

Functionality

How one dull word can boost your revenue by
£10,000 ... £260,000 ... £12,000,000 ... or even more



~~Endless~~
Gain

Contents

Why functionality testing really matters	2
What should you test?	6
Why manual testing is best	10
Would you make a good tester?	12
What you need to do	14
What tools do you need?	19
Accessibility testing	23
Other tests	26
What happens next?	29

Why functionality testing really matters

**We're sorry, "functionality"
is a rather dull word.**

But it's incredibly important to your business. To explain why, here's a question:

**When did you last feel like
screaming at your computer?**

**We'll wager it was because
something didn't work.**

And that is what functionality is about. Making things work the way they should, so you provide the service or products you should — and make the money you should.

Yet so few websites *do* work the way they should. The vast majority fall short in one way or another. Which leads to another question:

Are you sure your website works the way it should?

Is it fully functional? There is only one way to find out. Test everything.

You may prefer to do the job yourself. Or you may entrust us with the task, but either way this booklet will save you a lot of time, expense and frustration.

It reveals how and why you should test everything to make sure it works (functions) as it should.

To begin with, let us explain why this is so important – and why ignoring it will almost certainly cost you dearly.

Quite simply, the odds are money is leaking out of your business right now while you are reading this.

How much money are we talking about? More than you may imagine!

Two client examples

Let's suppose you see something you like on a website. You're ready to buy. But the process is so complex you just give up — or the system simply doesn't work.

It's more common than you might imagine. And the benefits of fixing the problem are huge.

Company A

People using Internet Explorer 8 and 9 (IE8 and IE9) simply could not check-out with the purchases they wanted to make. The checkout was not working.

Once that problem was fixed, and they could complete their shopping the company gained extra revenue of £18,000 to £22,000 a month.

So by simply fixing the bug that broke the functionality of the checkout in those browsers, the company added up to £260,000 a year in potential revenue.

Company B

Company B's website couldn't be used by visitors using tablets. Fixing the problem brought a potential revenue boost of £1 million a month.

In both cases, the bugs causing the problems were deep in the websites and had gone unnoticed for a long time.

The plain fact is that countless companies have websites that look fantastic — but fall short when it comes to functionality.

Chances are your own website is one. But you'll never know how much it's costing you — until you find the problems and fix them.

Before you worry about matters like conversion optimisation you must be 100% sure everything works perfectly.

You can't afford anything that makes life harder for your visitors and customers, who are the lifeblood of your business.

And that starts with eliminating bugs.

“What is a bug? It's anything that hinders or harms the user experience.”

Abi Hough, Device Experience Director, Endless Gain.

Know what you are looking for

What should you test?

+ Functionality

Does your website work perfectly, no matter what devices and/or web browsers your visitors use? Can they add products to their shopping baskets or delete them? Does the checkout add up correctly when they buy multiple items? Is VAT applied correctly? Is the contact form working? Is online chat working? *Does everything work?*

+ Accessibility

Can everyone access your content, be they blind, deaf, using a mobile device in bright sunlight, in a noisy environment — no matter what?

+ Usability

How easily can your visitors and customers use your website?

+ Conversion opportunities

Are there any areas of your website that work but could work better to encourage people to buy?

+ Exceptions

Is anything on the website a major conversion blocker? For example, the website doesn't function when using a range of browsers.

+ General improvements

Could anything on the website be done better? Things like correcting spelling mistakes, grammar, improving the resolution of images, etc.

+ Performance

How fast does your website load using different devices, browsers, network technology, etc.? What happens if your website is experiencing high traffic? Does it crash or is it resilient? Are images caching so the

website loads up smoothly and quickly? How much will it cost mobile visitors to use the website using a mobile connection?

+ **Multi device/browser**

Your website should function on anything and everything out there, be it desktop, tablet, smartphone and with all the various web browsers, new or old.

Many kinds of bugs can cause problems for your users. And to complicate matters those problems can cause trouble in two ways. They can affect a particular device, browser or version of a browser. Or they can affect different devices and/or browsers.

And, because many ecommerce businesses use ready-made “out of the box” solutions to manage their online shops, they can spread like wildfire. If there is a glitch in one solution, it’s probably replicated time after time wherever that solution is used.

You cannot hope to get genuinely responsive website visitors who buy more from you unless you deal with these threats.

Important! What to check before performing multi-device/browser testing

+ **Do get the hard facts on the devices and browsers your visitors are using to browse**

The best way to do this is to look at your analytics. Providing that your analytics platform is set up correctly, it will give you in-depth information that allows you to target the devices and browsers your site needs to work correctly and thus how to focus your testing.

