



Alliance and Technology unit

White Paper

# Best Practices for Designing a Responsive SharePoint Site

## About the Author

### **Muktikanta Sendha**

Muktikanta Sendha is a Solution Architect with more than 13 years of experience at Tata Consultancy Services (TCS). He has developed solutions on a diverse set of technology platforms in the areas of financial services, insurance, manufacturing, and retail. He has led TCS' Web 2.0 and User Experience Centre of Excellence (CoE) and is a Certified Usability Analyst (CUA) from Human Factors International. Currently, he is a Solution Architect for the Microsoft SharePoint platform and a User Experience consultant working for the Alliance and Technology unit of TCS.

Consumers now spend much more time on core digital media platforms (desktop computers, smartphones and tablets) than they did just a few years ago. The screen resolution across these devices ranges from 320 pixels (iPhone) to 2560 pixels (large monitor) or even higher. No longer are traditional enterprise web applications accessed only through desktops. The growing numbers of Gen Y users in the workplace are increasingly demanding that these applications be available on mobile platforms, along with corporate internet and intranet portal access.

Web designers face the difficult task of force-fitting their designs into a wide variety of devices and screen resolutions. Within this growing and diversified user-and-device-ecosystem, it is often a challenge for designers to create designs catering to different types of users and devices. Rather than having disconnected designs for each kind of delivery channel, is there a way to devise a single design that provides an optimal viewing experience, irrespective of the media on which it is delivered?

Responsive Web Design (RWD) is the answer to this problem. With RWD, we can create custom solutions for a wide range of users on an equally wide range of devices. Many responsive frameworks have been developed and are in use. This paper shows how RWD can solve the device-design conundrum and discusses the design approach for creating an intranet portal, using the Microsoft SharePoint 2013 platform and a popular responsive framework named Bootstrap.

# Abbreviations

| Abbreviation/ Acronym | Expansion                                     |
|-----------------------|---|
| CoE                   | Centre of Excellence                          |
| CUA                   | Certified Usability Analyst                   |
| TCS                   | Tata Consultancy Services                     |
| API                   | Application Programming Interface             |
| AWD                   | Adaptive Web Design                           |
| CSS                   | Cascading Style Sheet                         |
| DOM2HTML              | Document Object Model Level 2 HTML            |
| HTML                  | Hyper Text Markup Language                    |
| JSON                  | JavaScript Object Notation                    |
| RESS                  | Responsive Design with Server side components |
| REST                  | Representational State Transfer               |
| RWD                   | Responsive Web Design                         |
| UCD                   | User Centered Design                          |
| UX                    | User Experience                               |
| XHTML                 | Extensible Hyper Text Markup Language         |
| XML                   | Extensible Markup Language                    |

# Contents

|   |           |
|---|-----------|
| <b>INTRODUCTION</b>   | <b>6</b>  |
| 1.1. Possible Solutions for Multi-channel Web Design          | 7         |
| <b>RESPONSIVE WEB DESIGN</b>                                  | <b>7</b>  |
| 2.1. Core Ingredients of RWD                                  | 8         |
| 2.2. Tools for RWD  | 9         |
| <b>DESIGNING A RESPONSIVE WEB SITE</b>                        | <b>9</b>  |
| 3.1. The Business Case  | 10        |
| 3.2. The Design Approach                                      | 10        |
| <b>CREATING A RESPONSIVE SHAREPOINT SITE</b>                  | <b>14</b> |
| 4.1. Building a SharePoint Master Page                        | 15        |
| 4.2. Making the Master Page Responsive                        | 15        |
| 4.2.1. Responsive Frameworks                                  | 15        |
| 4.2.2. Understanding and Leveraging the Grid System           | 16        |
| 4.2.3. Making it Responsive                                   | 18        |
| 4.3. Fixing Common Issues                                     | 19        |
| 4.4. Some Best Practices to Make a Responsive SharePoint site | 19        |
| <b>CONCLUSION</b>   | <b>20</b> |

