

06

Web development and design

In this chapter, you will learn:

- » How the web development process works, from planning through to design and launch
- » Development and design best practices and the principles of designing for persuasion
- » How to assess the quality and effectiveness of web development and design implemented by suppliers or agencies
- » How to evaluate the need for either a static or CMS website.

6.1 Introduction

Websites are, in many ways, at the heart of successful digital marketing. They are your home on the web, a shop window over which you have full control and often the first place people stop to find out more about you.

Web development and design applies to more than just websites, the principles can be used for any digital assets you create, from mobile platforms to social media profiles.

Creating online assets involves three key processes: planning and design, which create the appearance, layout and style that users see; and development, which brings this imagery to life as a functioning web tool.

The fundamental principle of good development and design is to understand your users. They are the people who will actually be using and interacting with your website. What are they looking for? What are their objectives? Your offering must have user experience central to the process.

NOTE

Unlike social media properties, your website is not subject to changes in policy and content remains easily accessible regardless of its age.

6.2 Key terms and concepts

Term	Definition
Accessibility	The degree to which a website is available to users with disabilities, such as the visually or hearing impaired and those technically disadvantaged by not having the necessary device, software or browser.
Adaptive web design	Websites that respond to a user's screen size by loading predefined layouts.
Alt text	Alt text means alternative text. The 'alt' attribute is used in HTML to attribute a text field to an image on a web page. It normally has a descriptive function, telling a search engine or user what an image is about and displaying the text in instances where the image is unable to load. Also called Alt tag.
Anchor text	A text link, or backlink, that refers visitors to your site from another with SEO benefits, passing relevance and authority from the referring site.
Bounce	When users leave a site before navigating from their landing page to another.
Branding (or visual identity or corporate identity)	These terms refer to the look and feel of your brand. In this context it is used when discussing how your logo, colours and styling elements are translated from traditional print-based assets to digital.
Breadcrumbs	Links, usually on the top of each page, indicating where a page is in the hierarchy of the website. Breadcrumbs can be used to help users navigate through the website, as well as act as a page index for search engines.

Cache	Files stored locally on a user's browser to limit the amount of data called from the server on a return visit.
Call to action (CTA)	A phrase written to motivate the reader to take action (sign up for our newsletter, book car hire today, etc.). Calls to action are usually styled differently from other copy on a page so that they stand out and draw attention.
Content management system (CMS)	A software system that allows an administrator to update the content of a website, so that a developer is not required.
Common page elements	Items that appear on every page of a website.
Cascading Style Sheets (CSS)	A programming language that defines the styles, such as fonts and colours, used to display text and content. Web pages are one of the places that this language is used.
Client-side	Scripts that run in a user's browser, rather than on a web server.
dpi	Dots per inch (in an image). On the web, the screen resolution is 72dpi.
Graceful degradation	The use of both modern and antiquated web techniques and code to provide a safety net, or fallback, for users with older browsers and technologies.
HTML5	HTML5 is the most current iteration of the HTML (HyperText Markup Language) standard. It is a broad range of technologies that allow for rich media content and interaction on websites that do not require additional third-party plugins. It allows rich multimedia content to be displayed and easily viewed by users, computers and devices.
Information architecture	The way in which data and content are organised, structured and labelled to support usability.
JavaScript	A high-level, dynamic programming language commonly used to create interactive effects within web browsers.
Landing page	The website page that a user is sent to after clicking on any link or CTA, in an email or affiliated site, in a display ad, or a paid or organic search engine listing. The landing pages that have the most success are those that match up as closely as possible with the user's search query or intention.
Metadata	Information that can be entered about a web page and the elements on it to provide context and relevant information to search engines. Metadata includes meta and title tags.
Native mobile application	A mobile application designed to run as a program on a specific device or mobile operating system.
Navigation	How a web user moves through a website and the elements that assist the user in doing so.

Open source	Unlike proprietary software, open source software makes the source code available so that other developers can improve on or build applications for the software.
Plug-in	Often referred to as a module or extension, a piece of third-party code that extends the functionality of a website.
Progressive enhancement	The development of web technologies in a layered fashion, prioritising basic content and functionality for all web browsers, while allowing users with higher bandwidth or browsers access to an enhanced version of the page.
Proprietary software	Any software that one or more intellectual property holders own and licence to others in exchange for compensation, subject to certain restrictions. Licensees may not be able to change, share, sell or reverse engineer the software.
Responsive web design	Websites that fluidly respond to a user's device or screen resolution based on media queries sent between the site and the device regarding the specs of the device.
Search engine results page (SERP)	The actual results returned to the user based on their search query.
Server-side	Scripts that run on a server, as opposed to a user's browser.
Sitemap	On a website, a page that links to every other page in the website and displays these links organised according to the information hierarchy. While this is often physically available on a website (HTML sitemap), it should also be created as an XML file and included within the Robots.txt.
Universal Resource Locator (URL)	A web address that is unique to every page on the Internet.
Usability	A measure of how easy a system is to use. Sites with excellent usability fare far better than those that are difficult to use.
Web application framework	Software used to help create dynamic web properties more quickly. This is done through access to libraries of code for a specific language or languages and other automated or simplified processes that do not then need to be coded from scratch.
W3C standards	A common approach to development that focuses on accessibility and standardisation, overseen by the World Wide Web Consortium (W3C).
Web server	A computer or program that delivers web content to be viewed on the Internet.

Table 1.

6.3 Web design

Web design is the process of creating all the visual aspects of the interface. This covers the layout, colour scheme, images, logos, type, design elements (such as buttons and links) and anything else that you can see.

The web is a visual medium, so design is an important part of creating assets that are both engaging and effective. Designers need to keep in mind the technical aspects of design, while prioritising the human factor. Digital properties shouldn't just be beautiful. They need to create a good experience for the visitor and meet business objectives, such as increasing sales, creating brand ambassadors, as well as encouraging signups and, ultimately, conversions.

6.3.1 Visual identity and designing for persuasion

The visual interface or, the design of a website, is what user see and interact with. It's the visual representation of all the hard work that goes into developing a website. It's what the site will first be judged by and is the initial step in creating a delightful user experience. It matters a lot.

There is a close relationship between UX and visual design. Ideally the visual designer will use the documents created by the UX designer and add the visual skin, but often the designer has to manage both UX design and visual design.

Design is not just about aesthetics, although looks are very important. Design is about the visual clues we give users so that they know what to do next. It assures web visitors of our credibility and turns them into customers.

Good interface design involves many things, but here are a few basic considerations. These are closely linked to UX and the visual designer plays a key role in defining them.

- **Navigation:** the signage of the site, indicating to users where they are and where they can go.
- **Layout:** how content is structured and displayed.
- **Headers:** the element with a fixed position at the top of every page. They usually includes all primary navigation items which need to be presented on every page such as main menu, login and search.
- **Footers:** the usually consistent bottom part of the page.
- **Credibility:** telling users that you are who you say you are.

Visual identity

The visual identity answers the question, "How do users know it's us?" Certain design elements should be carried through on all web assets created for a brand, as well as print and traditional communication media. Often, the visual identity guidelines for the web are codified into a digital style guide document to ensure consistency across different agencies, designers and teams. This document can include guidelines for creating all manner of web assets, including banners, social media content and website design elements.

The logo is the most prominent way to reinforce your brand identity on the website. The logo is part of a brand's corporate identity (CI).

The primary font is typically used for prominent headings on the site, while body copy is often set in a standard web font that closely matches the primary font.

Menu and button style, as well as icons, are also part of a site's visual identity. Even when a user is viewing a small part of a site or page, it should look as if it belongs to the site as whole.



Figure 1. An example of Red & Yellow's visual identity.

6.3.2 Design theory

Design can be a pretty precise science and there is a lot of research on what makes for effective design. A lot is also common sense and practice based on accepted web standards. Design theory is discussed in the **User experience (UX) design** chapter.

6.3.3 Collecting and collating design assets

Elements such as your logo and brand colours represent your brand and form part of your brand expression. The latest versions of these brand assets need to be available to the designer or marketing agency designing your website.

Getting the right brand assets to designers in a good quality format that they can easily access saves time and avoids expensive mistakes.

