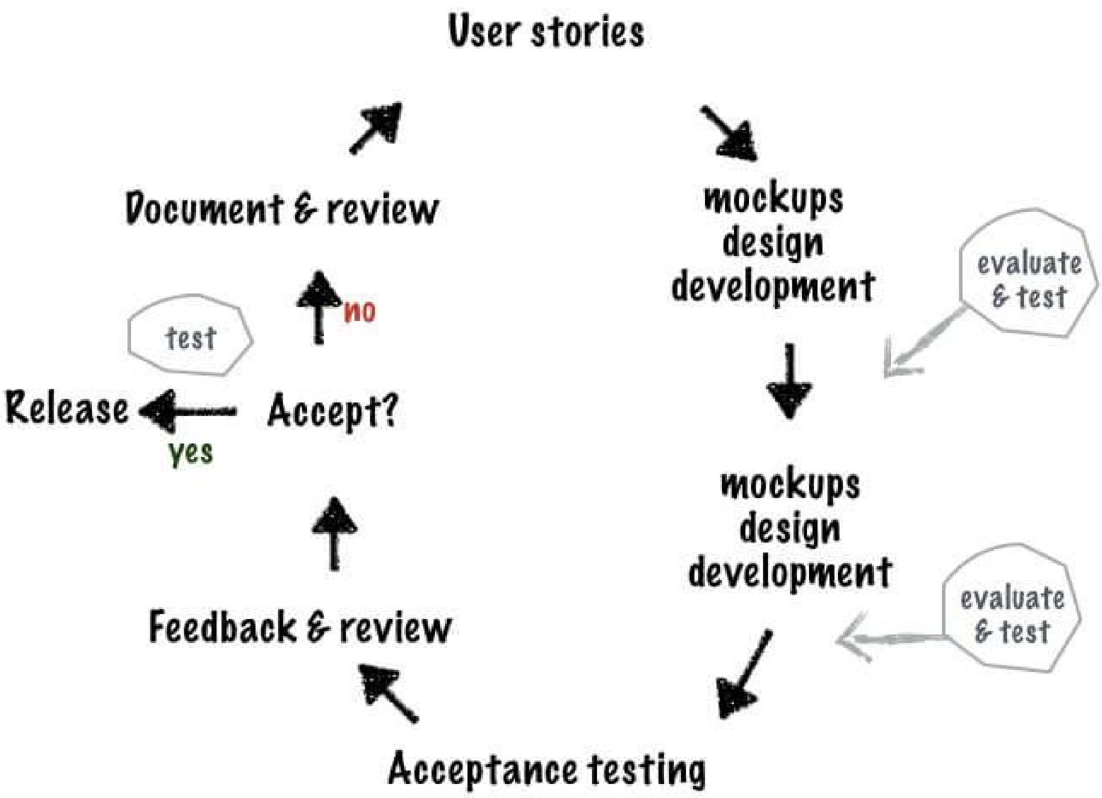
The main goal is to gather insights and lessons learned which will help you to improve your future test cases and UAT workflows. You might also start building relationships with some of your UAT testers during that phase as they are a helpful source for further feedback and insights.

UAT is not just a test. It's a cycle.

User Acceptance Tests are regularly conducted at the end of a software development phase, at a time when the product is nearly "finished". This must not be the case. The later problems occur, the more expensive they are.

UAT workflows can be implemented way earlier. When pursuing a more agile approach to UAT, UAT itself is not a separate project step. Instead, it requires a continuous collaboration between all team members and stakeholders.

Throughout the entire development workflow, UAT becomes an essential part of every project stage.



The goal here is to integrate User Acceptance Tests in each and every project step. UAT is not a phase anymore. Ensuring continuous progress means ensuring continuous test and feedback cycles.

usersnap

Bonus: Solution Tips for UAT with Usersnap

When selecting a solution to help make User Acceptance Testing and team collaboration easier, you can turn to Usersnap the feedback platform.

Add a simple button to your web project, let your testers just click, do a screenshot without any extra tools and send it with an annotation. Your team not only receives a screenshot which shows the issues in the test but you get all the technical detail information automatically attached.
Browser, operating system, screen size, URL, JavaScript errors. Making it easy for your engineers to reproduce and troubleshoot on the spot.

[Watch a demo about how Usersnap can help your team in User Acceptance Testing.]()

Or [sign up for a free 2-weeks trial and try Usersnap!]()