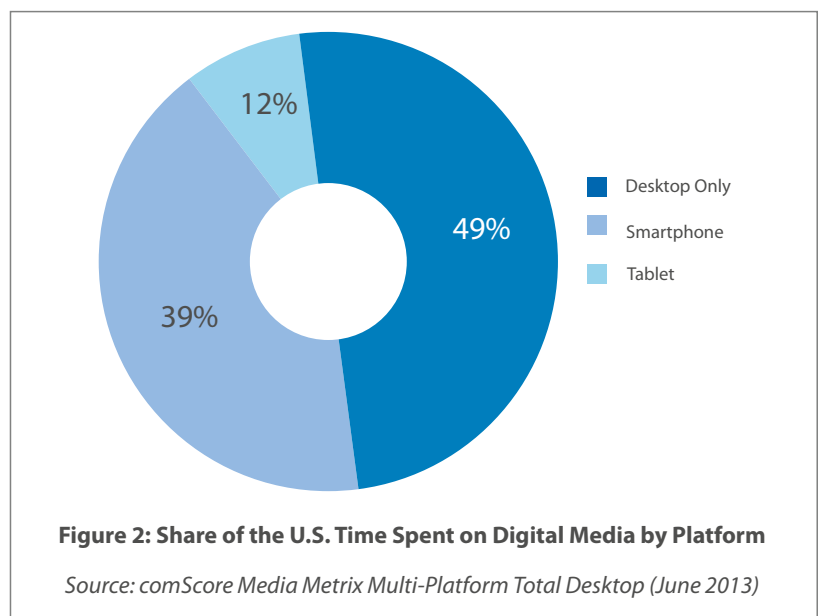
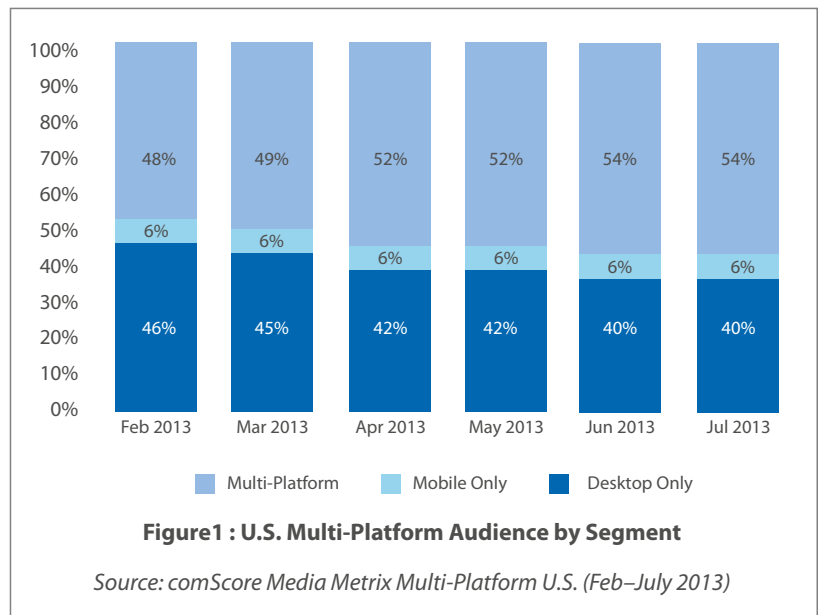
## Introduction

Statistics on internet usage patterns reveal a very interesting phenomenon — more than 50 percent of smartphone users throughout the world use the internet through their mobile devices to get relevant information quickly, especially when they are away from their computers<sup>1</sup>. Around 40 percent of users browse the internet while travelling.

Any business providing services through internet-based channels should not miss out on this growing user base and should avail of all possible delivery channels to provide information and services to its customers. Statistics provided by comScore's Media Metrix show that in April 2013, a multi-platform majority was achieved across the US digital population. By June 2013, 49 percent of digital consumers were engaged on desktop computers and 51 percent on either tablets or smartphones (Figure 1). According to comScore: "The percentage of users accessing digital media only from a PC is decreasing. While the desktop computer is not going away anytime soon, trends like these clearly illustrate that marketers and publishers must begin to put the multi-platform consumer first and change their orientation from the PC-centric consumer view that has dominated their digital strategy for the better part of two decades."

Mobile media usage now accounts for half of all time spent on digital media and is a revelation of how quickly the habits of

consumers can evolve. Mobile is now a significant element in all relevant digital media markets, including content search, video streaming, online advertising, commerce and day-to-day operations.



<sup>1</sup>comScore, 'Marketing to the Multi-Platform Majority', October 17, 2013, accessed January 20, 2014, [http://www.comscore.com/Insights/Blog/Marketing\\_to\\_the\\_MultiPlatform\\_Majority](http://www.comscore.com/Insights/Blog/Marketing_to_the_MultiPlatform_Majority)

## **Current Challenges in Web and UX Design**

As Gen Y enters the workforce and the need for remote connectivity grows, organizations are deploying their web applications and intranet and internet portals on mobile channels. Hence, designers can no longer assume that their sites will be viewed only on a desktop monitor with an average screen resolution. However, given the innumerable number of devices, players, operating systems, screen dimensions, and contexts to design for, User Experience (UX) designers face the challenging task of delivering an optimal viewing experience across these devices.

### **1.1.Possible Solutions for Multi-channel Web Design**

To address this challenge, there are two recommended strategies for UX designers:

- Responsive Web Design (RWD)
- Adaptive Web Design (AWD or mobile websites)

Responsive Web Design (RWD) is an approach in which a site is designed to provide an optimal viewing experience across a wide range of devices, from desktop computer monitors to mobile phones. RWD has been widely adopted by organizations across the globe, but it is not the only solution for designing websites that must work across various media. Adaptive Web Design (AWD), for instance, is an approach for device-specific design. In other words, it comprises specific designs suited for specific media, such as mobile devices, desktops, laptops and so on.

The choice to adopt RWD or AWD will depend on the business situation.