Here is a list of brand assets that a designer requires to start working on a site. The quality, format (or file type) and file size are all important considerations. You need to provide:

- **Brand guidelines or style guide.** In Adobe Reader (PDF) format.
- **Logo and other key brand elements.** These could be in Illustrator format (ai) or Photoshop format (psd). Best practice is to have your logo designed using vector graphics. If your logo or other brand assets are created in this format, they can be enlarged without losing quality. If you do not have a vector version of your logo available, then you should make sure that your image is at least 1 000 pixels wide.
- **Image libraries.** Photographs and images can be hosted online, where the designer can access them with a login. They can also be sent via Cloud file sharing services. Make sure the images are of sufficient quality. It is best practice to provide images that are 300dpi. Although all images on the web are displayed at 72dpi, a higher quality image will give your designer room to optimise and resize and crop or cut images where needed. It may also be necessary to consider different images for mobile vs. desktop because the viewpoint on mobile is so much smaller. You may need to consider using much fewer images for mobile or even none at all.

If you do not own the image and its copyright, it is illegal to use the image on your site without permission from the owner. If you require these images, they can be purchased from stock libraries online such as iStock or Shutterstock. Avoid using images from Google Image Search on your pages.

- **Fonts folder.** You will need to provide both Apple Mac and PC versions of the fonts that are listed in your Style Guide. Many designers work on Macs, which use different font versions from those read by PCs.
- **Brand colours** need to be given to digital designers in RGB format. RGB stands for red, green and blue and is the standard for colours online.
- Any **existing creative** assets that have been created for your brand over time, such as:
 - Print designs
 - TV ads.

Website copy should be made available before the final design is required. This prevents delays caused by designers waiting for material.

This applies to any additional assets your designer may need that can be downloaded or sent, such as your price guides or product descriptions.

Fonts

Copy conveys your brand message to your client or customer and should be easy to read and search-engine friendly. The CI is expressed through fonts, also known as typefaces.

Typographic layout can draw attention to the content users should see first. Indicate which pieces of information take precedence. Importance can be signified by text size, colour, weight, capitalisation and italics. Placement also contributes to how important text appears.

NOTE
The brand guidelines or style guide would be created in Illustrator, PS or Sketch, but saved for the client as a PDF doc.

NOTE

Some web safe fonts are: Times New Roman, Arial, Helvetica, Courier New and Lucida Console. See the full list here: www.w3schools.com/cssref/css_websafe_fonts.asp.

Some fonts are common to all computer users. These fonts are known as web safe fonts. Anyone accessing websites that use these fonts will be able to view them as the designer intended and search engines will be able to search these websites easily.

Arial	Comic Sans	Courier New
Georgia	Impact	Times New Roman
Trebuchet	Verdana	

Figure 2. Examples of web safe fonts.

NOTE

You can also use Google Fonts as an alternative, which are more stylish than standard fonts but which are still viewable by most people. The developer will need to implement these. See www.google.com/fonts.

To drive impact, designers typically prefer not to be limited to using only web safe fonts and brand guidelines in most instances don't take web safe fonts into account. This means that fonts must be embedded by a developer using tools such as Typekit, or loaded dynamically from tools such as Google Fonts.

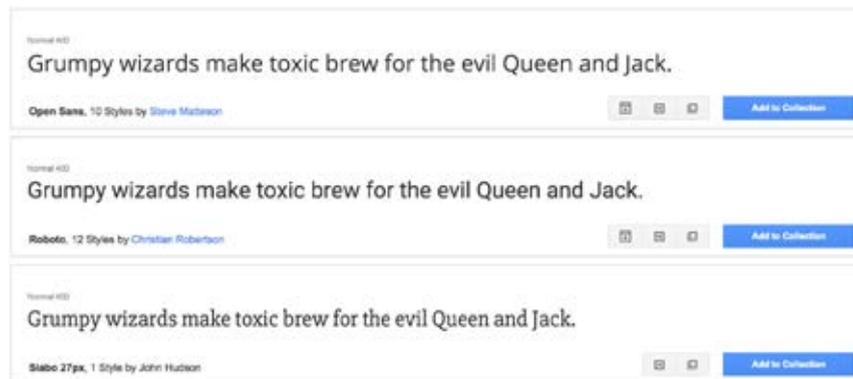


Figure 3. Example of Google Fonts.

6.4 Web development

Web development is the process of taking finished web designs and transforming them into fully functioning, interactive websites. Development is what gives life and movement to static designs and enables users to access the website through their web browsers. This is done by translating the designs into web coding languages that can be interpreted and displayed by web browsers.

Learning to code your own website is not in the scope of this textbook and requires years of practice and some considerable technical know-how. But we can teach you to understand the aspects that go into creating a website, the process that should be followed and how to help in making key choices about your website.

6.4.1 Assessing your development needs

It is important to identify what your development needs are as these fundamentally impact the options that are most appropriate to your site's development. Will you use a CMS? Will you need complex content management? Is it for eCommerce, or is it simply a brochure site? For instance, if you're building a small brochure site, you don't need someone to come up with a bespoke development solution.

Content management

The majority of websites today are built using a content management system, or CMS. Content management simply means a system for managing any forms of content. A web CMS is a software application that assists in managing your digital assets and content for your website. It needs to facilitate the creating, collecting, managing and publishing of any material for your site.

Managing a website is collaborative and involves various people, in various roles, working on the material, such as creators, editors, publishers, administrators and even visitors to your site. A CMS provides tools to allow users with little knowledge of web programming languages or markup languages to create and manage website content. A CMS enables a business to manage and update their own website without needing a web developer (Johnston, 2015).

Ideally, the CMS becomes a set of automated processes that facilitate the functioning, updating and management of your site. Using a CMS means it is cheaper and easier to update, manage and create new content as web developers are not required for each change. A CMS also allows for the content of websites to be updated from any location in the world by means of signing in to the system.

eCommerce

eCommerce, or electronic commerce, refers to any trading of products or services on the Internet. eCommerce sites are necessary across a range of businesses, from consumer based retail, through auction, music and video subscription sites, to inter-organize trading.

Learn more in our **eCommerce** chapter.

Brochure site

A brochure site is a static site. It provides content that does not need to be updated regularly and there is very little interaction with the visitor. A brochure site is essentially a brochure of a company's offering, providing relevant information and contact details to prospective customers. The site does not 'do' anything. The user cannot interact with the site in any way; they are unable to place orders, make payments or engage with the site.

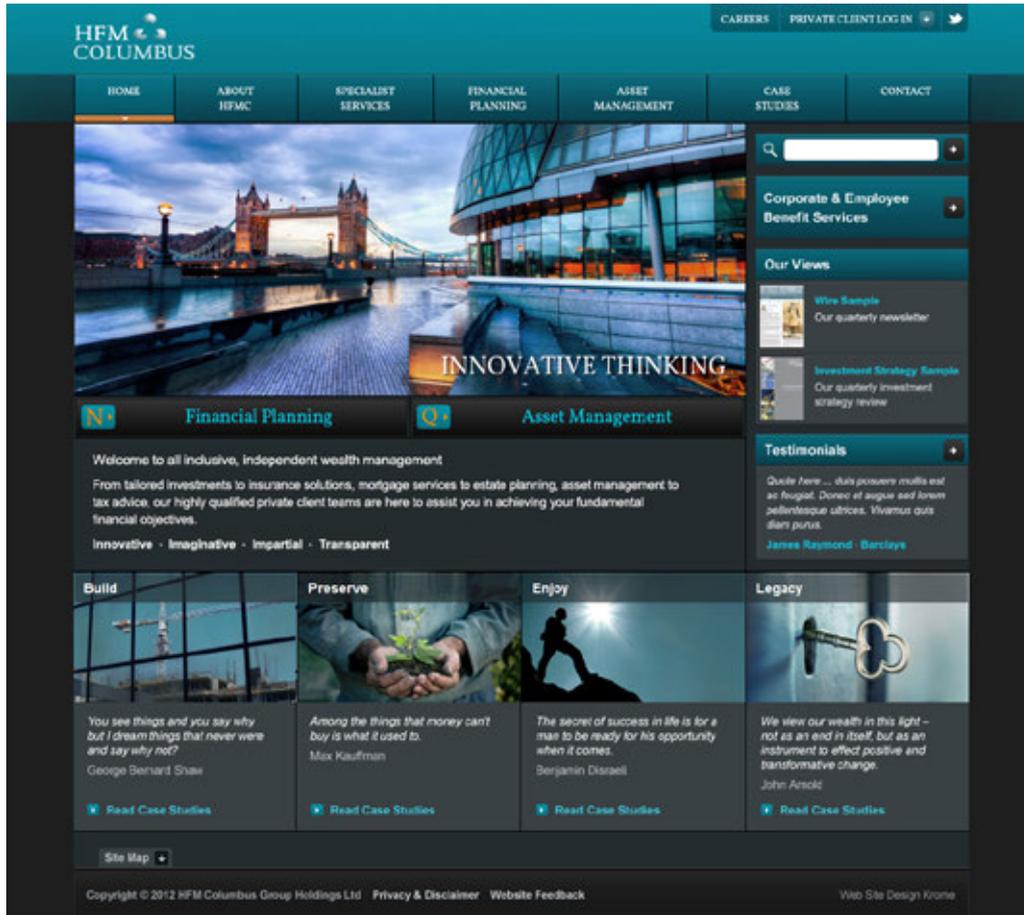


Figure 4. Static brochure site.

