

Mobile Web Design & Development Approach for Feature Phones and Smart Phones

Author: Anoop Kumar Gupta

Web Research, Senior Frontend Architect, Delhi, India

Email: gupta.anoop.kumar@gmail.com

Abstract:

The document provides a focus on best practices in order to improve the experience of the Web on mobile devices by finalizing the right approach during design and development phase of a responsive web site or application. As the amount of browser capable devices is growing from Feature Phones to Smart Phones and other Web Capable Future Devices, here we will discuss all aspect of right web design approach to achieve Responsiveness.

Keywords —Responsive Design, Mobile Design

1. Introduction

The document provides a **Point of View** to improve the experience of the Web on mobile devices by finalizing the right approach during design and development phase of a responsive web site or application. As the amount of browser capable devices is growing from Feature Phones to Smart Phones and other Web Capable Future Devices, here we will discuss all aspect of right web design approach to achieve **Responsiveness**.

1.1 What is Responsiveness

Responsiveness – may be a new term for people in web world, but this is what users are expecting during accessing any mobile website, desktop website, native application or hybrid application. As per the recent survey we have huge list of users who are using **phones** to access internet. Here I emphasized on term **phones** which consists of Smart Phones, Feature Phones, Tablets and other High End Devices like Kindle and iPod.

Here Responsiveness means providing information responsibly to user based on device context and other tangible or non-tangible geographical factors like performance, network bandwidth, network speed, and data access cost and device penetration.

We should always consider three major concerns for a mobile web site:

Cost

If you have to develop a mobile website, develop responsibly by using optimum use of images, videos and other heavy media items that may cost high to a user.

Content

Consumers move between multiple devices to accomplish their goals

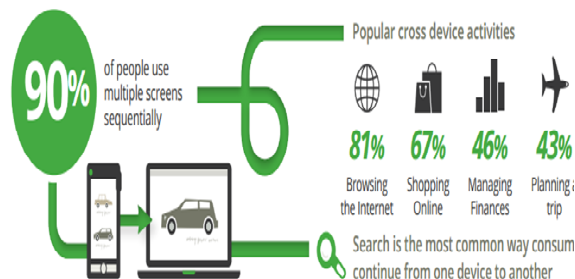


Image <http://www.opensourcetechnologies.com>

Courtesy:

A good content strategy should be there like image size, page size, use of scripts and all.

Context

Does your website add value to users to achieve their goal in a given context?

Now days to achieve the Web Responsiveness – we have three approaches that can be used, these are “**Responsive Web Design**” , “**Adaptive Web Design**” and a separate “**Mobile Site**”.

1.1.1 Responsive Web Design

One Site – One URL

One Code Base.

Serves the same page code to all users.

CSS and JavaScript reformat web pages in the user’s web browser on the client side.

Reorganize and resize each and every element with respect to browse/ device viewport context.

1.1.2 Adaptive Web Design

One Site – One URL – Multiple Elements

Elements or Features can be replaced for example: Drop down Navigation to Sliding Navigation.

Elements or Features can be added.

Unimportant features and content can be removed.

Serve the content and fix features or elements one by one with respect to device context.

1.1.3 Separate M-dot Site

Multiple Sites

Multiple Code Bases

Detects the user’s device type on the server side

Redirects to a separate mobile site based on the device context.

Separate mobile sites have different layout and URLs, but they function similarly to the main website.

They may offer similar, less, more or completely different options than the main site

1.2 Common Misconceptions

1.2.1 Everything can be responsive

Very common misconception is “Everything can be responsive”. We have to understand Responsive to the core and understand that “Everything cannot Be responsive”. But we can achieve responsiveness based on the right approach and that depends on numerous factors like requirements, design, experience etc.

1.2.2 Single information and services can be served for different devices

When Responsive Web Design came into the picture the fair assumption was “Single Web Multiple Context” but it is not true. As per W3C One Web means making, as far as is reasonable, the same information and services available to users irrespective of the device they are using. However, it does not mean that exactly the same information is available in exactly the same representation across all devices.

Reference: <http://www.w3.org/TR/mobile-bp/#OneWeb>

1.3 Defining best technical approach

Nobody can tell the “best” or “perfect” approach but keeping content parity, development time, cost, business goal, user goal and other factors in mind, you can find the optimally right approach and achieve User Goal.

Now a day’s responsive design is the trend, but it isn’t always the right one to take. Sometimes **Adaptive approach with some server side detection** is the more realistic approach to provide mobile-specific content to a mobile user than reformatting it with responsive web design.

1.3.1 Common Issues of Responsive Web Design

Media queries ignore the mobile context

Media queries add more code to create the “mobile” version of the site

CSS media query is a bad choice for multiple devices

Media queries do not optimize HTML or JavaScript or remove unnecessary code

Mobile needs are different from desktop needs

Issues with handling images on varying devices

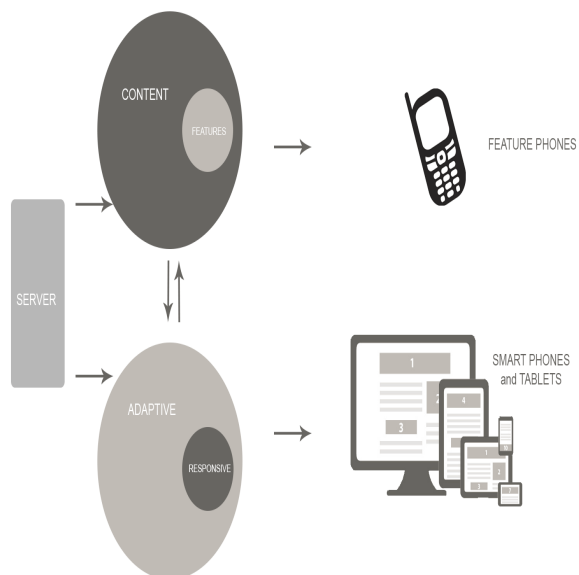
1.3.2 Right approach

So we have three main approaches that can be chosen with respect to the business and user goal: Adaptive web design, Responsive web design or entirely separate mobile sites.