This paper provides information on how to create and make an intranet site work for all channels by employing the RWD technique; specifically, using Microsoft SharePoint 2013 as the development and deployment platform.

## Responsive Web Design

The Responsive Web Design (RWD) approach is adopted to provide an optimal viewing experience, which includes easy reading and navigation with minimum resizing, panning, or scrolling. In other words, it is an approach that enables design and development to respond to the user's behavior and environment, based on screen size, platform, and orientation.

Figure 3 shows how the same website appears on different devices and screen resolutions.



**Figure 3: Responsive display (the same site across different screens)**

RWD comprises flexible grids, layouts and images, and makes intelligent use of Cascading Style Sheet (CSS) media queries to adjust screen resolution and automatically resize images. The intention is to build a site that is resolution- and device-independent.

### **2.1.Core Ingredients of RWD**

The responsiveness of a design is governed by the following three fundamental factors: Fluid Grid, Flexible Media and Media Queries.

#### **Fluid Grid**

Grid systems are used by designers to align and organize elements in a visually pleasing manner on a page layout. The grid system uses a combination of margins, rows and columns. While early designs had pixel-based layouts using fixed grids with the 960 pixel system, later layouts were designed to be more adaptive by using different pixel sizes for different screen sizes. The current practice is to create percentage-based designs which are known as fluid designs.

#### **Flexible Media**

Typically, when adding a media file to a web page, its width and height are set using absolute measurements (usually in pixels). However, this fixes the image size, making it completely inflexible and either too small or too



large for different resolutions. To solve this challenge, flexible media — image, video, and audio files that can grow and shrink to fit within a flexible grid column — are used.

The typical code for making media flexible is as follows:

```
img, embed, object, video { max-width: 100% }
```

### Media Queries

Media queries, a feature added in CSS3, modifies the presentation of content to suit a specific range of output devices without needing to change the content itself. One can make an existing website both mobile and tablet compliant with very little change.

## 2.2. Tools for RWD

The main tools for RWD are HTML5, CSS3, and JavaScript, and jQuery.

**HTML5** is the most recent version of Hyper Text Markup Language (HTML), with features that provide device independence and better error handling, and reduce the need for external plug-ins (such as Flash). HTML5 includes the features of HTML4, XHTML1, and DOM2HTML. It is a development framework with CSS3 and JavaScript, along with a number of new elements, attributes, 2D and 3D graphics, video, audio elements, local storage, and local SQL database support.

**CSS3** is the latest version of the Cascading Style Sheet (CSS) standard and is completely backwards-compatible with earlier versions. New CSS3 features such as orientation (portrait versus landscape), device-width, min-device-width, border-radius, opacity, box-shadow, and text-shadow, help create beautiful web pages which improve the user experience.

**JavaScript and jQuery:** A JavaScript named `css3-mediaqueries.js` can help in cases where browser versions and devices do not support CSS3 media queries. This pre-built JavaScript library allows older browsers (IE 5+, Firefox 1+, Safari 2) to support CSS3 media queries. A copy of the file can be downloaded from <http://code.google.com/p/css3-mediaqueries-js/>.

# Designing a Responsive Website

In this section we outline the process of designing an organization's intranet site using the SharePoint 2013 Web Content Management (WCM) platform. However, the process is not limited to intranet sites alone, and can be used to design external websites and extranet sites as well. We will describe this process by applying a user-centered design method for a fictitious company called ABC Inc. using a specific business case. For easy reference, we will name the application 'Synapse'.

### 3.1. The Business Case

ABC Inc. requires a new intranet site on the SharePoint 2013 platform. The existing intranet has many problems and employees are not using it to the fullest extent possible. The new intranet design should enable users to:

- Know who the 'go to' experts are in the business
- Keep up with the latest updates within the organization
- Share information about best practices with fellow employees
- Ensure that essential documents are updated
- Access important policies while on the road (for employees in the field)
- Act on pending requests round the clock
- Access the site through any device

### 3.2. The Design Approach

Many techniques have been developed over the years to design a website. Our preferred method is the User Centered Design (UCD) methodology which focuses on creating a user experience based on user input rather than forcing a change in user behavior when using the product. The UCD methodology incorporates user concerns and advocacy right from the beginning of the design process, and ensures that the needs of the users are foremost in any design decision, so that the site becomes relevant and valuable to them.