Brochure sites are a means of getting your business an online presence quickly and relatively cheaply and if it meets all your business requirements then such a simple site may be sufficient. Consider the limitations of a brochure site before making a choice, as they can be difficult or impossible to build on later.

6.4.2 Development options

When selecting how to proceed with your website development, you have a few options at your disposal. The choice between an off-the-shelf solution and bespoke development comes down to how flexible the off-the-shelf solution is. If too much customisation is required, or it does not support the business' requirements, it may be better and cheaper to develop a bespoke solution.

Off-the-shelf solution

The CMS you choose can be pre-built by an external company or developer. This can be bought like any other software on the market. While this option may provide fewer custom features, it's potentially a more cost-effective option than a bespoke CMS. It is certainly quicker if little customisation is required.

Bespoke development

This involves a CMS that is built specifically for a certain website. This option is highly tailored and customised to your website and can be more expensive than other options. It is possibly less future proof, as finding alternative agencies to support custom-built code is challenging.

Advantages and disadvantages of off-the-shelf and bespoke development

Off-the-shelf		Bespoke	
Advantages	Disadvantages	Advantages	Disadvantages
Cheaper	Overly complex with large sections you will never use	Created specifically for the business' needs	More expensive
Sophisticated software due to wide range of resources input for development	Generic features	Tailored and unique	Requires experienced developers to maintain
Easy to find support and literature widely available	Long time to learn and in-house training required	Customised to interface with software you already use	Less future proof as tied to specific agency to maintain
Easy to share files as software widely used and available	Workflow may have to change to meet software design	More intuitive to your business' way of working	Large investment of time for development, testing
No company time needed for specs and testing	Features you need may not be available	More flexible, can be modified and changed as required	Takes much longer to implement
Available sooner	Individual requests to overall developers will not carry weight	Receives better support	Difficult to get support if developer does not provide it
	Long time to have things fixed if through the corporation that developed the software	Provides significant business advantage	Difficult to choose appropriate developer that will provide reliable and stable software.
	Difficult to gain competitor advantage	Option to sell application to others (if you own rights)	

Table 2. (Cooper, 2015)

Open source vs. proprietary

There are many open source, pre-built CMS options available, some of which are free. Open source means that anyone can see the code that the CMS is built with and can manipulate or improve it and share this with everyone else using the CMS. An open source CMS can be more rudimentary than paid options, but is also easy to tailor to your needs and there is often a community that can create the solutions you need.

Some widely used open source CMS solutions include:

WordPress (www.wordpress.com)

Drupal (www.drupal.com)

Blogger (www.blogger.com)

Joomla (www.joomla.org)

Proprietary		Open source	
Advantages	Disadvantages	Advantages	Disadvantages
Predictability	Less current so large investments in legacy systems	Customisation	Upfront cost can be high if you want highly customised product
Plenty of options	Licensing fees	Flexibility	Less 'out of the box' features and features you need may be expensive to create
Robust and filled with features	Supported only by the company that sells it or its agents	Supported by an online community and not a company	Less predictable support which only happens as needed and not regularly
Relatively cost effective to implement	Lack of customisation	Existing frameworks to work from	Less predictable
	Lack of flexibility	New technology helps to stay up to date with bugs and fixes	
		You own the customised versions	

Table 3. (Robbins, 2015)

A CMS should be selected with the goals and functions of the website in mind. A CMS needs to be able to scale along with the website and business that it supports and not the other way around. Many content management systems have become famous for certain needs.

Examples include:

- WordPress for personal blogs or brochure type sites
- Drupal for more complex community and publishing sites
- Magento for eCommerce sites.

6.4.3 Development frameworks

The back-end of a website refers to the server-side layer. This layer is hidden from the user's view. The interaction between the user and the back-end is handled via a presentation layer known as the front-end or client-side layer. A website is a marriage between these layers (Ferguson, 2016).

Back-end/server-side languages and frameworks

Server-side languages are the hidden web coding languages that determine how your website works and communicates with the web server and your computer.

When choosing a server-side language, you need to consider:

Cost: The cost varies depending on the language you choose for your web development project, as some are more intricate than others, the developer may charge more. So the language chosen may directly influence the salary of the developer. If information is processed where your website is housed, as opposed to on the client's computer, it increases the costs. Some languages also require ongoing website management and maintenance, which is an additional cost to consider.

Scalability: When planning a project where scalability is a factor, consider whether there are developers readily available to develop in this language. Also find out if there are supporting libraries and frameworks available that may suit possible changes to your project.

Some of the most common and popular server-side languages include PHP, Java, Ruby, .NET and Python. Ask your web developer to advise you on the best language for your specific project (Ferguson, 2016).



Figure 5. Logos for some popular back-end developing languages.

Some common back-end frameworks are Django, Zend Framework and Ruby on Rails.

Front-end/client-side development languages and frameworks

Web users have come to expect rich, interactive experiences online and interactive website interfaces are a part of that. Front-end languages, or client-side languages, are languages that are interpreted and executed in users' browsers rather than on web servers.

These experiences range from simple animations through to highly responsive interfaces that require input from the user. There are several technologies available to create such experiences, each with its own opportunities and challenges.

NOTE

There are many free resources online that teach you how to code. One is Codecademy: www.codecademy.com

As with server-side languages, you need to consider a few properties of the front-end language you want to use. Bear in mind that server-side languages and front-end languages are often used together, as all web projects require front-end languages for development.

Cost: Front-end language development costs are comparatively lower than back-end costs, although rich interface developers often demand premium rates.

Features: HTML, CSS and JavaScript open source languages are often used together and are compatible with most hardware and software. Content developed in these is also more search engine friendly. Today, Flash is rarely used despite its interactive multimedia capabilities. In many cases, richer experiences can be achieved with HTML, CSS and JavaScript. What your end users will be able to view should always be the most important consideration.

Scalability: Depending on the capabilities of the device executing the language, certain features may not be available or certain code may run too slowly to create a good user experience. The development of front-end code needs to take all the considered devices into account.

Browser and OS support: With front-end languages, you have to cater for browser and operating system support. A website will look different on each browser and operating system and this needs to be factored in. If a feature cannot be displayed under certain conditions, workarounds have to be implemented. This is typically the case for older versions of Internet Explorer.

Open source or proprietary software: Any developer can create add-ons for or improve on open source software, while proprietary software is owned and its use is restricted. It can be cheaper to develop in an open source front-end language, such as HTML, but as HTML is needed to host all web pages, combinations of open source and proprietary software are sometimes used. However, in most cases and for the languages we cover, this is not a major consideration.

There are several front-end language options to choose from although the most popular by far is HTML coupled with CSS and JavaScript.

HTML5

HTML is the language for creating websites and HTML5 is the fifth iteration of the language. It is also the name for a range of technologies that enable modern web browsing features. It's a specification published by the web standards body, W3C, describing what features are available and how to use them. HTML5 is different from proprietary web software such as Adobe Flash in that the specification is the result of contributions from many organisations and can be implemented by anyone without having to pay for royalties or licensing fees. You do, however, pay for the development tools provided by the companies.



Figure 6. The HTML5, JavaScript and CSS logos.

HTML5 simplifies many common tasks when building a web page, such as including multimedia content, validating forms, caching information and capturing user input data, such as date and time.

HTML5 allows browsers to play multimedia content without the use of Flash or a similar plug-in. There is also a technology called Canvas, which allows developers to create rich, interactive experiences without the constraints that came with previous versions of HTML. For example, a 3D animated video can now be played, something that used to require the use of Flash or Silverlight.

The goal is a website that just works, without the need for particular browsers or plug-ins to enable certain functionality. Having a standardised way of implementing common features, means that the web is open and accessible to all, regardless of competency.

CSS

CSS stands for Cascading Style Sheets and is a style sheet language used to instruct the browser how to render the HTML code. For example, the plain text on a web page is included in the HTML code and CSS defines how it will appear. CSS can set many properties including the size, colour and spacing around the text, as well as the placement of images and other design-related items. CSS pre-processors such as LESS, SASS and Stylus are also available to make CSS more easily maintainable and scalable. They allow for more functional CSS compilation. The latest standard of CSS is CSS3. It is backwards compatible with all versions of CSS and provides many more useful features such as text effects, 2D/3D transformations and animations to name a few (w3schools, n.d).

