ICTSAS519 Perform systems tests – Part 3

Contents

3. Analyse and Classify Results

In previous resources we've discussed the factors that need to be considered when planning to test a website and looked at some testing techniques in more detail.

In this resource we'll examine what needs to be done once testing has been completed. We'll consider how to go about summarising the results from testing and determining the success or failure of the test. We'll also consider how these results will be used to determine the next phase of the project.

The topics contained within this resource are:

- Summarise and Classify Results
- Compare Results to Specifications
- Notify Operations of Test Completion
- Gain Required Signatures
- Schedule Meetings
- Ensure test reporting complies with documentation and reporting standards
- Summary

3.1 Analyse test results against acceptance criteria to identify variances

Testers should keep detailed notes about the tests they have performed and the outcomes of those tests. This ensures that the results are accurately documented and not left to the memory of individual testers. If you've video taped usability sessions, the problems encountered by the test subjects and any suggested outcomes should still be documented in writing.

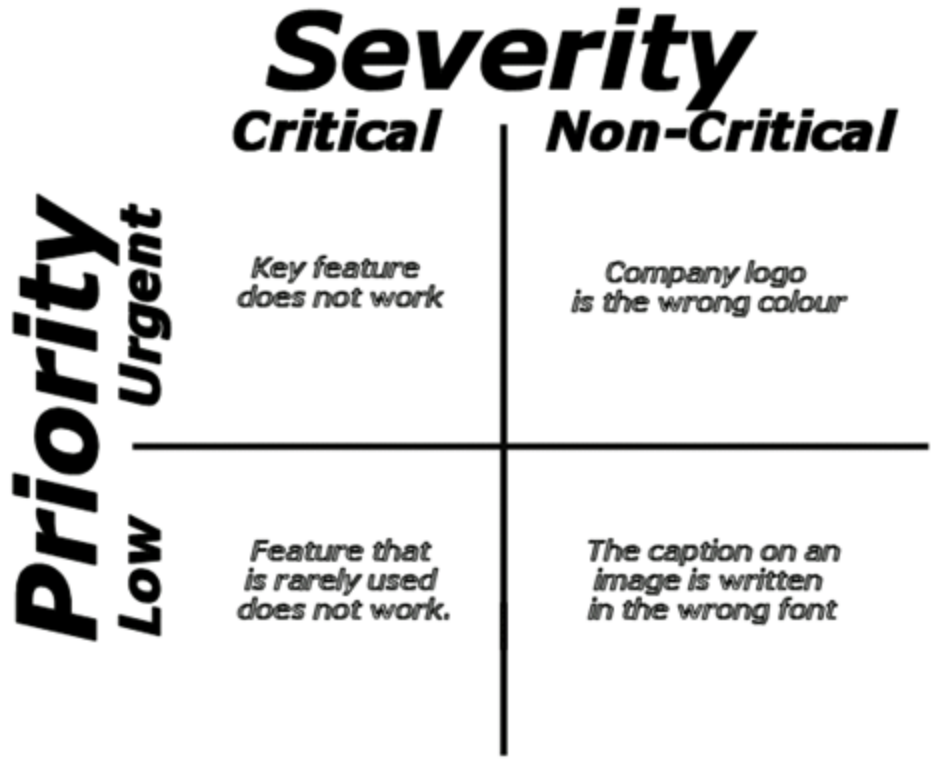
Once website testing is completed, your testers will have accumulated a large quantity of data, which we need to condense into a form that is more easily understood. The testing team should collaborate to produce a report explaining what tests were performed and their results. This should include all forms of testing conducted, whether it be usability tests, load tests, security code reviews or proofing of content.

This report should include the results of all benchmark tests conducted and any tests against acceptance criteria. Management will require these results when seeking a customer's sign off at the end of the project.

The report should present a summary of the tests that failed and classify those tests into different levels of severity and urgency. Tests that failed usually represent "action items" which will need to be followed up. These items may need to be fixed by the developers of the project and retested. If there are a significant number of issues that will need to be followed up, it can help to prioritise them to ensure that they can be dealt with in a timely manner.

A severity classification indicates how severe the bug is. This can be anything from purely visual items with no effect on functionality, up to entire modules or functions that simply do not work. A low severity item might be "the logo at the top of the home page is 2 pixels out of alignment". Things that rate a medium severity might include a name and address form that crashes when Irish names, such as "O'Halloran" are entered (many websites have problems with the ' in Irish surnames, but work perfectly otherwise). A bug rating a critical severity might include a credit card validation module which always reports a card number as invalid, thus preventing any sort of ordering from the site.

Classifying issues by urgency allows us to establish an order of priority for dealing with issues. Urgency often follows severity, critical bugs are often those which require the most urgent fixes, but this is not always the case. Priority is a somewhat subjective measure and may change over the life of an issue. A cosmetic issue may be low priority for a website which is to be launched in six months time, but considered urgent when it is discovered a week before launch. Severity on the other hand is a "hard" measure which does not change. The following diagram illustrates the relationship between Severity and Priority and provides examples of issues that might fall into each category.