The User Centered Design approach (see Figure 4) is described below:

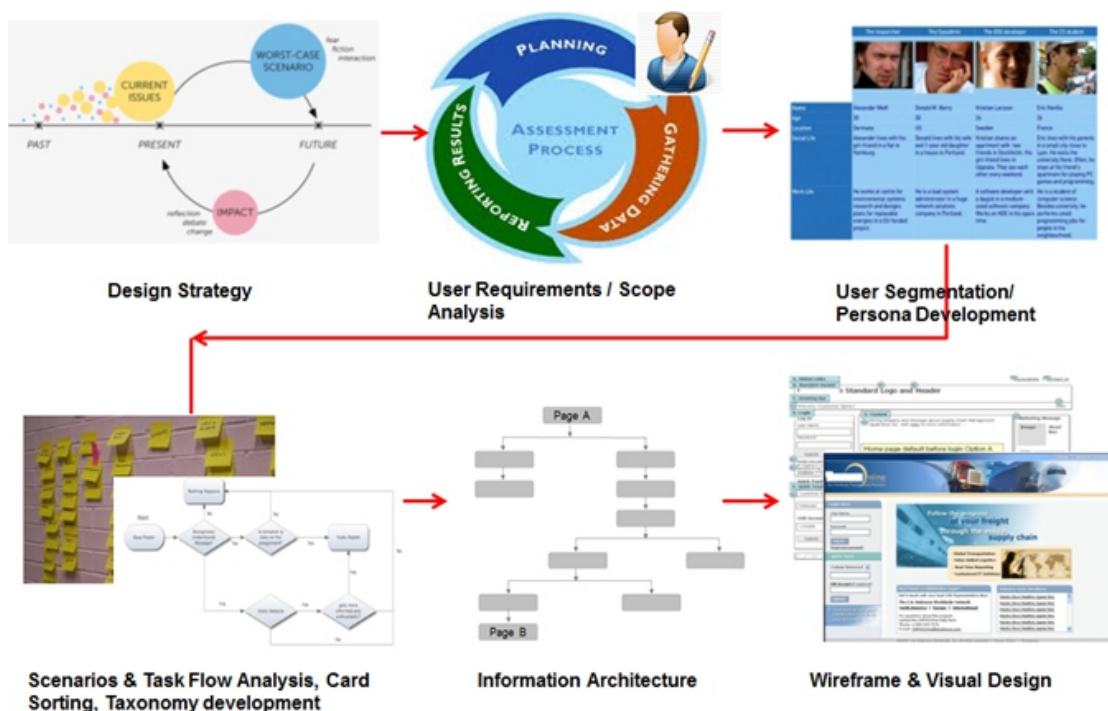





Figure 4: Key User Centered Design activities

**Design Strategy:** To create a user-centric web site, we must draw on the organization's vision, goals, and objectives to create a design strategy. For external websites, business goals (such as increasing revenue, containing costs, and increasing market share) and critical success factors (such as conversion rates, satisfaction, and branding strategy) also play an important role in defining the overall design strategy. Similarly for internal websites, business goals (such as increasing productivity or employee satisfaction) and critical success factors (such as enabling seamless internal collaboration, speed, and accuracy) play an important role. To achieve these goals, we should ask the users of the website about their goals and tasks, current challenges they face, and their expectations and needs. In addition, the business stakeholders' views on the intended future state of the application, as well as the worst-case and best-case scenarios, should be taken into account.

**Usability Assessment:** In the case of existing applications, we should undertake a usability assessment process wherein we analyze user requirements and try to identify the gaps between the new requirements and the existing functionalities. If required, usability testing techniques should also be employed on a case-to-case basis.

**User Segmentation and Persona Development:** Here, we should focus on understanding who the actual users are, their characteristics and interests, their work environment and task profiles, experience level, cultural variance, and so on. This information is then used to develop user-specific personas, such as the three described in Figure 5.

**Task Flow Analysis:** This activity focuses on determining the key user tasks and scenarios based on the type of users. Different methods such as card sorting and primary noun architecture are employed to identify the taxonomy and labels of the system.

|             | <b>Manager</b>  | <b>Employee</b>   | <b>Administrator</b>  |
|-------------|---|---|---|
|             |                      |                                      |    |
| Name        | Michael Smith   | Shannon Huffman   | James Cartel  |
| Age         | 32  | 29  | 26  |
| Location    | USA   | USA   | USA   |
| Personality | Michael is very straightforward. He thinks clearly, loves playing with numbers, and does not talk much. | Shannon is friendly, talkative, and likes having coffee at her desk.  | James stays up-to-date with technology and gadgets, and has the latest phone. He is very talkative and likes to dress casually. |
| Work Habits | Michael prefers to work on his laptop/desktop and likes to play around numbers.                         | Shannon prefers to work on her desktop and likes tools that perform fast and provide information as quickly as possible | James knows the IT systems of his company in and out. He likes coding and is quick to find a solution to a problem.             |

**Figure 5: Examples of user personas**

**Information Architecture (IA) and Interaction Design:** This is the process of logically grouping content in a purposeful and service-oriented way, so that the content is intuitive and easily accessible. Menu structure and page templates can be used for this purpose.

Information Architecture enables all the individual task flows in the user interface. It helps you organize all the tasks and information into groups and levels, increases efficiency by facilitating the location of items, and helps you present content in an easy-to-understand fashion.

Information Architecture and Interaction Design help in assigning navigational schemes at each level. This becomes a vital input for the visualization of the system through site structures, wireframes, and visual designs. The wireframes depict the placement of user interface elements on a page. The site structure defines how the users arrive at a page and where they will go after completing the task.