JavaScript

This is the most common client-side language used to create rich, dynamic web properties. Because it is an open source language, many developers have added functionality that can be more quickly implemented. For example, there are over 1 000 different gallery systems created by JavaScript developers for JavaScript developers.

Flash

Adobe Flash is a language for creating rich, interactive experiences. It supports video and is often used to create game-like web experiences. Although widely supported by desktop browsers, it has limited (and lessening) support on mobile devices and is not usable on Apple devices such as the iPhone and iPad. It has a history of being problematic for SEO, although there are ways to work around much of this.

Flash usage has been on the decline since some security holes were exposed and many believe that it is on its way out. In February 2016, Google announced that its advertising networks, AdWords and DoubleClick, would no longer be supporting Flash. The ads would have to be updated to HTML5 (Google AdWords, 2016). YouTube announced that it would not be using the Flash player by default anymore. It switched to HTML5 for all the latest browsers. Adobe discontinued Adobe Professional CC and released Adobe Animate CC which is now Adobe's premier tool to support HTML5 content. There are still a few traces of Flash left as it is still used as a video player and for the creation of online games.

Frameworks

Frameworks are packages that are made up of a structure of files and folders of standardised code (HTML, CSS and JavaScript, for example) which can be used as the basis for developing websites. Essentially, frameworks are templates to provide

the common structure for websites so that developers don't need to start from scratch each time. Frameworks save a lot of time and money.

Some examples include Backbone.js, AngularJS, EmberJS, React.js and the very popular jQuery libraries. Bootstrap is also increasing in popularity as a front-end framework.

6.4.4 Development best practice

Meta and title tag customisation

The CMS you have either selected or created should allow you to enter your own meta tags for each page, as well as allow full customisation of title tags for each page. It is important to note that Google does not use keyword meta tags for ranking anymore [Lincoln, 2015].

URLs

Instead of using dynamic parameters, the CMS should allow for clean URLs by using server-side rewriting. Clean URLs consist only of the path to a web page without extra code. A clean URL could look like this: example.com/cats, while an unclean URL could look like this: example.com/index.php?page=cats. It should allow for the creation of URLs that are:

- static
- rewritable
- keyword rich.

Be careful when building clean, descriptive and dynamic URLs from CMS content. Should you use a news heading, for example, 'Storm', as part of your URL (www.site.com/cape/storm) and someone changes the heading to 'Tornado' (www.site.com/cape/tornado), this will alter the URL and the search engines will index this as a new page, but with the same content as the URL which had the old heading. Bear this in mind before adding dynamic parameters to your URLs.

Customisable navigation

A good CMS will allow for flexibility when creating the information architecture for a website. For the purposes of adding additional content for search engines, a CMS should not require that all content pages be linked to from the home page navigation. Responsive considerations also need to be in place for mobile devices.

The CMS needs to have good support for managing SEO considerations such as URL rewriting and avoiding duplicate content issues.

Customisable image naming and alt tags for images: A good CMS will allow you to create custom alt tags and title attributes.

robots.txt management: robot.txt files are .txt files that restrict search engines from indexing certain pages of information. Ensure that you are able to customise the robots.txt to your needs, or that this can at least be managed using the meta tags.

Finally, using a CMS that supports standards compliant with HTML and CSS is very important, as without it, inconsistencies may be rendered across various browsers. Compliance also ensures faster loading time and reduced bandwidth, makes markup

easier to maintain, supports SEO efforts and ensures that every visitor to a website, no matter what browser they are using, will be able to see everything on the site.

6.4.5 Developing for multiple screens

Accessing the Internet has changed drastically over the last few years. Desktop is no longer the only or even primary means of accessing the web. Mobile Internet usage surpassed desktop in 2014 and continues to grow year on year, while desktop usage is declining [Chaffey, 2016].

Because of this, it's important for all brands to be accessible on mobile devices. As you learnt in the **User experience (UX) design** chapter, mobile devices can fall into a range of categories and not all mobile devices have the same features and screen size. This means that websites need to be designed to be accessible and are optimised for a variety of screen sizes and devices.

Developing for a variety of screens and mobile requires an understanding of the opportunities and challenges presented by mobile technology. Challenges include the obvious, such as a smaller screens and navigation limitations, different operating systems, as well as more complex issues, such as file formats and bandwidth restrictions.

Mobile devices

A mobile device is a small device with computer-like functionality. It allows for an Internet connection and various features such as Bluetooth, NFC, Wi-Fi and GPS. These include smartphones and PDAs [MDN, 2016]. Remember, mobile goes beyond just the mobile phone, also consider tablets, game consoles, netbooks, wearables and a range of other web-enabled devices

The constraints with developing for a range of devices

Due to the size constraints of mobile device screens, various considerations must be taken into account to allow your website to render correctly.

Fluid CSS layouts will allow the site to manipulate its contents based on screen size. Additionally, CSS media queries can use target styles for a specific screen width, height and pixel density.

Working with touchscreen means that no hover effects will work. Adequate space must be allowed around inputs due to the touch-area of some user's fingers being larger than a mouse cursor.

Images must be optimised for mobile screens and bandwidth restrictions.

Development approaches

Specialised mobile development

Mobile devices allow users to access information about your brand on the move. Because mobile penetration is so heavy and many users worldwide will access the Internet first and primarily through mobile [Chaffey, 2016], every website needs to be designed with the mobile device in mind. Mobile website interfaces demand a simpler approach and a consideration of screen size and input method. A specific design for various mobile screen sizes in the form of an adaptive design may be an option. However, Google prefers responsive to adaptive sites, so creating an adaptive site may not be within the best interest of your business in terms of SEO.

NOTE

Read more about this in the **User experience (UX) design** chapter.

NOTE

A great online JPEG/PNG compression tool is tinyPNG.org. Simply upload your images and let them compress it for you [MDN, 2016].

A native mobile app is software designed to help users perform particular tasks. Examples include a tool for checking the weather, a fuel calculator, or an airlines app to check-in or to track flight progress.

Mobile apps can be sold or made available for free. Many developers create apps to derive an income, while free apps that offer users value are often sponsored by brands or advertising. An app can be an excellent tool for connecting with your customer.

The key difference between native applications and mobile-optimised websites is that native applications are designed for particular handsets and operating systems and have to be downloaded to the mobile device. Whereas mobile-optimised websites can be accessed using any Internet-enabled mobile device. That said, mobile apps can allow for more integration with the device and hence a better user experience, depending on the complexity of the functionality.

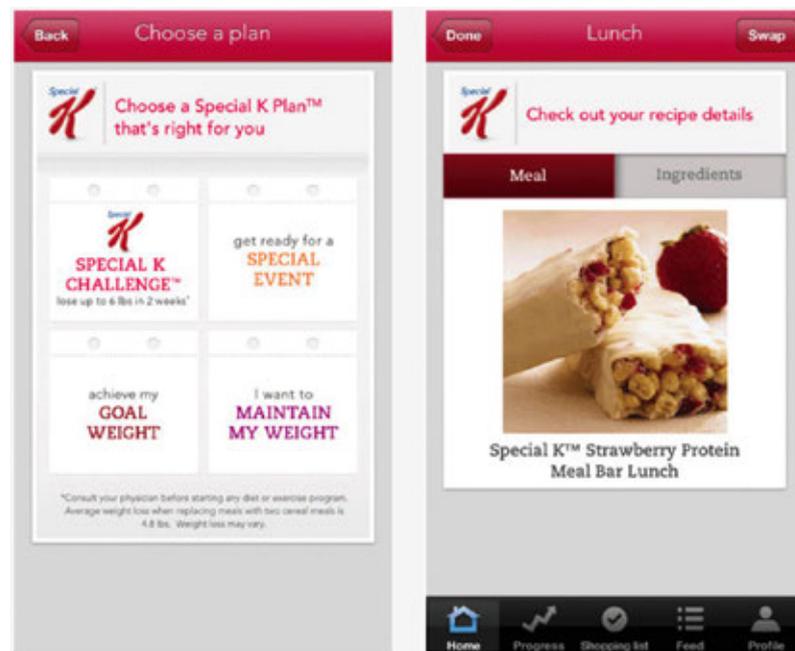


Figure 7. An example of a branded app from the iTunes App Store.

It is a good idea to focus on mobile-optimised sites when targeting a broader group and building an application when wanting to reach a niche or targeted audience.

NOTE

Read more about mobile-optimisation in the **Mobile channels and apps** chapter.

Some brands still create a separate mobi site just for their mobile users, but this is falling out of fashion as Internet access via mobile increases. Google recommends responsive sites as best practice.

