

White Paper

Six Tips for First-Class User Acceptance Testing

Providing you with:

- ✓ An understanding of the importance of UAT
- ✓ Insight into successful testing strategies
- ✓ Practical advice to optimise your own UAT activities



Introduction

Conducting robust tests that confirm your site or application performs as expected is an essential phase of any successful development project. However while your supplier can, and should, be conducting multiple tests as part of their own stringent Quality Assurance (QA) process, there is also expert domain knowledge of your specific industry, organisation and audience needed to ensure that critical issues, such as legal or accessibility requirements, have been met. As a result project output must also be reviewed by subject-matter specialists in a process known as User Acceptance Testing (UAT).

“ **User Acceptance Testing: the final stage in the development of a piece of software in which the customer is asked to use it to see if there are any problems that still need to be corrected before it can be accepted.**”

Cambridge Dictionary

This guide offers expert advice and handy hints to help you optimise every stage of UAT; enabling you to conduct this testing as effectively as possible to ensure your websites and applications deliver the highest levels of quality and performance.

What is User Acceptance Testing?

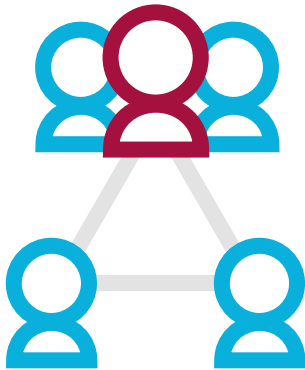
User Acceptance Testing is the period of testing that checks whether the solution meets the requirements of the user, by ensuring that real-world tasks can be performed without issue. UAT is usually conducted by the project client and forms a key phase of the sign-off process.

If UAT is not carried out properly then issues may not be discovered until after the project has gone live, which could result in a poor user experience, missed opportunities for conversion and even costly site

downtime. In some cases these issues may not be found until after the warranty period has expired, leading to organisations facing expensive fixes that could have been avoided had the problems been picked up earlier.



Get everyone involved early



It's vital that you include representatives from across your organisation at the outset of the project; getting buy-in on the value of User Acceptance Testing and commitment to performing testing activities at this early stage will help ensure that every area of your site or application is assessed and approved against a wide range of criteria by people highly experienced in their respective fields.

It's also important to put in place the structures and processes needed for regular communication between your internal teams, to prevent easily-fixed issues from snowballing into major problems later in the project lifecycle. For example, getting the individuals who are writing specification documents to talk to your legal team before the details are signed off will enable compliance requirements to be appropriately incorporated and addressed during development, rather than having these potentially project-blocking issues discovered in the final stages of testing when resolving them is likely to significantly delay go-live.

Finally, make sure you talk to your supplier's own testing team at the beginning of the project, so that they understand which features and requirements are most important to you and can focus their efforts appropriately. Remember too that priorities can change as a project progresses; scheduling in regular updates will ensure that all parties remain aware of the current status along with any actions required.

Remember

- Consider the skills and expertise you'll need in place before embarking on UAT

- Form a cross-functional team that will be responsible for managing and conducting testing activity
- Make it easy for questions and concerns to be raised through the appropriate channels

2

Understand your users



It may sound obvious, but the clue to User Acceptance Testing is in the name; you're confirming that your site or application is suitable for your users, whether they're internal team members or external prospects and customers. Consequently you must have a detailed understanding of who your users are, why they've come to you, what they're looking for, and the goals they want to achieve. This is easier in some cases than others; for example, if you're an online fashion retailer your users will likely be browsing for or purchasing clothes and accessories, whereas a utility provider may have visitors to their site who are instead focused on finding out more about the company, checking their accounts, paying a bill and more.

In this situation **personas** can be incredibly helpful. Creating a persona for each of your user types enables you to assign them lifelike character traits that help build a picture of how they may interact with your site; information that can be used to target the most valuable areas and journeys during the UAT phase.

“**Personas: imaginary versions of your prospects and customers based on historical and benchmark data.**”

Personas are also useful for identifying how users access your site - whether they're likely to be on the move and therefore using a mobile device, or prefer larger desktop screens for example - and this insight can be similarly applied to your UAT activity. Prioritise the devices and browsers you'll test on according to their popularity with your target audience, to cover as much ground as possible in the time available.