Once the Information Architecture is frozen, the next step is to design the sitemaps and the global navigation system. Global navigation provides users with the ability to navigate to the most important areas with ease. In our example of ABC Inc., we have designed the navigation based on the following four focal points of usability<sup>2</sup> and three governing principles (see Figure 6).

**Focal Points:**

- Navigation – Can users easily find desired information or functionality?
- Content – Do users get what they want?
- Presentation – Can users easily understand the content?
- Interaction – Can the users act on the interface intuitively?

**Usability Principles:**

- Allow users to intuitively navigate from one web page to another.
- Make it easy for users to understand how menu items are related to each other.
- Ensure that the navigation elements help users land on the pages they actually intended to land on and understand the relationship between the menu items and the pages.

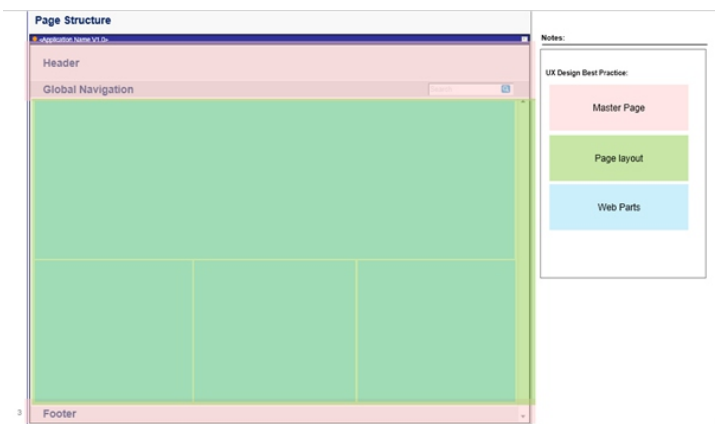


**Figure 6: Global navigation design for the ABC Inc. website**

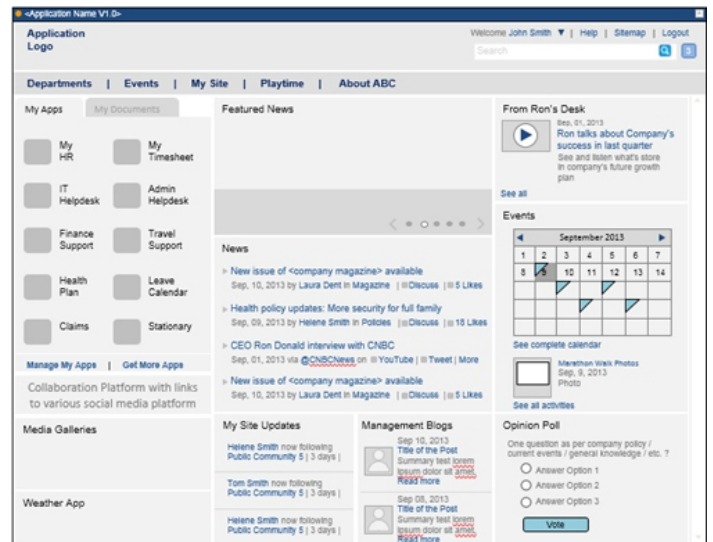
<sup>2</sup>Human Factors International, *The Business of UX Metrics: How to measure and manage the user experience* (July 30, 2007), accessed April 15, 2014, <http://www.humanfactors.com/downloads/documents/Business-of-UX-Metrics.pdf>

**Wireframe and Visual Design:** This activity focuses on the visual representation of the Information Architecture and Interaction Design patterns. It illustrates key features, content, and typical navigation in a visual format. A wireframe is the basic skeleton of visual representation and is used by developers and project managers to validate requirements with the end-users. Usually, it is easier to incorporate changes during the wireframe stage, after which visual mock-ups are created to showcase the look and feel of the site.

Using the information that we have gathered so far, we can now proceed with the wireframe creation for ABC Inc. Let us take a look at the overall site template that will be used for the base SharePoint 2013 master page. This includes global site elements such as header, global navigation bar, main content area, and footer. Figure 7 depicts a core page structure and Figure 8 shows a home page wireframe.



**Figure 7: Core page structure**



**Figure 8: Home page wireframe**

Once the wireframe is reviewed and signed off by end users and business stakeholders, the next step is to create visual mock-ups, keeping in mind the organization's branding and style guides for the look and feel of the web pages. Figure 9 shows the visual design for ABC Inc.'s home page.



Figure 9: Home page visual design

Now that the visual design is in place, we can develop the web pages in subsequent sections.

## Creating a Responsive SharePoint Site

SharePoint 2013 has been enriched with new technologies, including client-side APIs, using standards such as REST and JSON, JavaScript, CSS, and HTML5. It also introduces the concepts of HTML master pages and HTML page layouts.

In order to create a responsive SharePoint site we need to adhere to the following sequence of steps:

1. Create an HTML prototype of the application.
2. Create an HTML master using any design tool.
3. Build the SharePoint master page out of the HTML master page.
4. Make the master page responsive using any popular responsive framework.
5. Test and fix any problems.