A responsive site

A responsive website is a website that changes its layout depending on the device it is displayed on so it looks one way on a desktop computer, but then adapts to the smaller screen size and layout on a tablet or mobile phone. In this way, a single development project can cater for multiple device form factors.



Figure 8. Responsive web design across a range of devices.

A responsive website is not the same as an adaptive website. A responsive site uses the browser's screen space to determine how to reflow the original design content that was probably optimised for desktop, while an adaptive site provides a specifically tailored design for the device you are using. Designing an adaptive site requires multiple fixed layout sizes, usually the six most common screen widths and the relevant one is deployed depending on the screen size of the device.

Many users prefer responsive design as it provides familiarity, uniformity and seamlessness, which are important considerations in user experience (Soegaard, 2016). Responsive design should be mobile friendly. This helps to maintain usability when reflowed for a mobile device's screen.

Adaptive designs are more labour intensive and more expensive. As an approach, it's being used less and less. Although industry professionals often prefer adaptive sites, industry preference won't translate into higher listings on SERPs while Google's ranking methodology favours responsive design.

Creating a responsive website means you only need to build one website for the full range of devices, from desktop to mobile. This can be a technically challenging exercise and will require a lot of planning upfront to make sure that the site displays correctly on each device.

Here is a table that compares the relative strengths and weaknesses of each option. There's no right or wrong answer on which one to pick. Choose the option that best suits your brand, target audience and digital objectives.

NOTE

Try visiting roxik.com/cat and resize your browser to see the cat change shape based on the screen width. This is a great illustrative example of a fluid responsive layout.

	Strengths	Weaknesses
Mobile-specific adaptive sites	<ul style="list-style-type: none"> • Design best suited to screen size • More aesthetically pleasing and provide better UX • Fully optimised for mobile 	<ul style="list-style-type: none"> • Google prefers responsive sites • Need at least six different specific screen-size designs • Expensive • Labour intensive
Native app	<ul style="list-style-type: none"> • Versatile and creative tools can be created • Interactive and fun • Can create real added value through innovative approaches not possible via a web browser • Ideal for frequently repeated or routine tasks • Promote brand loyalty • Enables access to core phone functions such as GPS, camera, etc. • Could generate income as a 'paid application' • Performance benefit in some cases 	<ul style="list-style-type: none"> • Doesn't work on feature phones • Different versions needed for different phone makes and models • Entirely different and complex development process • User needs to choose to download them • Users without additional phone storage may not have enough space to install the app • All apps must go through formal app stores and need to be approved in some instances • Changes need to be released through version updates
Responsive site	<ul style="list-style-type: none"> • Device is an 'agnostic' solution • One consistent site accessible across many devices • One data set to work from • Future-proof option that will work on most devices • Preferred by Google and scores highly in algorithms for SEO • Users prefer the uniformity and consistency from responsive sites used across devices 	<ul style="list-style-type: none"> • Could be complex to develop • Site needs a lowest common denominator approach to cater for all devices • May not work correctly on all sizes and shapes of devices • No agreed standard way yet to develop responsive sites

Table 4.

Designing for multiple screen sizes

Designing a site that will display consistently across multiple devices and screen sizes is difficult, but understanding and sticking to web standards will bring you closer to this goal.

Design your site so that the information your users want is not only on display, but also easy to get to. The limited screen space is valuable, so you can't necessarily have the full site navigation on every page. Well thought-out information architecture is essential to ensuring you make the most logical use of navigation in line with what your site visitors need.

Standards

There are few standards currently in place to ensure your design will be optimised across multiple screens. Creating content, including images, text and beyond, that can be correctly formatted on most mobile devices, or at least legible on phones where formatting is flawed, is still not entirely possible. There is therefore a certain

amount of trial and error involved in designing a site optimised across a variety of devices. The process is certainly worth it, though, considering that there are 4.7 billion unique mobile subscribers and the majority of these are accessing the mobile web (GSMA Intelligence, 2016).

Web standards are managed by the **W3C**. The standards were created to promote consensus, fairness, public accountability and quality. Complying with web standards means the site uses valid code and adheres to stipulations from the W3C. Read more about web standards at www.w3.org/standards/about.html.

Mobile handset emulators allow you to see how your work-in-progress website will be formatted, depending on which device you are emulating. It has been suggested that nothing can replace testing on actual mobile devices, so if you are doing the testing, recruit contacts with different handsets to show you the difference in display. Some emulators:

- BrowserStack – www.browserstack.com
- TestiPhone – www.testiphone.com
- Mobile phone emulator – for Samsung, iPhone, BlackBerry and other - www.mobilephoneemulator.com
- Safari on the iPhone can be tested with IOS simulator.

Responsive design

Responsive websites are designed for a range of screen widths. When deciding whether to create a responsive site or adapt desktop sites, consider your customer first:

- How much of your website traffic comes from specific mobile devices? If this is a large percentage, consider building a responsive site designed for optimal viewing on mobile screen sizes.
- Do your desktop users have the same goals as your mobile users? Here you need to keep in mind your CTAs, drop down menus and the like and ensure they can be accessed correctly by the relevant device.
- What is your budget and how quickly do you need your website to be built? Responsive websites take a while to build and can be expensive. You could save money long term by going this route, but there is a sizeable upfront investment.
- Do you have an existing site and can it be converted into a responsive website, or will it need to be rebuilt (Du Plessis, 2012)?

Responsive design comes with a fair bit of terminology, but you should be familiar with three key concepts.

Flexible grid

Typical websites are designed as large, centred, fixed-width blocks. With responsive design, the page elements, such as the heading, the text or copy, the main image and other blocks of information, are arranged in a grid of columns that have predefined spacing. Each element relates proportionally to the other elements. This allows elements to rearrange or resize in relation to each other whether the screen is tiny or huge and the screen quality is high or low. Although this system allows for

flexibility, an extremely narrow screen can cause the design to break down. In this case, we can make use of media queries.



Figure 9. Flexible grid used in responsive design.

Media queries

Media queries are bits of code that allow websites to ask devices for information about themselves. The website style that will suit the device best is then selected from a list of styles defined in a CSS. Media queries ask for information about the device's browser window size, orientation (landscape or portrait) and screen display quality.

Flexible images

Images are designed to move and scale with the flexible grid. How fast the website loads is an important consideration, so high-quality images are made available for larger screens and lower-resolution images are made available for smaller screens. Parts of images can also be displayed for smaller screens to maintain image quality. Images can even be hidden completely. Image optimisation is done in CSS, which queries the screen height, width and pixel ratio of the device and then adjusts the images accordingly (MDN, 2016).

For more information about responsive websites watch Methods for mobile (Responsive vs. Adaptive) from Brian Wood Training: www.youtube.com/watch?v=lqojC1D3QpU

6.5 The web development process

This section discusses the process of building a website from the client's perspective.

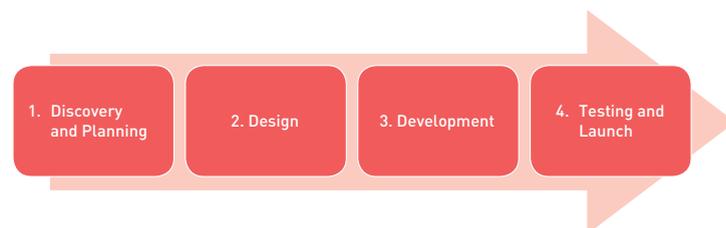


Figure 10. The web development process.

6.5.1 Step 1: Discovery and planning

Planning a website starts with research of your market, your users, your competitors and your business. If you already have a website, you can use existing web analytics data to understand how well you are meeting your users' needs. It's also worth running some user labs to watch how users interact with your existing site.

Key questions you need to ask.

Business: What are your business objectives? How should this digital property help you to achieve those objectives? For example, should it generate leads for you to follow up on? Is it an eCommerce store?

Users: Who are your users, your potential customers? What problem does your website need to help them solve? For example, maybe your website could collate information in one place like www.tripit.com.

This research helps you to plan your website strategically, ensuring that it is aligned with both user needs and business objectives.

In research and planning, you should also reach an understanding of what tasks or actions users need to do on your website. These are usually in line with your business objectives. Some tasks a user may need to do include checking the availability of a hotel, signing up for newsletters, or printing information.

Before any web design project starts, decide what browsers, operating system and devices you will develop for. Usually for browsers you use the latest, plus one before. For devices, iOS and Android are quite different so if you are optimising for both this can be quite expensive. Technology moves so fast; you cannot keep up with all the latest OS and devices.