+ **Don’t rely on the general consensus and opinions of those around you**

It is a common misconception that sites don’t need to support older browsers, but believe us, this may not be so at all. So if your website doesn’t work on one of them but your users are visiting your site with it, you will be losing significant revenue.

What your functionality tests reveal should be reasonably easy to fix. Your in-house or contract website developers will be able to do the work, so there’s no need to ask for additional budgets, go through corporate sign off, etc., because you’re fixing what’s already broken and it’s already live. In reality fixing bugs can be a cheap, low-effort job for your developers, and it pays back relatively rapidly.

Additionally, it gives you a clean(er) baseline for further development work.

Man vs. machine

Why manual testing is best

You can do some testing using various automated systems, but to experience real life user functionality nothing beats doing it manually.

Manual testing gives a true impression of what your visitors go through. A computerised system has no emotion attached to the problems it finds. It won't be frustrated by a link that goes nowhere — it will simply tell you it's broken.

But that only gives you a partial view of the problem — the factual view.

With manual testing, though, you experience everything your visitors are likely to. And that helps you prioritise the bug fixes you'll need. The bugs that frustrate you most, like being unable to complete a checkout, or find a suitable result from a product search, etc., will be at the top of the list and will be quick revenue wins once fixed.

So being frustrated by your website is good. It helps you fix high-priority bugs — and thus make more money. Conversely automated software testing tools only give you a quick, limited overview of the problems. You don't have the advantage of appreciating how they infuriate the user.

And manual testing is better for agile development because you aren't re-scripting automated software testing routines every time your developers make changes on your website. You simply jump in and do what's needed manually, which saves a lot of time.

A word of caution

Don't ask your website developers to comprehensively test for functionality bugs. They are too close to the website. They're prone to only test in their own environments based on how they assume it's supposed to work.

Have you got what it takes?

Would you make a good tester?

+ Do you get bored easily?

Testing is repetitive and can be mind-numbing.

+ Are you observant?

You need to see everything.

+ Are you obsessed with detail?

You need to dig beyond high level, superficial findings.

+ Are you inquisitive?

You must be motivated to experiment with the various functions to find out how they work.

+ Are you consistent?

You must apply yourself methodically.

+ Can you take on the personalities of different visitors and use the website as they would?

You will often need to act out a range of customer personas on your website.

+ Do you like finding mistakes?

You need that craving to identify problems.

+ Are you easily offended?

You need a thick skin. Your developers and/or IT people will defend themselves strongly against your findings.

Get started

What you need to do

+ Check your analytics set up

Before starting comprehensive functionality testing you must set up your analytics platform, whether you use Google Analytics or not. And the set up must be meticulous — or you'll waste a lot of time fishing in a big sea of problems in the hope of finding the one you're looking for.

You must track and tag the things you want to monitor closely, such as all the steps in the checkout process. So the analytics set up must be thorough.

+ Let it run

Once you're sure your analytics will deliver the data you need, let things run for enough time to give you usable data.

When you come to review the data, look for the browsers and devices that are converting versus those that aren't. Also look at the various screen resolutions visitors are using and link them together.

For example, are IE8 users on 1024 × 768 screens having difficulty or is it IE9, IE10, Chrome, etc.? Unless you can segment the data and identify and link the fine detail you could be testing forever.

+ Prioritise

Once you have a detailed breakdown from your analytics platform, you can drill down to specific problems to see what will give bigger wins. When you've listed those, you can prioritise them so as to fix those that will give the most return first. Things that block the checkout clearly matter more than something that may cause a minor irritation on your FAQ webpage; you can look at that later.

+ Initiate a code freeze

After you have your prioritised list, initiate a code freeze on your website. This is essential, for without it your testing will be pointless. You'll be running tests on a changing website, making it virtually impossible to link problems to what causes them.

While you're testing functionality check if any conversion optimisation experiments are running. For example, if there's any A/B testing underway, you must know exactly which variant you need to test.

Make sure your IP address is excluded from the analytics or it could skew the results — particularly if you're load testing the website performance. If you suddenly have 100,000 connections hitting the site from IE9 in an hour, the data will confuse you.

Unfamiliarity breeds great testing

Have one person test all device and browser options. If they find something that seems unusual and struggle to make sense of it, ask another person to look at it. If it is a real problem, their added feedback will give more insights. Or if it really is a "user error", you won't waste time chasing something that really isn't a problem.

Oscar Wilde reportedly said, "My first impressions of people are invariably right". That is how you must approach website testing. Being unfamiliar with the website makes you more like a real visitor. You have no expectations, you haven't figured out any workarounds, you don't really know what you're supposed to be doing there or how to do it.