Let's assume that the HTML prototype of ABC Inc.'s Synapse site is already built using HTML, CSS, JavaScript, and images. We will briefly discuss the process of transforming an HTML prototype to a SharePoint 2013 HTML master page.

#### **4.1. Building a SharePoint Master Page**

Most companies prefer custom branding for their websites rather than using the default template. SharePoint 2013 allows designers to build custom master pages or a template with a uniform look for the whole site, with a header, a footer, and other elements.

We recommend converting the HTML home page into a SharePoint master page using a common design tool such as Adobe Dreamweaver. The Web Distributed Authoring and Versioning (WebDAV) protocol of SharePoint 2013 makes it possible to use tools such as Microsoft Expression Web, Adobe Dreamweaver, and Microsoft Visual Studio, in addition to Share Point Designer 2013, to connect to and manipulate site files. Using SharePoint Design Manager, designers can also create and edit HTML master pages and deploy supporting images and other elements.

To convert an HTML page to a SharePoint master page, we need to first map the master page gallery to a local drive and upload the entire prototype folder along with all associated files. Using the Designer Manager Interface, we can convert the HTML home page to an HTML Master page. After fixing common warnings and performing a general clean-up we can make style sheet adjustments and add code snippets which enable customization of the HTML master page with SharePoint and ASP.NET controls and components. Out-of-the-box snippets such as Site Logo, Sign In, Breadcrumb, Top Navigation, Vertical Navigation, and Search Box can easily be added to the master page at this point.

Once the new master page meets the design standards and requirements, and all conversion errors are fixed, it can be published and set as the default master page to be used by other site collections.

#### **4.2. Making the Master Page Responsive**

Now that the SharePoint Master page is ready, let us look at how we can make it responsive. In our example, ABC Inc. expects its intranet site to be available through any device at any time. That means we have to capitalize on responsive web design principles which will need a one-time branding effort but can then be adapted to any viewport.

Let us now discuss how we can leverage a responsive framework in the following sections.

##### **4.2.1. Responsive Frameworks**

Some pre-built and well-supported responsive frameworks that are helpful in working with SharePoint 2013 are:

###### **Foundation**

Foundation is a 12 column responsive grid framework with rapid prototyping capabilities and a flexible grid system that can be nested further. This framework can be used to approach web design for smaller devices first and then progressively upgrade those to bigger displays or vice versa.

## **Skeleton**

The Skeleton framework consists of a small collection of CSS and JavaScript files, and comes with a responsive grid layout and standard media queries for device-specific designs. Designs can be made for scaled down browser windows, mobile phones and tablets in landscape and portrait views. Also designers can convert Adobe Photoshop templates (.psd extension) to mock-up web pages.

## **Twitter Bootstrap**

Bootstrap is one of the most popular responsive frameworks with a 12-grid responsive layout. It comes with 13 custom jQuery plug-ins such as carousels and modal windows which is useful during the UI design process.

## **Responsive SharePoint Codeplex Solution**

The Codeplex solution is created as a collaborative effort of a group of contributors and works on both SharePoint 2013 and 2010 versions. It is built on the Twitter Bootstrap and Foundation frameworks and is compatible with SharePoint 2013 Designer Manager, SharePoint Server 2013, SharePoint Foundation 2013, and the Office 365 platform.

From the performance aspect, it supports the Minimal Download Strategy (MDS) of SharePoint 2013. Since it uses the Sandbox Windows SharePoint (WSP) solution-based approach, it can be used by site collection administrators/ site owners as well.

## **SP BluePrint**

SP BluePrint is a light-weight framework with a responsive master page template for customizing and branding a SharePoint 2013 site collection. It has minified versions of SharePoint CSS files to address performance payloads of core Share Point 2013 CSS files, JQuery Pattern Library (Navigation/Tabs/Slideshows) for Web Part Zones, and an out-of-the-box master page for MySites.

In addition to CSS frameworks, we can also make use of JavaScript frameworks. jQuery and Modernizr are two popular JavaScript frameworks used extensively by web developers to create responsive sites.

### **4.2.2. Understanding and Leveraging the Grid System**

Today, using tables for layout design is considered an obsolete idea. There are rendering issues with designs using HTML tables across different browsers, and challenges in scaling down for mobile devices as well. A grid-based design system is a better approach.

To transform a fixed width HTML master page into a responsive one, the first thing to do is to identify a suitable Responsive framework and include its files into our solution. Then we have to adapt the design of each container by leveraging the grid system.

In a grid-based design system, the entire layout is divided into one or more containers. Each container can have one or many rows and each row can again contain one or more columns. We can also include a container inside a column — which means that that column can have its own set of rows and columns.

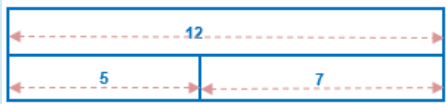


In our example, let us select Bootstrap as the primary CSS framework as it offers more features, documentation and community support than other frameworks.