Remember

- Reference any existing user research you have available to better target your testing
- Be sure to cover the different audiences likely to use your site or application
- Consider not just who will be accessing your site but how, and include a range of devices, browsers, operating systems and contexts

3

Dedicate enough time to testing



Even if you've set your team up in advance and understand the most important elements to test in order to meet your users' needs, as with every other stage of the project lifecycle a strict timeframe will likely be set for the completion of UAT activities to enable delivery deadlines to be met. When higher-priority issues are fighting for your attention it can be difficult to find the resources required to carry out UAT thoroughly, but it's crucial that you set aside time to conduct a detailed assessment of your site or application, ideally in an environment that enables you to concentrate solely on the task at hand. If you're not distracted or feeling rushed you'll be better able to work your way methodically through the site, rather than resorting to a 'big picture' review that can lead to critical issues being missed, and potential delays to go-live should they be discovered hours before the final delivery is due.

Ideally, UAT should take place continually throughout the project to provide ample time for any issues raised to be resolved and re-tested, in case the fixes have created new issues as a consequence. This constant testing is particularly relevant if an **Agile** methodology is being employed on your project.

“ **Agile: a leading software development methodology, famed for its lean framework minimising risk and maximising flexibility, that provides a set of guiding principles based around regular customer feedback and responsiveness to change to ensure the features with the most value are delivered first for the greatest business gains.**”

No matter how frequently you're conducting testing however, in all cases it's recommended that a schedule be created in collaboration with your supplier that defines key testing milestones informed by major project deliveries. Not only will this improve your ability to schedule resources as required, but it will measure your activity against expected progress, helping keep the project on track.

Remember

- Set up an appropriate environment for productive testing; for example reserve a separate room or turn off your email access
- Work with your supplier to understand their delivery schedule and plan your activity accordingly
- Regularly review how your testing is progressing and take remedial action through the appropriate channels if required

4

Create a structured test plan



To ensure full testing coverage, your UAT activities should be supported by a structured test plan. Having this formal document in place facilitates increasingly focused testing and, when time may be limited, provides confidence that the business-critical areas of your site have been assessed and are working correctly.

An effective test plan should describe all the features and functionality you intend to cover during UAT, along with considerations such as the technology you'll use and how you'll approach testing. This high-level overview should also be supported by test scripts; specific instructions that will enable your testers to replicate and validate important user journeys to meet the requirements of your test plan.

Everything included in your test plan should, of course, be guided by the needs of your users; highlighting the importance of building the detailed understanding of your audience noted earlier. If you've incorporated user insights into your project from the earliest stages, the requirements defined in your specifications can act as an effective and easily-applicable guide.

You should also include clear acceptance criteria against which success can be measured; if **user stories** have been employed during the requirements gathering phase of your project then these can be leveraged to inform the criteria.

“**User stories: a pragmatic way to determine the ‘who?’, ‘what?’ and ‘why?’ of user and business requirements in an easy-to-read narrative format.**”

Taking the form “As [user], I want to [task], so that [motivation]”, user stories clearly signpost the outcomes that are desired from your tests, and provide a deeper insight into behaviours and expectations.

It's important to run your test plan past the team of stakeholders you should have gathered together at the start of your project (mentioned previously), to ensure that no crucial elements of your site are overlooked. It's also a good idea to involve your supplier in the creation of your plan, or have them review it once it's completed, as they will likely be able to offer expert advice on technical issues such as the most appropriate tools, approaches and reporting frameworks to use.

Remember

- Cover important conversion points and essential functionality in your test plan to guarantee their efficacy as a priority
- Use clear and easy-to-understand language, avoiding unnecessary technical terms and complexity
- Don't forget acceptance criteria; your testers need to know what they're looking for!



Don't forget exploratory testing



Structured testing is vital to confirm that the main features and functionality of your site are performing as expected, but to fully safeguard against potential bugs you should also conduct exploratory testing during the UAT phase of your project. In contrast to the type of testing covered by your test plan and carried out by your supplier, which focuses on walking through pre-defined a-to-b journeys, exploratory testing requires you to act naturally, as a user would, while attempting to complete common tasks that might include finding a piece of information or working out how to complete a transaction.

Placing greater emphasis on cognitive processes and problem-solving, this more informal approach to testing often uncovers issues that would not be found with the structured plan alone. However, because exploratory testing doesn't have a clear set of steps to follow it's vital to note down what you're doing as you do it; if you encounter an issue it may be due to something that occurred several steps previously or as a result of a unique combination of actions. As such, providing the information needed to replicate exactly what happened is vital to enable your supplier to isolate and resolve the underlying cause of the issue.