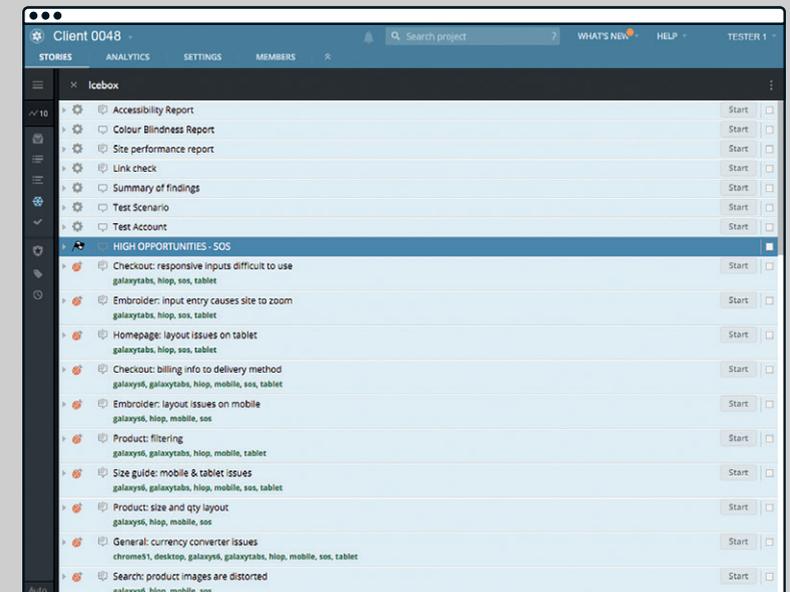
This lack of familiarity is, therefore, important; you'll likely find 200 or more problems when you first see a website, and the more you revisit it the fewer new ones will come to light.

Now you can try specific tasks and run with your first instincts for performing them.

You'll soon discover how easy or difficult it is to do them. That may leave you feeling happy or frustrated; it was easy to buy that product, or I don't think I can trust this website as all I get is an error message each time I try to checkout.

+ Log your findings

Put all your findings in your functionality testing report. Use a project management system for your reports as it makes it easy to compile your findings. This also allows you to assign fixes to the developers and close down each job as it's completed.



Pivotal Tracker, a useful project management system

+ Begin with the oldest browser

Your analytics will identify the browsers your website visitors use. It's always good to start with the oldest browser because it will highlight most of the problems. So, for example, IE8 may show 200 issues, IE9 may share 50 of those issues and IE10 15 or so. As you work up the browser list, you'll likely find fewer and fewer bugs to report.

Note the bugs in your report, recording the issues as accurately as possible with detailed descriptions, URLs, screenshots, and videos. Videos are particularly useful as they overcome the need for lengthy explanations regarding the browsers, devices, screen resolutions and operating systems used.

+ Check the detail

Now you have done those simple tests, start getting down to more granular details. For example, thoroughly testing the website form functionality.

You might look at whether there are character limits on fields and what happens when they're exceeded. What happens when you try to fill out a telephone number while using a mobile device (does a numeric keypad display automatically or do you have to use a standard QWERTY touch keyboard)? What happens if you try to input letters into a form field that should only accept numbers? Does the search work with keywords and return expected results? and so on.

The initial testing is the cake and these additional tests are the icing.

Tools of the trade

What tools do you need?

You cannot avoid using real devices because you must emulate the experience of website visitors. Their devices will be identified by your analytics.

This is particularly relevant with tablet and smartphone users.

Using device emulating software won't give you the insight you need, because a finger jabbing away at a tablet screen isn't the same as moving a mouse and clicking.

Use physical devices to give you a real world feel for what your website visitors are going through.

The need for real devices can be a problem unless you have your own collection, but you can rent them. So there's no excuse for not using the same sort of technology your visitors use.

Having a real device in your hand also lets you test network coverage across WiFi, 3G, GPRS and even old EDGE networks. The benefit is that you can ascertain whether your website will load and whether your visitors' data usage is costly/excessive. And of course that is also something you can remedy.