One thing to consider when establishing priority is whether an issue is critical to the success of a website, or whether it can be left unfixed for a while. Those issues which are critical to the websites success will obviously need to be fixed prior to the website's launch and should be allocated a higher priority. The website might be launched with less critical bugs still unfixed and these can be allocated a lower priority.

For example, let's say we are developing an e-commerce website that goes online in July and a critical bug has been found in the module used for the Christmas sales. The Christmas Sales module may not be used for several months after the website's launch. While critical in severity this bug may receive a lower priority than non-critical issues with the website's appearance.

It's important that bugs be rated by both severity and urgency because this allows us to assess both the impact of a bug and the timeliness with which we require a fix. Both factors can then be weighed when considering which issues need to be tackled first.

3.2 Summarise and classify test results to prepare report highlighting critical and urgent variances

In some instances tests may have been designed to produce a result that looks like the test didn't work. For example, when testing an e-commerce site the tester may be instructed to try ordering a product using a specific credit card number. If the credit card number was invalid (and that may be deliberate) the tester might report that the order could not be completed because the credit card number was invalid.

When preparing the report outlining the results of the testing (as discussed in the previous section), it is important to refer back to the test plan and verify what the outcome of the test should have been. In the example above, the test plan would indicate that when presented with an invalid credit card number the system should prevent the order being placed. So although it looks like the test failed because we couldn't place the order, it actually yielded the desired result and should be considered a success.

Test results should also be compared to the original design specification for the website. We're checking to ensure that the functionality described in the specification is present in the website we're going to deliver. The test plan should have been written based on the specification, but it's a good idea to verify the results against the original document as well.

Pay particular attention when reviewing the original specification to any acceptance criteria that were specified. Acceptance criteria are the specific tests that the website must meet in order for the customer to accept the work. In our testing we need to explicitly test each acceptance criteria and demonstrate that the site meets or exceeds the required specifications.

Typically acceptance criteria may include usage of logos, fonts or colours in a particular way so that a consistent corporate image is presented. They may include demonstrating that the site can deal with certain load conditions. Acceptance criterion will almost certainly include ensuring that certain functions and features are present and work correctly.

When dealing with clients, management will need to be able to clearly demonstrate that all acceptance criteria have been met, so it is useful to highlight this in your testing report. However, it is impossible to identify every possible condition or circumstance that may affect a website and test every single one. When reviewing the original specification it is important to identify any areas of the specification that have not been tested. Functionality may have been deliberately omitted from testing because it provides little risk of failure to the overall system and it is a good idea to highlight these areas. Alternatively functions simply may not have been

considered for testing in the test plan and these may require further testing. Where functionality has not been tested, it's important to establish why this was the case and detail this in the testing report.

3.3 Notify contact in operations of test completion to communicate implications

Before we began testing we discussed the importance of notifying your network administrators or network operations people that testing was to take place. This allowed them to take appropriate steps to deal with any problems that might arise. If you were conducting security penetration testing, network administrators might have had to disable their Network Intrusion Detection Systems to allow your tests to run. They may also disable or ignore certain warnings generated from network monitoring software while the tests were taking place.

It's equally important that you notify operations once testing has been completed. This allows them to activate any systems they might have disabled during testing, or undo any temporary configuration changes.

If your test environment used hardware that was on loan for the test, operations will need to know they can reclaim the hardware and use it elsewhere. You may want to consider making disk images of your test machines using a product such as Symantec Ghost (http://www.symantec.com/ghost) before the hardware is reallocated. This allows you to take a snapshot of the test environment and easily recreate it again if required. It also ensures that subsequent test sessions are performed using an identical configuration to the original test session.

Different organisations will have different procedures for contacting the network administrators. You will need to refer to your organisation's policy in order to determine the correct procedure for your organisation.

3.4 Obtain and incorporate feedback from superior on test results report to finalise report

Testing Notes

As discussed earlier, it is important that all members of the test team record their results and comments during testing. Otherwise potentially important test results may be lost or forgotten.

Some organisations may have specialised forms for this, or the test plan may require that certain pieces of information be recorded. If formalised procedures are in place it is important that you make yourself aware of them and ensure they are followed.

If an organisation has no formalised procedures for recording test results, it is important that each individual tester keep accurate details and complete notes of the testing they have performed and the results obtained.

Each tester will have their own preference on how they take their notes, whether it is directly into a word processor, in a notebook, or scribbled directly onto the test plan. Unless there are formalised procedures in place, it's probably best to allow each tester to take their notes in whatever form they are most comfortable with. These notes will be an important source of information when compiling a report on the testing once it is complete.