Choosing a domain name

Domain names are important. They are part of the URL of a website. A domain name looks something like this: www.mycompany.com

A lot more information can be included in this. Domain names can carry the following information: subdomain.domain.tld/directory

Domain – the registered domain name of the website

Subdomain – a domain that is part of a larger domain

TLD – the top level domain, uppermost in the hierarchy of domain names

directory – a folder to organise content

The TLD can indicate the country in which a domain is registered and can also give information about the nature of the domain.

com – the most common TLD

co.za, .co.uk, .com.au – these TLDs give country information

.org – used by non-profit organisations

.gov – used by governments

.ac – used by academic institutions

Domain names must be registered and in most cases there is a fee for doing so.

NOTE

Read more about this in the **Market research** chapter.

NOTE

Read more on users in the **User experience (UX) design** chapter.

NOTE

Is responsive design right for your company or client? See how some companies tackled this question here: www.zdnet.com/does-your-companys-website-need-responsive-design-7000021417

Many hosting providers will register domain names on your behalf, but you can also do it yourself.

Domain names should be easy to remember and, if possible, include important search keywords for your business. For example, if you were building a website for your restaurant named Omega, www.omegarestaurant.com could be a better choice than www.omega.com as it contains the important keyword 'restaurant'.

UX and content strategy

You also need to gather, analyse and map out what content is needed on the website. This content is then structured in a process called information architecture. A sitemap should reflect the hierarchy of content on the website and the navigation (how users make their way through a website).

NOTE

Read more about this in the [User experience \(UX\) design](#) chapter.

NOTE

See this link designmodo.com/wireframing-prototyping-mockuping for a good explanation about the differences between wireframing, prototyping and mocking up.

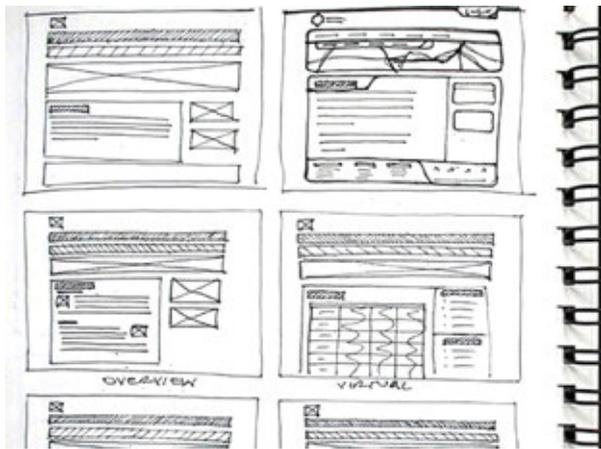


Figure 11. A sketched wireframe of what the web pages will look like and how they will link to one another.



Figure 12. An example of a wireframe against what the final website looks like.

At the same time, consider what content you want to include on your site. Will it be a relatively static site that doesn't change often, or will you need an editable CMS to regularly add and update content, such as blog posts, images and products?

Should the website be large enough to require it, a functional specification document should be created, using all the information compiled so far. This document should detail the development requirements for the website and can be used to communicate any specific design constraints.

It's now time to move on from planning to building.

Search engine visibility

Search engine traffic is vital to a website; without it, the chances are that the site will never fulfil its marketing functions. It is essential that the search engines can see the entire publicly visible website, index it fully and consider its relevance for its chosen keywords.

Search engine optimisation (SEO) has its own chapter in this textbook, but here are the key considerations for web development and design.

In web development, the copy that is shown on the web page needs to be kept separate from the code that tells the browser how to display the web page. This means that the search engine spider will be able to discern easily what content is to be read and therefore scanned by the spider and what text is an instruction to the browser. Cascading Style Sheets (CSS) can take care of that.

If the search engine cannot see the text on the page, it means that it cannot crawl and index that page.

6.5.2 Step 2: Design

Design usually happens before development. According to the steps explained earlier in this chapter, the designer will transform the wireframes and basic planning materials into beautifully designed layouts. These are static images that show how the website will look once it's coded.

6.5.3 Step 3: Development

The development phase usually kicks in once the design is finished. Developers will sometimes start their involvement as early as the wireframe stage by creating low-fidelity prototypes to support the user-testing process. Normally, the developer uses the design templates to code the actual website, using the front-end language that you have chosen. Server-side development and CMS considerations may also be part of this phase.

6.5.4 Step 4: Testing and launch

Once you have planned an amazing site, designed it beautifully, built it skilfully and filled it with fantastic copy, it's time to test it fully and then take it live!

Testing is an important part of website development and design and it should take place throughout the process of planning, designing and building, leaving just final quality assurance (QA) testing before the site goes live. Test subjects should be real potential users of the website, not just members of the development team. The site needs to be tested in all common browsers and devices to make sure that it looks and works as it should across all of them. All links should be tested to make sure

NOTE

Have a look at the discussion on choosing a static versus a CMS website earlier in this chapter.

NOTE

Read more about this in the [Digital copywriting](#) and [Search engine optimisation \(SEO\)](#) chapters.

that they work correctly and it's always a good idea to get a final check of all of the copy before it goes live.

Tools such as W3C's HTML validator, validator.w3.org, should be used to validate your HTML. See more about testing in Section 6.7 Quality assurance.

Make sure your web analytics tracking tags are in place, after which it will be time to take your site live. Now, you need to move on to driving traffic to your newly launched site. That's where all the Engage tactics in this textbook come in handy.

6.6 Development methodologies

There are different approaches to building a website. The one described above is the waterfall process, where one step follows the other. This is in contrast with other methods such as the agile methodology, which involves faster iteration and greater collaboration, but doesn't afford clients as much control and upfront clarity on the deliverables and timelines.

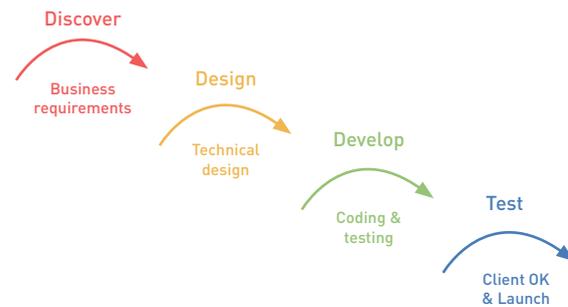


Figure 13. The waterfall method.

The Agile method is a series of sprints and involves working through iterative, incremental cycles. Agile methodology is more collaborative, less rigid and requires incremental investments. It often results in being able to release the final product to the market faster. The collaborative approach means that instead of handing over the project to the next person in the chain, you work together catching any issues and working on each iteration as you move through each process. The agile method often involves scrum methodology and requires (or at least tries for) each sprint to produce something that has an increment of product functionality (Joel, 2015).

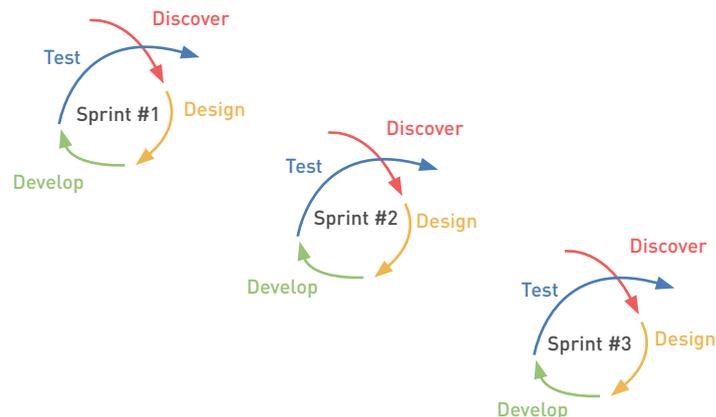


Figure 14. The Agile method.

Careful consideration of the specific project, the goals and ultimately the client's needs will inform the decision on which of these methods is appropriate for a specific project. The method chosen should be the one that will best fit the required project goals.

6.7 Quality assurance

The software development cycle has one final step before the website goes live, quality assurance. This is often referred to as QA. It is a crucial step to ensure that website delivery is of the highest standard and that the client expectation is in line with the agreed Statement of Work. This step is independent of the design and development phases and involves various end-user test cases. Test cases ensure that the graphical user interface (GUI) promotes a great user experience (UX). QA involves two steps: functional and user-interface testing (Thompson, 2015).

6.7.1 Functional testing

This involves testing the features of a website to ensure that they are functioning correctly. Functional testing should be done early in the development cycle as it speeds up development, increases quality and reduces the risk of errors towards the end of the project. Testing can take place either manually by a tester or be completely automated using an application (AppPerfect, 2016).

