

Meta: It is best practice to understand why manual testing is important. This way, we can know which useful methods to select to ensure successful test coverage.

Why Manual Testing is Important

Manual testing techniques are those which require the physical involvement of human beings. Automated testing has a reputation for speed, efficiency, and economy, and its rise in popularity suggests to some that manual testing is falling out of favor. To disregard the value of manual testing would be a mistake, so this post explains why manual testing is important.

It is impossible for automated testing to replicate precisely how humans engage with products designed for their use. To compensate, software testing plans generally employ a range of test methods that involve both manual and automated testing, which enables all requirements to be addressed. Automation works best for routine and repetitive testing, which can be extremely time consuming and tedious if performed manually. Manual testing, however, works better in areas where automation cannot replicate the intuitiveness, overall comprehension, and iterative assessments that a person can bring to the process. In addition to this, some forms of testing can only be performed manually, and both the setting up of automation testing and assessing the results also needs human intervention.

11 Reasons Why Manual Testing is Still Important

Manual testing is the last line in QA because human intervention is needed for examining any flaws or problems encountered in automated testing. A combined approach is the best way to proceed, but let's now take a look at some reasons for the importance of manual testing.

1. Usability and UX Testing - By Humans for Humans

Usability/UX testing is the way testers discover if a website or other software behaves as expected by the designer when in the hands of a real-world user. [Usability/UX testing](#) is conducted by an observer who monitors invited user test subjects, as they work through a list of pre-designed tasks. The tester records how well they succeed, and if they struggle with any of the set tasks. They can also gather useful data such as if the path to a goal has more steps than is satisfactory for the user, and if navigational signposts are easy to identify and successfully selected. With usability/UX testing, the tester is free to run with their instincts to pick up on any unprescribed paths that a user might take. They can then discuss with their team any perceived merits in making changes to accommodate.

Because the method harnesses the unique attributes of people to bring useful information to light, fully automating usability/UX testing is not possible. However, tools such as eye-tracking software to monitor levels of visual interest across a website can be useful.

Distinct from the test professionals who run the sessions, it is preferable that usability/UX testing is not performed by anyone connected with the build. This is because the most valuable test data comes from users with no prior knowledge of the product who will be more likely to spot unexpected issues, and not be swayed by any unconscious bias.

2. Issues are Found in the Least Obvious Places

The success of manual testing can be defined by the thoroughness of the human intervention brought to a project. At the same time, automation's most valuable contribution is a robotic adherence to defined sequences that would be time-consuming and boringly repetitive for testers. It is the reliably non-thinking aspect of automated testing where improvisation is impossible, which makes it entirely unsuitable for use in ad hoc or exploratory testing.

As introduced in our companion post [Types of Manual Testing](#), ad hoc and exploratory testing invite a hands-on, non-scripted approach to finding bugs and usability issues. Testers are encouraged to follow their curiosity and initiative to work through often spontaneous paths of inquiries, and to explore areas not usually tested. When something problematic does crop up, human testers can quickly adapt their lines of inquiry. In automated testing, this is not possible because the script would need to stop, so parts could be rewritten before testing could resume. Sometimes plain common sense is required, such as when something may have seemed correct at the time of writing, but when coded, the feature could need some adjustment. Having a feel for when something is not right is the preserve of human beings, making yet another case for the continued importance of manual testing.

3. The Economy of Short Runs

For the types of testing suited to automated testing, such as regression, smoke, and sanity testing, set up can be expensive and time-consuming before any savings can be made in either time or cost. In the longer term, automated scripts offer many benefits to test teams by providing regular, repetitive testing, which contributes to a faster turn around and can aid the overall quality of the product. However, for short projects, the extra hours and processes needed to set up a sequence of automated testing, before testing the script itself, may negate any possible savings.

Once a test run has been automated and set to work, there will be a reduction in time needed for the development cycle, but if an interface, for instance, receives a thorough re-design, these changes can cause problems and numerous extra hours will be needed to update the automated tests. Also, if the changes are part of an ongoing process, meaning that the interface undergoes constant review with considerable refinements, it may not be practical to automate GUI testing until the build is more stable.

Once continual testing is underway, the workday should be easier for testers because automation alleviates the natural boredom of repetitive test runs, as well as reducing the need for continuous concentration. Sometimes, though, automated tests contain bugs themselves, which makes manual testing essential for finding out where the issues lie so the situation can be rectified.

Because of the necessary investment in time and money before automated testing can begin, there is little to be gained by going through this process for short test runs. Their brevity and relative simplicity will defeat any economic objectives.

4. Seeing the Broader Picture

In testing, nothing compares to a tester being able to distill a detailed overview of a product or project just by tapping into their training, their on-the-job nouse, and life experience. No automated setup can replicate this kind of manual testing that does not need test plans, and assessments rest on just a feeling that something's not right. Further structured testing will take place from here to locate the problem before flagging it up to the developers, but the initial impetus comes from a tester's unique and unquantifiable skills.

