



MCIT Testing Plan Guide

The purpose of this guide is to describe how website testing is done at MCIT.

HTML VALIDATION

HTML code can contain syntax errors, logic errors, nesting errors and mismatched code according to declared document types. The W3C provides an excellent validator that provides specific feedback for errors.

<http://validator.w3.org/>

ACCESSIBILITY

We test our website for accessibility by checking the site using the following tool:

<http://wave.webaim.org/>

BROWSER TESTING

We test our website for functionality in the following browsers:

- Firefox
- Internet Explorer versions 8-10
- Chrome
- Safari

USE CASES

For our website development, “use cases” represent the various interactions website visitors will have with our site. For a basic information site, the main test cases would include:

- navigation
- link functionality
- information retrieval

TESTING TIPS (FROM WEBDEPEND)

REQUIREMENTS GATHERING

- Ask questions about what the website, web app or mobile app is designed to do and keep asking until you fully understand.
- Request documentation, wireframes and designs, anything that is up to date and will help you gain further understanding. You can't fully test something if you don't know how it is supposed to work.

TEST PLANNING

- Get organized before you start testing, a little bit of planning helps massively as you progress with testing.
- Really think about all the possible aspects that could be tested and prioritize the most important. For example, testing a site search can break down into many different items.

WEBSITE TESTING

- Don't try and test everything at once, test the website in several passes, for example, functionality in first pass, layout and styling in second pass, proof reading all copy in third pass, etc. This allows you to focus better and not get overwhelmed.
- When raising issues, be as clear as you can and attach a screenshot so that the developer can easily recreate the issue and understand what the problem is. The longer a developer spends trying to work out what the problem is, the less time they have for fixing it.

BROWSER TESTING TIPS

- Start testing with a browser that is fairly solid, such as Chrome or Firefox. This allows you to test the functionality and what the website should look like without many weird browser related issues such as may be found using IE8.
- Then move onto IE8, IE9, etc. and compare with the browser you just tested on. This should highlight any differences between the browsers in terms of layout and styling related problems.

GENERAL TESTING ADVICE

- Keep learning about how websites work, about HTML and CSS and design principles. You don't need to be a programmer but building a foundation of knowledge can really help to understand what you are testing in the future. I've been around websites for over 15 years and still learning something new most days.
- Be inquisitive and talk to developers about programming and building websites. Picking up on the correct terminology will help you to make yourself understood (and gain further respect) with developers.
- Similarly, talk to designers about design principles and what makes a well designed website. Learn what to look out for when testing so you can pick up on more design and styling issues.
- As you build your own knowledge and expertise you can impart your viewpoint onto the developers and designers to forge stronger relationships. These relationships are worthwhile, as developers will go that bit further to hunt down an issue that they may be struggling to recreate.