Bootstrap has a fixed-width grid system and each row container can hold up to 12 columns resulting in a total width of 940 pixels. Each column's width is 60 pixels, amounting to 720 pixels, and there are 10 pixels of padding on either side of each column, adding up to  $11 \times 20 = 220$  pixels of padding, totaling up to 940 pixels.

Although the fixed container is 940 pixels wide, we can include Bootstrap responsive features to vary the grid width in a range from 724 to 1170 pixels. For viewports less than 767 pixels wide, the columns stack vertically and become fluid to fill up the entire width of the viewport.

Figure 9 shows the sample code and layout of a Bootstrap grid that has two rows, where the first row has one column span of 12 column width, and the second row has two spans of 5 and 7 column width respectively.

|  |  |  |
|--|--|--|
| <pre>&lt;div class="container"&gt;   &lt;div class="row"&gt;     &lt;div class="span12"&gt;       Row 1 – Column 1     &lt;/div&gt;   &lt;/div&gt;   &lt;div class="row"&gt;     &lt;div class="span5"&gt;</pre> | <pre>Row 2 – Column 1   &lt;/div&gt;   &lt;div class="span7"&gt;     Row 2 – Column 2   &lt;/div&gt; &lt;/div&gt; &lt;/div&gt;</pre> |  |
|--|--|--|

**Figure 10: Sample code and basic grid layout**

In our example of ABC Inc site's master page, the grid layout can consist of the following (see Figure 10):

**Header:** This contains the company logo, search field, stock ticker, notification bell, welcome user, help, sitemap and main navigation. It has one row with two columns of 8 span and 4 span width.

**Main Content:** This will vary based on the layout of the current page. For the home page, the layout can be one row with three columns of 3, 6, and 3 span width. Each column can again be split into rows and columns.

**Footer:** This contains the quick links, important numbers, Terms of Use, privacy policy, and social media integration points. The footer can have two rows with two columns of 6 span width each.

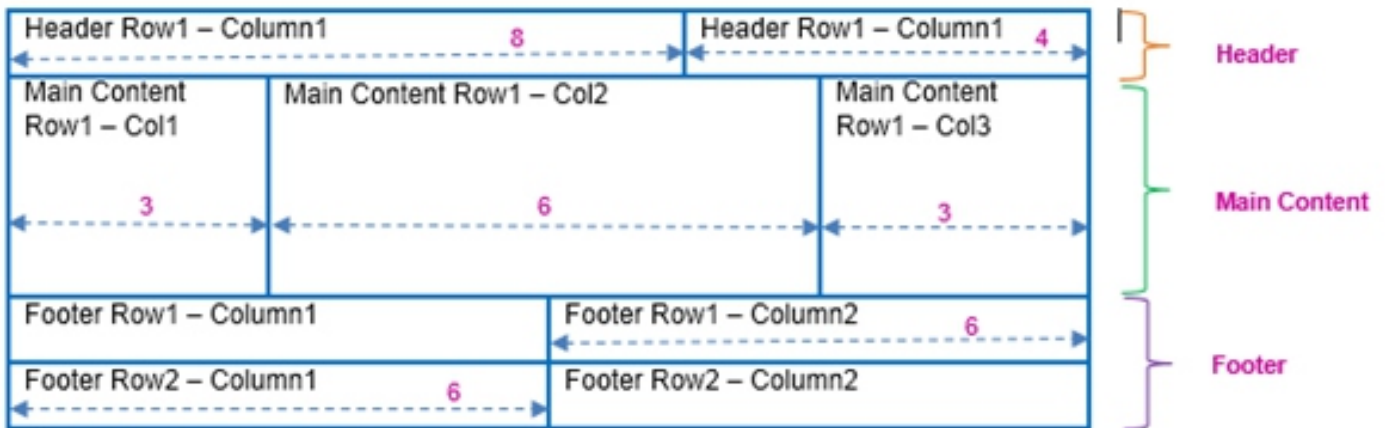


Figure 10: Grid layout for ABC Inc. home page

Once the layout is defined, the corresponding HTML and CSS code are to be developed.

#### 4.2.3. Making it Responsive

To make the page responsive, the Viewport meta tag has to be set correctly. A viewport is the visible part of the screen or monitor on which we view a site. In case of web browsers, the viewport is the visible portion of the entire document. If the document is larger than the viewport, the user can shift the viewport around by scrolling. In the case of mobile devices, the viewport is the screen width.

Using the new CSS3 meta tag 'viewport', we can set the rendered site width to the viewport's native width with the 'zoom at 100% initial-scale' to 1 as shown in the sample code below:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

The browser will adjust the site width to the viewport width of the device (on which the page is loaded) due to the presence of the 'viewport' meta tag while the initial-scale option will enable the browser to proportionately zoom the page to best display the web site for that device's screen size.