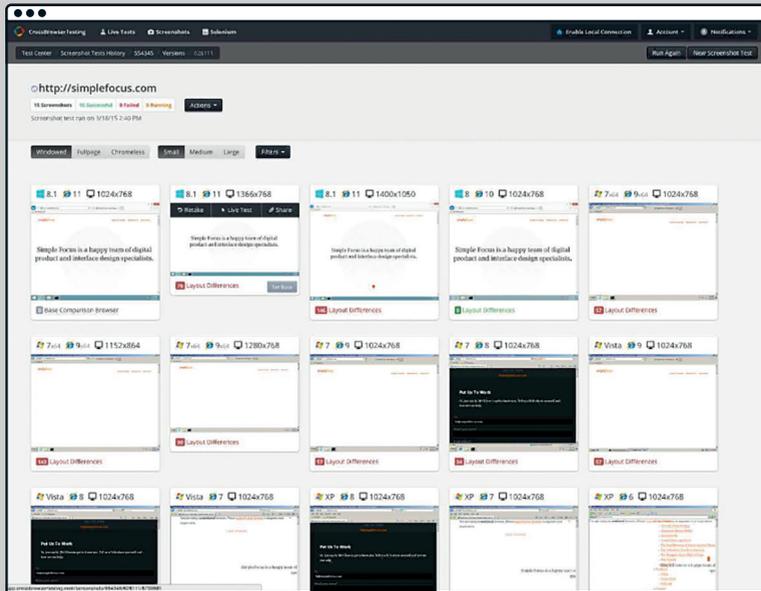


Endless Gain's device testing package

From left to right

- | | | |
|-------------------------|--------------------------|----------------------------|
| 1. Samsung Galaxy Tab S | 5. Apple iPhone 5s | 9. Samsung Galaxy S3 mini |
| 2. Apple iPad Mini 2 | 6. Apple iPhone 5 | 10. Samsung Galaxy S3 |
| 3. Apple iPad Air 2 | 7. Apple iPhone 6s | 11. Sony Xperia Z3 Compact |
| 4. Apple iPhone 4s | 8. Apple iPad (original) | |

Sandbox



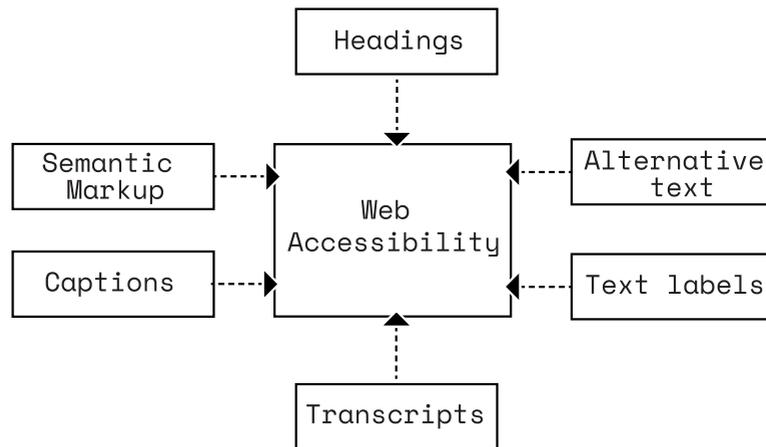
Example sandbox test environment for different browsers and resolutions

If your analytics throws up unusual devices that are no longer available, you can emulate them using a special testing environment called a sandbox. Some of these are online systems and some are desktop apps. They are relatively easy to use and convenient.

If a sandbox throws up something unusual about a certain device and/or browser, try to repeat it manually on a physical device as they can be inaccurate at times.

Don't turn less able customers away

Accessibility testing



Accessibility can sometimes be overlooked in testing. But you must look at it for many reasons which are sometimes legally required — besides limiting digital discrimination against those with disabilities who may use assistive technologies to browse your online content.

You can use a number of automated accessibility testing platforms which give you a synopsis of problems that may need addressing. Invariably though, these tools need a specialist accessibility auditor to interpret results and differentiate between actual and potential problems.

Accessibility testing reviews your website against a set of guidelines known as WCAG 2.0. This outlines recommendations, techniques and considerations for making web content more accessible.

Three levels of compliance are available: A, AA and AAA. Level A is the minimum compliance level; Level AA is the recommended level.

Online checkers will help you to identify accessibility issues.

Accessibility testing and ensuring your site complies to at least WCAG 2.0 Level AA guarantees that many of the techniques used to create accessible content benefit all your users. This is called the “Curb Cut Effect”. Curb cuts/dropped kerbs were first introduced to help wheelchair users, but they’re used by nearly everyone.

By their nature curb cuts make your web content easier for users to understand and interact with. Accessibility and usability have a symbiotic relationship. The easier your site is to use the more conversions you obtain.

What do your customers see that you don't?

Other tests

You can video user testing sessions for replay, showing heat maps, gaze plots and conversion paths using special software/platforms. The information they collect adds to the wealth of insights you can gather about users and functionality, particularly if they reveal problems your analytics have missed.

Such tests help you understand how usable your website is with no personal bias. You get direct feedback of their experience (usually with commentary). From experience of running many of these tests, the results often shock clients, and their designers and developers.