Non-Disclosure Agreements

Sometimes the existence or nature of a project needs to be kept under wraps until it is finally launched. Sometimes this is for competitive reasons, to prevent a competitor from using your ideas and bringing a product to market before you. Sometimes the project may be a proprietary system, which is to be used within a company that wants to keep details of their internal systems confidential. In these situations it's important that everyone involved in testing understand the degree of confidentiality required.

In situations like the examples used above, you may also wish to have all testers sign a non-disclosure agreement (NDA) that prevents them from talking to outsiders about the project. Where confidentiality issues are a concern, it's important that you have each of your testers sign an NDA. You should allow the tester to keep a copy of the agreement and retain a signed copy for your own records.

3.5 Schedule Meetings

The end result of the testing process described to date should be a test report that identifies successful tests, issues that need to be resolved and areas requiring further testing. A review meeting should be organised with all of the key stakeholders in the project to discuss the results of testing. This will most likely include representatives from the development team, testing team and management. It may also include client representatives and sales and marketing people.

A chairperson should be appointed to ensure the meeting stays on topic and minutes should be kept. The minutes will form an important record of who accepted responsibility to follow up each issue.

Client representation is particularly important if you are practising development methodologies, such as Extreme Programming (XP), which emphasise constant feedback between the development team and clients. In any case, including client representatives can give them a strong sense of involvement in the project. Clients may also be more receptive to changes if they feel that they've had an opportunity to contribute their own views.

The purpose of the meeting should be to review the test team's severity and urgency classifications for outstanding issues and adjust where required. Clients may demand the site be ready to go live on a particular date, or be more concerned about less critical visual aspects of the site. Their representation can be beneficial here, as their priorities are often different to those of the development team.

The meeting should also try to formulate a plan and timeline for fixing the outstanding issues and retesting if necessary. Areas that have not been tested should be reviewed by everyone at the meeting and referred back to the test team if the group feels they require testing.

After the meeting, the test team should formulate a plan for any additional testing required and retesting any outstanding issues, which are to be fixed by the development team. This should include consideration of all aspects covered during this course, including test environment required, types of testing, test scenarios, etc.

Once issues are resolved and retested, further meetings may be required in order to properly determine the status of the project; whether it's "finished", further work required, etc.

If early usability testing (using paper models) has been successful, approval may be given, either by management or the customer to proceed with development of the website itself.

If the consensus of the meeting is that the website is "finished" and no significant issues remain which need to be addressed before the website goes into production, management will need to be informed. They can begin the process of seeking the customer's approval for the website and achieving a formal sign off and hand over of the finished product.

3.6 Ensure test reporting complies with documentation and reporting standards

Conformance testing or **type testing** is testing to determine whether a product or system complies with the requirements of a specification, contract or regulation. It is often physical testing but may involve chemical testing or requirements for efficiency or interoperability. To aid in this, many test procedures and test setups have been developed, either by the standard's maintainers or external organizations, specifically for testing conformance to standards. Conformance testing is sometimes performed by external organizations, which is sometimes the standards body itself, to give greater assurance of compliance. Products tested in such a manner are then advertised as being certified by that external organization as complying with the technical standard. Service providers, equipment manufacturers, and equipment suppliers rely on this data to ensure Quality of Service (QoS) through this conformance process.

During the development life cycle of a web application many things need to be tested, but what does testing actually mean? The Merriam-Webster Dictionary describes testing as:

- To put to test or proof.
- To undergo a test.
- To be assigned a standing or evaluation based on tests.

Read Testing Guide Article

For the purposes of this document testing is a process of comparing the state of a system or application against a set of criteria. In the security industry people frequently test against a set of mental criteria that are neither well defined nor complete. As a result of this, many outsiders regard security testing as a black art. The aim of this document is to change that perception and to make it easier for people without in-depth security knowledge to make a difference in testing.

Sample Test Report

3.7 Summary

In this module we're primarily focused on the output from the testing process. We've considered some of the things to look at when compiling a

report on the testing process. We've also examined the role of a review meeting in determining what happens after testing.

We've now examined the testing process from start to finish. We've considered the issues when planning to test a website and looked in some detail at some of the tools available to assist with website tests. Finally we've considered how to document the test results and ensure that all key stakeholders in the project are aware of those results.

Like the web itself, the field of website testing is constantly evolving. New techniques and concepts spring into life, gain popularity and fall out of favour on a regular basis. Forums such as SitePoint (http://www.sitepoint.com/forums), QA Forums (http://www.qaforums.com) and DevShed (http://forums.devshed.com) can be an invaluable source of new ideas and techniques. Jokob Nielson's site (http://www.useit.com) is also an excellent source of usability advice.

Further Reading

Symantec Ghost	http://www.symantec.com/ghost
SitePoint	http://www.sitepoint.com/forums
QA Forums	http://www.qaforums.com
DevShed	http://forums.devshed.com
Jokob Nielson	http://www.useit.com