Bootstrap also includes targeted media queries, which support displays ranging from smaller smartphones to large displays, even up-to more than 1200 pixels. Each media query has the ability to change the size of the columns to resize the layout to more appropriate viewports, thereby making the page responsive.<sup>3</sup>

Once the Bootstrap framework and basic grid layout has been applied to the master page, there might be a few layout issues, such as misplaced elements, menu items, images, and icons, which should be fixed.

<sup>3</sup>For more details on how to convert a HTML page into a Responsive SharePoint Master page, read 'Pro SharePoint 2013 Branding and Responsive Web Development' by Eric Overfield, Rita Zhang, Oscar Medina, and Kanwal Khipple, published by Apress Publications.

### 4.3. Fixing Common Issues

A few common issues that arise after conversion of the master template are discussed below.

#### Rotating Images Transition Issue

Sometimes, for jQuery-based rotating images and banners, issues arise with the transition from one image/banner to another if image height and width scale beyond the container space, leading to fit issues. The problem can be solved by fixing the image dimensions to that of container height and width. Using the containing class of the slider, we can specify a height for only images within our sliders.

Example code: `.theme-nivo-abclnc .nivoSlider img { height: 100% !important; }`

#### Misplaced Elements

Some elements must be set at a fixed position on the screen. For instance, in our example, the Search Box and Notification Bell controls are placed at the bottom right corner of the main banner area. However, during conversion through Bootstrap, these elements can sometimes disappear from their positions. To avoid this, the container of the element must be set to relative.

Example code: `Header > .container-fluid { position: relative; }`

#### Broken Viewport Layouts

Sometimes, merely using the Bootstrap framework does not always ensure that all elements of our website scale the way we want. Custom media queries and device channels are useful in such cases. We can use viewport breakpoints to scale particular elements and change their positioning, or use Device Channels to detect the end user's device at run time and render the page that is optimal for the device. The viewport breakpoints must be set under 979 pixels for tablets, under 767 pixels for small tablets, and under 479 pixels for smart phones.

The new SharePoint 2013 feature called 'device channel' helps map devices and browsers to the appropriate templates, master pages, layout, and panels. Using the device channel feature, device inclusion rules can be defined and added to user agent substrings for device-specific optimal rendering for the devices included in the rule.

### 4.4. Some Best Practices to Make a Responsive SharePoint site

A few best practices that should be followed to make a responsive SharePoint site are discussed here.

1. Create a base site prototype using a standard HTML editor and then use a responsive framework like Twitter Bootstrap to convert the prototype into a responsive one.
2. Wherever possible, use the grid system to save time instead of creating your own grid systems.
3. Replace pixels with percentages in the layout design.
4. Scale images with `max-width:100%`. For sharper images, use the popular JavaScript file `imgSizer.js`.
5. Enable the Blob Cache in `web.config` file during SharePoint configuration for better image handling.

6. Use CSS3 and Media Queries extensively for styling output (to match the width of the display in the device and then apply the styles to present the content in the available space).
7. Avoid table-based layout design and browser-based HTML, JavaScript, and font tags.
8. If required, use YepNope.js and Modernizr JavaScript library to detect HTML5 and CSS3 features in the user's browser.
9. Create device channels based on matching User Agent.
10. Create a set of image renditions that each image will automatically follow.

## Conclusion

Making a design responsive is relatively easy, less cumbersome, and more productive than developing code for every device available. With RWD, we can create custom solutions for a diverse set of users on a wide range of devices.

With platforms such as SharePoint 2013 and new features such as device channels, responsive frameworks, and the increasing adoption of jQuery and media queries, creating responsive websites has become much easier than before.

However, RWD techniques are not the final answer to the challenges presented by the dynamic mobile landscape. While it is a concept, which, if implemented correctly, can improve the user experience significantly, it is not a universal solution for all users, devices and platforms.

Only by keeping pace with the evolving technology and continuously researching new devices and screen resolutions, can companies aspire to provide consistently rich user experiences.

**Contact**

To know more about us, contact [atu.marketing@tcs.com](mailto:atu.marketing@tcs.com)

**Subscribe to TCS White Papers**

TCS.com RSS: [http://www.tcs.com/rss\\_feeds/Pages/feed.aspx?f=w](http://www.tcs.com/rss_feeds/Pages/feed.aspx?f=w)

Feedburner: <http://feeds2.feedburner.com/tcswhitepapers>

**About Tata Consultancy Services (TCS)**

Tata Consultancy Services is an IT services, consulting and business solutions organization that delivers real results to global business, ensuring a level of certainty no other firm can match. TCS offers a consulting-led, integrated portfolio of IT and IT-enabled infrastructure, engineering and assurance services. This is delivered through its unique Global Network Delivery Model™, recognized as the benchmark of excellence in software development. A part of the Tata Group, India's largest industrial conglomerate, TCS has a global footprint and is listed on the National Stock Exchange and Bombay Stock Exchange in India.

For more information, visit us at [www.tcs.com](http://www.tcs.com)

**IT Services**  
**Business Solutions**  
**Consulting**