Remember

- Use a variety of testing approaches to increase coverage and ensure all issues are logged
- Apply your understanding of your users to anticipate how they might interact with your site or application

- For even more powerful results test with actual users; you can't predict every possible outcome and may be surprised at the decisions they make!
- Support your testing with detailed notes that record the pages you visit and the actions you take

6

Report issues correctly



A core stage of the UAT process is documenting your results, and doing this in a clear and concise manner can significantly streamline future development activities and help speed up your site or application's time to market. All issues must be reported with enough detail to enable the problem to be understood by anyone working on the project; for example, if a member of the development team leaves the organisation and the individual replacing them can't quickly access the information they need to begin work immediately then valuable time may be wasted.

The details of issues (or 'tickets') raised during UAT will likely depend on the requirements of your issue-tracking solution; however, it's good practice to include the following:

- Issue location: include the URL, page title, and area of the page as necessary
- Expected outcome: what was supposed to happen according to the initial specification?
- Actual outcome: provide a comprehensive description of what is happening, including screenshots if required
- Reproduction steps: detail the actions that generated the issue

It's also important to discuss and agree with your supplier at the outset of the project the different severity levels along with the response required for each, and correctly assign tickets according to these definitions. For example, if all issues labelled as 'critical' must be resolved before the project can go live then only issues that will prevent the site from functioning effectively should be labelled as such, to ensure these are looked at as a priority. If a number of issues are classed as critical when in fact they're only minor points, this may have a negative knock-on effect as to when your site or application can be released.

As well as severity levels you should also ensure that everyone involved on the project is aware of the terms used to describe the different kinds of issues that may be reported. Again, this will depend on the conventions used by your supplier and the tools you have in place, but it's most important that a shared understanding is created as to what is meant by each term, and how they should be applied. For example, if a 'defect' refers to something that doesn't match the specification while a 'change' indicates a general improvement or update, make sure you continually refer back to the specification to check that tickets are being classed correctly. Not only does this make it easier for your development team to know how the site should behave, it's also important to keep track of your UAT output; introducing too much change in the latter stages of a project introduces risk and should be avoided if possible, and assigning tickets incorrectly can mask the true extent of changes being requested.

Remember

- Familiarise yourself with the issue-tracking solution in use on your project before you start logging tickets
- Share the definitions of different severity levels and issue types with everyone involved in conducting UAT
- Get high-level summaries of the output from each testing session to maintain an overview of progress

Conclusion



A thorough, methodical and disciplined period of User Acceptance Testing is crucial for the production of high-quality software that effectively responds to the needs, requirements and expectations of your users, to ensure that your digital products and services are supporting your business objectives.

This guide has looked at some of the key considerations for organisations about to perform UAT on their own project, from the early strategic decisions that must be made with input from both internal and external stakeholders, through to executing professional testing activities and reporting the findings in a way that can be easily and instantly understood.

To learn more about the various types of testing that can be applied to web and software development projects visit our [Software Testing](#) page, and download our [Ultimate Website Launch Checklist](#) for a comprehensive overview of the additional elements you should be examining to ensure your site or application is launched successfully.

Glossary

Exploratory testing: testing activities where decisions are made as the process is taking place, imitating the likely actions of users

Go-live: the public release of a project when the site or application is made accessible to its end-users

Issue-tracking solution: collaborative software that enables issues to be raised, assigned, and tracked through to resolution

Quality Assurance: the process of testing undertaken throughout a project to ensure the highest levels of performance, security and stability in all deliverables

Severity level: an indication of the urgency of a ticket, determining the priority order in which they are addressed

Structured testing: testing activities that follow set paths to validate results against expected outcomes

Test plan: a centralised document prescribing the scope of the testing period

Test schedule: a timetable for the completion of the various stages of UAT to ensure deliveries can be made on time

Ticket: issues that have been raised in the issue-tracking solution; typically these are given a unique identifier for easy reference

Users: the individuals who will be interacting with your site or application; these may be employees, prospects, customers, partners and more depending on the purpose of your project



Simply brilliant thinkers making software brilliantly simple



**Strategy &
Planning**



**User Experience
& Design**



**Software
Consultancy**



**Software
Development**

At Box UK our team of simply brilliant thinkers, consultants and application developers mastermind simply brilliant solutions to the world's toughest web and software projects.

It's why blue chip organisations around the world, from airlines to car manufacturers, international banks to global charities, choose to trust us with the execution of their digital strategies, from the development of enterprise-level custom software platforms to the delivery of award-winning multi-channel experiences.

So if you want an effective, compelling software solution, talk to us.

info@boxuk.com • +44 (0)20 7439 1900