6.7.2 User interface testing

This is the process of testing whether users can engage with the site as envisioned during development. It also includes testing all features, such as screens with controls, – like menus, dropdowns, buttons, icons, toolbars, dialogue boxes, forms and all other user interface features on the site.

User interface testing is crucial as it is the users' experience of the site that will determine if they will use the site or application in the future or not. If an interface is not intuitive and is difficult to understand, users are unlikely to use that product again. Testing is essential.

The following should be checked during GUI testing:

- Can users input the necessary information into the user fields?
- Does the feature execute the desired function when activated?
- Are error messages displayed correctly and for the correct function?
- Is the font appropriate?
- Is the text aligned?
- Are the colours and fonts and even error messages, visually appealing?
- Are the images clear and displaying correctly?
- Are the images correctly aligned and do they appear where they are supposed to?
- Are the GUI elements positioned correctly for different screen sizes and resolutions?

[Guru99, n.d.]

NOTE

Read more about this in the [Data analytics](#) chapter.

NOTE

Read more about scrum methodology here: www.scrumalliance.org/why-scrum

NOTE

You can read more about GUI testing at www.guru99.com/gui-testing.html.

6.7.3 Cross browser and device testing

With the myriad of browsers and devices available, developers need to ensure that their websites render acceptably across all of them. Perfection is extremely difficult as each browser and device renders a website slightly differently. To ensure compatibility, developers write cross-browser code. If a feature is not supported, a fallback must be in place to ensure that it degrades gracefully.

If you are deploying your site across a range of devices, each version needs to be checked. And if the site is designed to be responsive, check the GUI across a range of devices to test the responsiveness of the design and that all the elements work across the various devices and possible views.

Test websites on different browsers and operating systems, for example Google Chrome on IOS and Google Chrome on Android devices. Various tools are available to assist in this process, one being BrowserStack (www.browserstack.com). BrowserStack allows you to test various operating systems and devices from within your browser (MDN, 2016).

6.8 Case study: Offspring

6.8.1 One-line summary

How moving to a responsive design website helped a UK-based online sneaker store called Offspring to increase mobile conversion rate and mobile revenue.

6.8.2 The problem

Offspring (www.offspring.co.uk) had no mobile website and mobile users were having a poor user experience when accessing the site on their devices. This caused poor conversion rates and low revenue for mobile.

Mobile users are the largest growing online market and Offspring witnessed a consistent increase in mobile traffic to their website. Their current site was not mobile-friendly and was not providing a good user experience. This was also impacting on their SEO score. Offspring knew they needed to ensure mobile users had a good user experience to improve conversions, revenue and SEO.

6.8.3 The solution

Offspring considered a mobile-friendly site, but decided to opt for a fully responsive site to provide an optimised viewing experience irrespective of the device used to access the site. Offspring wanted to ensure that tablet users were also catered for in the design.

The brand also wanted to ensure that they provided existing customers with a familiar look and feel so that they could still easily access and navigate the site.

Certain key design features were included to improve user experience. A new sticky header was included that followed users down the page as they scrolled, providing easy navigation without having to scroll back to the top each time.

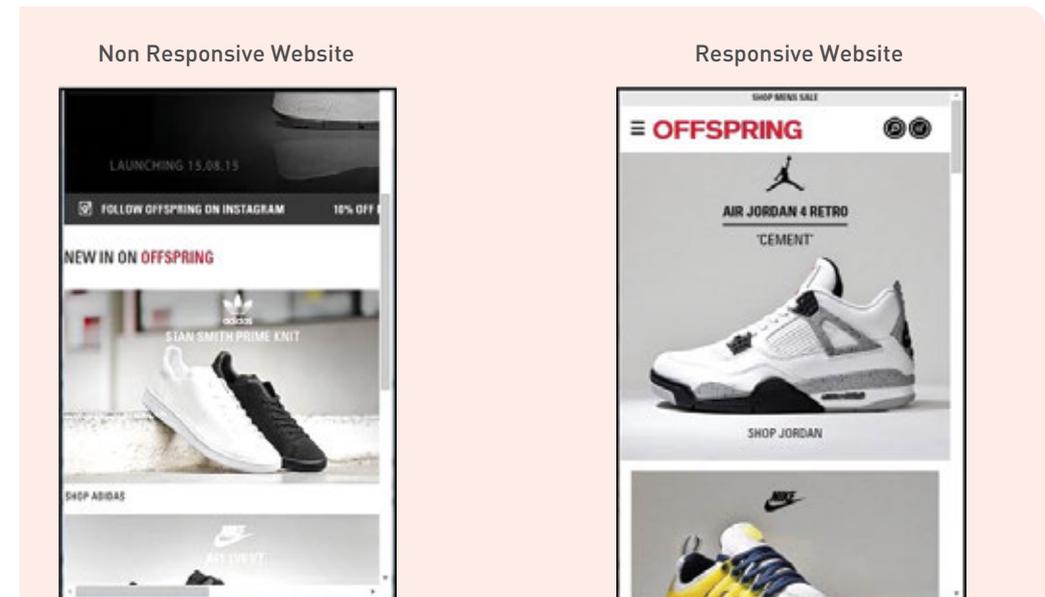


Figure 15. Offspring's responsive vs. non responsive site.

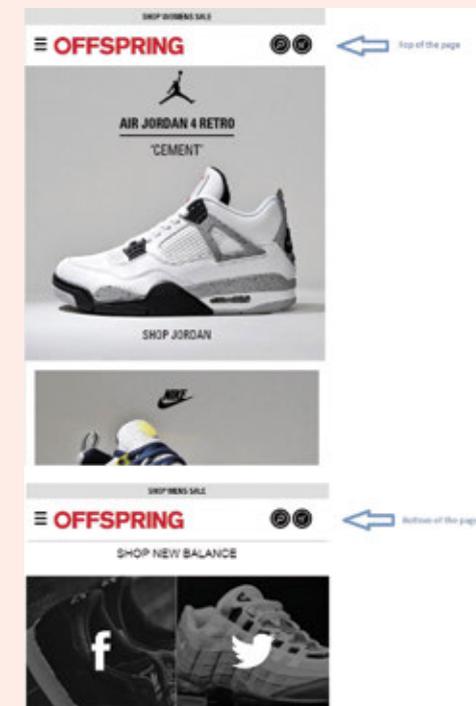


Figure 16. Mobile-friendly elements of Offspring's new responsive design.

A new search bar was included with predictive search. This allowed users to search for their product more quickly.

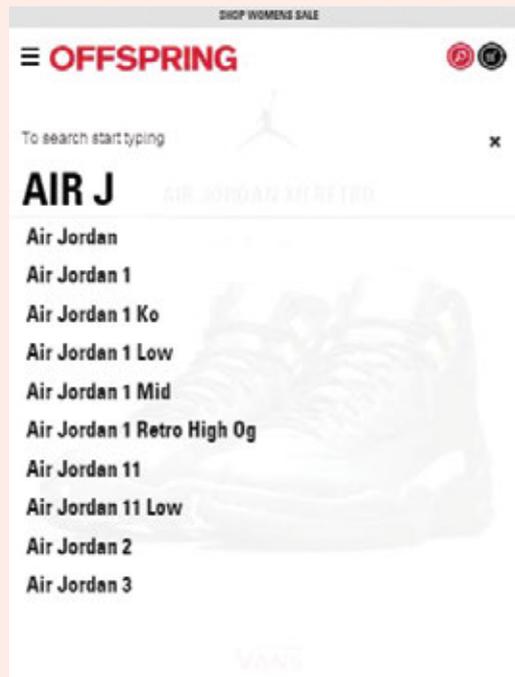


Figure 17. Search bar with predictive search on new responsive site.

Other design features included:

- A new product listing page that was three columns wide on a desktop that could automatically scale down to two or one depending on the size of any different device used.
- New image view and selection process was included on the product detail pages.
- A 'Don't forget' section was included on the basket page so that users could automatically add items to their basket.
- An improved checkout process was created with new delivery options.

The intuitive navigation process was crucial to ensuring that users on any device would be able to find the information they needed quickly and easily.

6.8.4 The results

Users spent more time on the site, which led to an increase in conversions and sales. The SEO visibility of the site showed dramatic improvement after moving over to the responsive site mainly due to the mobile site usability score on Google rocketing from 60/100 to 100/100. This saw organic traffic to the site increasing by over 25%.