The most common question from most clients after viewing a user testing session is, "Why are they using it like that?".

The answer is simple. Every person is different and will interpret an interface differently. What's more, that interpretation can change depending on the mood or circumstances of the user. So if a user is in a rush they will likely look for shortcuts. Another user with more time on their hands may take a more convoluted route.

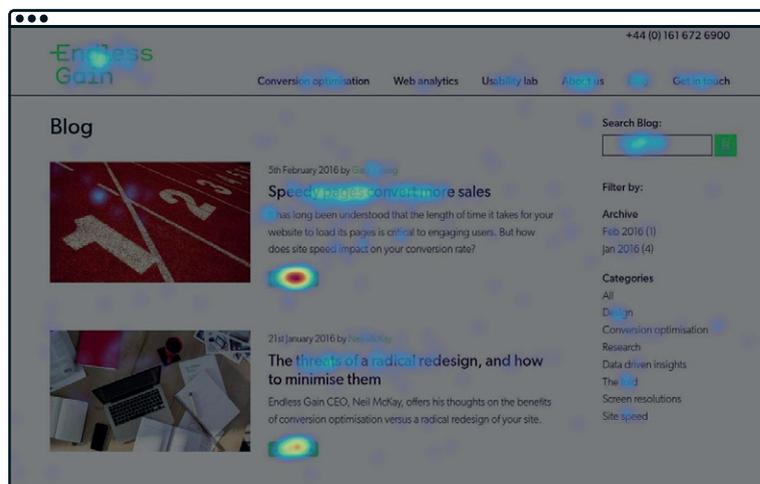


In-house user testing at Endless Gain

Gaze maps track a user's eye movements to see where they are looking on the page.



Heat maps show what users are clicking on. This helps if you do not have events set up in analytics as it will at least give you an approximation of what is happening.

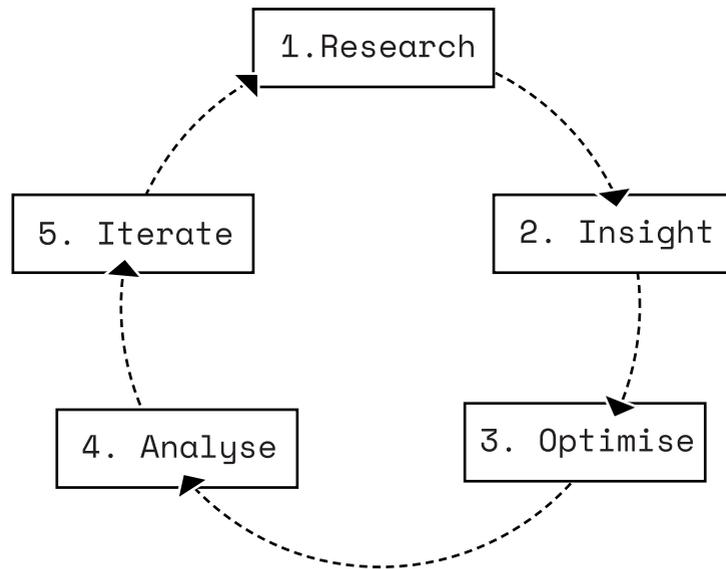


What happens next?

So, you've done your functionality tests. What next?

Well, it doesn't stop there.

As you do conversion research on your website and come up with variants for A/B testing, implement new solutions and so on, you will continually create problems that need fixing. So you go back to the beginning.



You'll soon adopt a new mantra of prioritising bugs, fixing them, testing the fixes, putting them live, testing them again in the wild, checking whether they've caused other problems, and then testing those — and so the cycle goes.

You never really finish because websites change and new bugs arise. But now you are prepared to deal with them. You're ensuring that you are always doing your best to avoid leaving money on the table. Money your customers want to give you, but because of your website bugs, you are not getting.

And as you saw in our two examples at the start, you are talking about serious money — that functionality testing will find for you.

If finding up to £260,000 a year in potential revenue, or a potential revenue boost of £1 million a month isn't enough for you ...

... let me give you one, final example of why functionality testing is so important.

Our device experience director, Abi Hough, has used it to find **over £100m worth of lost revenue for clients in the last four years.**

Now you know how she did it.

I hope this short booklet helps you find and fix any functionality issues with your website. And, of course, brings you greater profits.

Neil McKay
CEO, Endless Gain
neil@endlessgain.com

Endless Gain will help you convert more of your website's visitors into customers. We use data, psychology and science to fine tune your webpages and change your visitors' behaviour so that you make more money, are more efficient and profitable.

+44 (0) 161 672 6900
hello@endlessgain.com
endlessgain.com

**Endless
Gain**