5. Automated Tests Can Contain Errors and Holes

Despite the positives offered by automated testing, automated scripts can only test what they have been written to test. This means there's always the possibility that the person writing the script will miss a potential issue that should be included. Without introducing human input at this point, the chances are that the potential issue won't be tested. Errors of omission are common mistakes, so it's vital to schedule a manual testing sweep to make checks as part of the setup procedure before testing can be approved, then started. Even then, if a script has inaccuracies, these might not be revealed until testing is well underway. Automated testing, when in progress, cannot deviate from the script, so any errors or holes, if left to run, could eventually deliver an inaccurate test report. Situations like these call for exploratory testing performed as only humans can.

6. Testing on Different Browsers and Devices

Cross browser testing plays an essential part in the full testing process. Testing for bugs in functionality is usually an automated process, so a cross browser testing tool is used, which repeatedly runs the same script on potentially vast numbers of browser, device, and version combinations. When testing the more physical, visual and tactile elements however, manual testing is the best option, because human testers have the attributes that allow for free-ranging initiative, and can explore using the senses of sight and touch. Occasionally, a decision will have to be made that involves compromise. A site may not look perfect on one particular browser, so a judgment call on what is acceptable will have to be made.

Economy of scale is a consideration in cross browser testing. For major household name product runs, careful checking across many of the combinations and more recent versions of hardware and software is absolutely essential. This makes function testing by automation a necessity, but for test runs at the other end of the size scale, manual testing offers a manageable alternative.

7. The Advantage of Variations in Manual Testing

Under normal circumstances, automated testing will not deviate from the test script path it is following, which is either excellent or a problem, depending on factors. When a person

physically tests something, even if they are following test cases, how they test will slightly vary each time due to human instincts and because they may have to process different inputs, and so on. This somewhat "rogue" approach can be a real advantage because a tendency towards randomness invites the possibility of testers stumbling across bugs in places previously unconsidered. Of course, this human talent for not being robotic can also lead to errors, but if any human-made testing errors do occur, it falls to manual testing to correct.

8. Testing Connectivity Issues

Testing internet connectivity is an area that demands human awareness and intervention due to the differing array of options on offer, such as cell phones, tablets, TVs, and integrated Smart homes, which makes manual testing the best and most reliable testing choice. Modern high-speed broadband, and up to 5G mobile internet, has enabled the creation of interconnecting devices that intervene in areas of our lives that would have been unimaginable even ten years ago. But all this impressive technology is only as good as the connectivity that powers it.

All connectable hardware uses web browsers, which will suffer from occasional drops in connection speed and have connectivity issues from time to time. When dropouts happen, a site or application needs to already have in place a form of fallback to protect the system from complete failure when connectivity is lost or reduced. Working in the same way as the familiar Safe Mode found on a computer, the significantly reduced functionality of a fallback will hold the system in a form of stasis until repair or recovery can be made. The fallback used by Google Documents prevents any new text or data being entered during an internet drop out. In this way, the person attempting to continue their typing knows that nothing entered, means that nothing has been lost during the connection drop. Variations in the types and application of fallback systems make automated testing almost impossible, which leaves manual testing as the only practical option.

9. Supporting End Users

Away from the test lab environment, a related and equally important strand of testing concerns the support offered to end-users once a product has been finalized and released to the market. Unfortunately, stray bugs can be discovered by customers who notice that their purchase malfunctions in some way, which disappoints their user experience. Because customer satisfaction is better managed when their problem is attended to by a human support agent, this situation is rarely the time for automation. Automation could risk customers becoming resentful - possibly permanently - if no timely solution is found or even an offer of a purchase refund.

Customer support agents commonly receive complaints of bugs. When a customer reports a malfunction, they want a swift solution to their problem. Testing needs to happen to find out if the problem is due to human error or misunderstanding, or if there is a bug preventing the item or interface from working properly. Manual [exploratory testing](#) to check the situation is usually the first step. If the problem is not one of human error, further exploratory testing, or

testing that follows pre-written scripts, will be needed. By following this process, the agent can offer a more satisfactory response to the person raising the complaint.

10. Accessibility Testing

Many people find it difficult or inconvenient to use devices or to access websites in the standard way for a number of reasons. These can range from needing a larger page size that's easily solved by pressing Ctrl +, to requiring a more complicated set up involving extra software and hardware to support people with specific physical disabilities.

Accessibility testing aims to draw attention to how specific improvements can be made to a website so as many people as possible can successfully use and enjoyably the site. Because improving accessibility will benefit many site visitors and is not just a tick box exercise, the best approach is to [embed the accessibility testing process](#) within the entire QA, rather than treat it as an afterthought just before release. Areas for consideration will regularly include improving visual accessibility for color blindness, and the availability of an efficient screen reader to support the hard of hearing.

Again, the human touch is best for this kind of testing because value judgments are required. Tools such as emulators can be useful for gathering information on how the site or application performs, and it can also be informative to gather input from a range of people with disabilities and age-related limitations.

11. Other Forms of Testing

Aside from the recognized [Types of Manual Testing](#) listed in our companion post, there will still be other tests to be performed that fall outside the set categories. For instance, periodically, we may want to check the page loading speed of a website, or that the site appears correctly in Google. These types of tests are unlikely to be scripted, but tools could be used, such as for finding out the page load speed.

Conclusion

A comprehensive testing project will include both manual and automated methods in the test plan, but the ratio depends on the type of product under test. As has now been explored, manual testing is the way human value judgments can be made on bugs, flaws, errors, and omissions, while automated testing can free up the tester from the repetitively boring aspects of software testing.