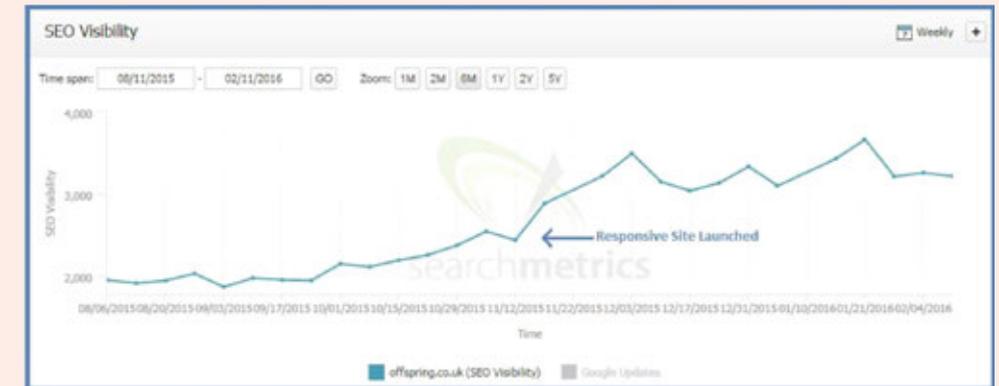


Figure 18. Offspring's growing SEO visibility score.

Revenue increased as well, with:

- A 15.19% increase in mobile/tablet conversion rate
- A 102.95% uplift in mobile/tablet revenue year on year
- A 20.25% increase in the e-commerce conversion rate from tablet users alone.

6.9 The bigger picture

Web development and design can be seen as the thread that holds digital marketing together. After all, websites are the first thing we think of when talking about the Internet.

With the crucial role that search engines play in the way people explore the Internet, web development and design go hand in hand with SEO. And, of course, online advertising campaigns, social media channels, email marketing newsletters and even affiliate programs lead people to click through to your website and sometimes to a customised landing page. That's web design jumping into the mix again.

Setting up analytics correctly on your website is also essential to managing and monitoring your marketing success.

Successful website development and design is all about the right preparation and the resulting website usually forms the foundation of any digital marketing to follow. Make sure you understand your users' needs and that you're building on a strong base.

6.10 Summary

Successful websites come from strong planning with a focus on user needs. Websites should be built to be accessible and usable, search engine optimised and shareable.

Key considerations include:

- Designing your website according to best practices, following the process of developing a website from start to finish
- Developing a strong, stable and usable website

- Creating a suitable mobile web experience for your users
- Enhancing user experience through design and guiding a visitor seamlessly through a website, as opposed to distracting visitors from their goals
- Ensuring consistency in visual messaging across all properties
- Supporting a wide range of web browsers and mobile devices.

6.11 Case study questions

1. When deciding on which type of site to choose, why did Offspring opt for a responsive site?
2. Which two elements of the responsive design were most important in optimising the user experience across a variety of devices?
3. Would you have made the same design decisions? Are there any further considerations you would have designed for?

6.12 Chapter questions

1. What are some key design elements you would ensure were included when designing a website?
2. Why is it so important to realistically assess the needs of your site before development?
3. What do you think are the biggest challenges for optimising your design across multiple devices?
4. What, to you, is the most important step in web development?
5. What role does a website play in an overall digital marketing strategy?

6.13 Further reading

www.alistapart.com – a website for people who make websites, A List Apart has regular articles from web designers and developers on building user-friendly, standards-compliant websites.

www.html5weekly.com – a weekly newsletter filled with the latest must-know HTML5 tips and trends.

www.smashingmagazine.com – a website for web designers and developers.

6.14 References

AppPerfect, 2016. Web functional Testing. [Online]

Available at: www.appperfect.com/products/web-testing/functional-testing.html
[Accessed 31 October 2017]

Chaffey, D. 2016. Mobile Marketing Statistics Compilation. [Online]

Available at: www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics
[Accessed on 10 January 2017] -Link no longer active

Cooper, S. 2015. Bespoke vs Off-the-shelf software,

www.hero-solutions.co.uk/articles/bespokevs-off-the-shelf.asp
[Accessed 31 October 2017]

du Plessis, J., 2012. Responsive versus mobile website design comparison. [Online]

Available at: www.gottaquirk.com/2012/09/17/responsive-versus-mobile-website-design-a-comparison
[Accessed 8 January 2013] -Link no longer active

Ferguson, N. What's the difference between frontend and backend? [Online]

Available at: blog.careerfoundry.com/web-development/whats-the-difference-between-frontend-and-backend
[Accessed 30 October 2017]

Google Adwords, 2016. Google display ads go 100% HTML5 [Online]

Available at: plus.google.com/u/0/+GoogleAds/posts/dYSJRrrgNjk
[Accessed 30 October 2017]

GSMA Intelligence, 2016. [Online]

Available at: gsmaintelligence.com
[Accessed 30 October 2017]

Guru99, n.d. Complete guide for GUI testing. [Online]

Available at: www.guru99.com/gui-testing.html?
[Accessed 30 October 2017]

Johnston, M. 2015. CMS or WCM – Which is Which? [Online]

Available at: www.cmscritic.com/cms-or-wcm-which-is-which
[Accessed 30 October 2017]

ImproveIt, 2015. What should a CMS be able to do? [Online]

Available at: improveit.org/understandit/managing-client-information/what-should-cms-be-able-to-do
[Accessed 25 October 2016] -Link no longer active

Joel, 2015. A Designer's Introduction to "Agile" Methodology. [Online]

Available at: webdesign.tutsplus.com/articles/a-designers-introduction-to-agile-methodology--cms-23349
[Accessed 30 October 2017]

Lincoln, J. 2015. Google Meta tags for SEO, what does Google actually understand? [Online]

Available at: ignitevisibility.com/google-meta-tags-for-seo-what-does-google-actually-understand
[Accessed 30 October 2017]

Moz, 2016. Case Study: Here's how moving to responsive design website helped our brand [Online]

Available at: moz.com/ugc/case-study-heres-how-moving-to-responsive-design-website-helped-our-brand
[Accessed 30 October 2017]

Mozilla Development Network (MDN), 2016. Mobile Web Development. [Online]

Available at: developer.mozilla.org/en-US/docs/Web/Guide/Mobile
[Accessed 30 October 2017]

Network Solutions, n.d. What is eCommerce? [Online]

Available at: www.networksolutions.com/education/what-is-ecommerce
[Accessed 30 October 2017]

Robbins, Z. 2015. Proprietary vs. open source content management systems [Online]

Available at: www.viget.com/articles/proprietary-vs.-open-source-content-management-systems
[Accessed 30 October 2017]

Soegaard, M. 2016. Adaptive vs. Responsive Design. [Online]
Available at: www.interaction-design.org/literature/article/adaptive-vs-responsive-design
[Accessed 30 October 2017]

Sharp, 2014. Why a brochure website doesn't work anymore. [Online]
Available at: www.protofuse.com/blog/why-brochure-website-doesnt-work-anymore
[Accessed 30 October 2017]

Thompson, G. 2015. What is Quality Assurance on the Web? [Online]
Available at: www.unleashed-technologies.com/blog/2015/04/30/what-quality-assurance-web
[Accessed 30 October 2017]

6.15 Figure acknowledgments

Figure 1. Red & Yellow's Brand identity docs, 2018.

Figure 2. Stokes, 2013.

Figure 3. Stokes, 2013.

Figure 4. Screenshot HFM Columbus homepage, 2017. www.hfmcolumbus.com

Figure 5. Lotusnotus, n.d. lotusnotus.com/lotusnotus_en.nsf/dx/does [Link no longer active]

Figure 6. Codecondo, 2016. codecondo.com/10-essential-front-end-development-frameworks-for-web-development-projects

Figure 7. Screenshot, Kellogs branded app in itunes store, 2016.

Figure 8. Interaction design foundation, 2017. www.interaction-design.org/literature/article/adaptive-vs-responsive-design

Figure 9. igentics, 2017. www.igentics.com/responsive-web-design

Figure 10. Own image.

Figure 11. Onextrapixel, 2010. onextrapixel.com/40-brilliant-examples-of-sketched-ui-wireframes-and-mock-ups

Figure 12. Wirify, 2017. www.wirify.com

Figure 13. Adapted Evatotuts+, 2015. webdesign.tutsplus.com/articles/a-designers-introduction-to-agile-methodology--cms-23349

Figure 14. Adapted Evatotuts+, 2015. webdesign.tutsplus.com/articles/a-designers-introduction-to-agile-methodology--cms-23349

Figure 15. imgur, 2016. imgur.com/AQ8ZYT7

Figure 16. imgur, 2016. imgur.com/AQ8ZYT7

Figure 17. imgur, 2016. imgur.com/AQ8ZYT7

Figure 18. Moz, 2016. moz.com/ugc/case-study-heres-how-moving-to-responsive-design-website-helped-our-brand