Each approach has their advantages and disadvantages, and it's really a complicated decision from developer's perspective which makes more sense to use.

Our recommendation for sites serving smartphone users is to use "**Hybrid Approach**" which is a combination of **Responsive+ Adaptive** if possible. However, since feature phones do not have the capability to follow CSS media queries, and have limited support of CSS, JavaScript and latest front end technologies. We would need to configure sites to use separate URLs to serve the feature phone content.

Here we have an illustration of the Hybrid Approach Workflow



The idea behind "Hybrid Approach" it is that we combine responsive web design techniques and server side techniques to serve an optimal experience for each device based on device and feature detection. This means that we will serve slightly different requests i.e. URL redirection to some older devices like Feature Phone for a given URL, but we still use responsive techniques for Smart Phone and Tablets.

Hybrid approach allows us to adapt things on server side as well as on client side.

Server side adaptation allows us:

Optimize the pay-load from the server to the user, for example, add or remove markup / scripts / styles for different device groups The ability to alternate content appropriate for context

Client side adaptation allows us:

Ability to test for capabilities – Feature Detection – Modernizr Ability to modify the layout as screen real estate increases - Media Queries – Responsive Web Design – JavaScript solutions

As per Google recommendation, we can use below configuration to serve for Smart Phones and Feature Phones

1.3.3 Separate URLs configuration

In this configuration, a website serves desktops and smartphones on one URL using responsive web design or dynamic serving, and feature phones on a separate URL.

For example:

<http://www.example.com/page-1> serves desktop and smartphone users

<http://m.example.com/page-1> serves feature phone users

1.3.4 How Hybrid Approach Works

Hybrid approach mainly works for the Server-Side Component, handling HTML, CSS, JavaScript, and images.

Based on the user's needs source order, URL structure, media like CSS, Adaptive Images, JavaScript, and application designs can be finely optimized for a specific device class before ever reaching its browser. It generally relies on device/feature detection to device-specific web site.

1.3.5 Device Detection

Nokia has a very good topic on device detection at <http://developer.nokia.com>.

Device detection allows us to identify device properties and characteristics in order to determine the best content, layout, mark-up or application to serve to a given device. These characteristics include screen size, browser type (or version), media support, and the level of support for Cascading Style Sheets (CSS), HTML and JavaScript technology.

1.3.5.1 Why use device and feature detection?

The ability to identify a device, browser or feature enables the developer to perform actions such as:
Serving a mobile formatted site rather than the desktop site.

Swapping style sheets to adapt layout and content to the device's specific HTML, Cascading Style Sheets (CSS) and JavaScript technology capabilities.

Modifying the amount or nature of content to suit the device capacity. For example, a JavaScript based slide show may be served to a higher-end device, and a simple list served to a lower-end device.

Serving smaller or larger images to suit screen size, or swap SVG graphics for bitmap images on less capable devices.

Enhancing functionality on more capable devices by applying progressive enhancements (based on feature and object detection).

Increasing or decreasing font size, margins and padding to scale the actionable areas on a touch device.

Serving different sizes and/or formats of media such as video to suit the device codecs or capabilities.

Read <http://developer.nokia.com/Resources/Library/Web/#!/nokia-browsers/common-elements-of-nokia-browsers/device-and-browser-detection.html> more:

1.3.6 Feature Phone Development Best Practices

In Feature Phone where we have limited or no support of CSS and JS, we used to refer the browsing of site by "click and browse the path", by which user reaches to a destination to get the information. We can plan the site by using below guidelines

Limit main categories to 5

Limit links to 10

No more than 5 levels deep

At least one page per category

Prioritize links by activity

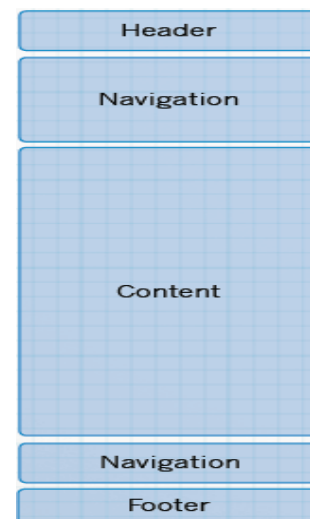
Provide "Back and Home Button" on internal pages

Use Mobile Web Standards for HTML (XHTML-MP)

Use Simple Internal CSS not External

Do not use long forms

Design Vertically



1.3.7 What tool do we use for server side adaptation?

[Device Atlas](#), and [WURFL](#) are great resources as well as [Handset Detection](#) to handle the server side logic for websites adaptation.

<http://51degrees.mobi/> also provides a rich database and services for device detection.

All above are paid database and services.

However we have some free services and one of them is called as **Categorizr**, it is a modern device detection script.

You can read more about [Categorizr here](#).

2. Design Process

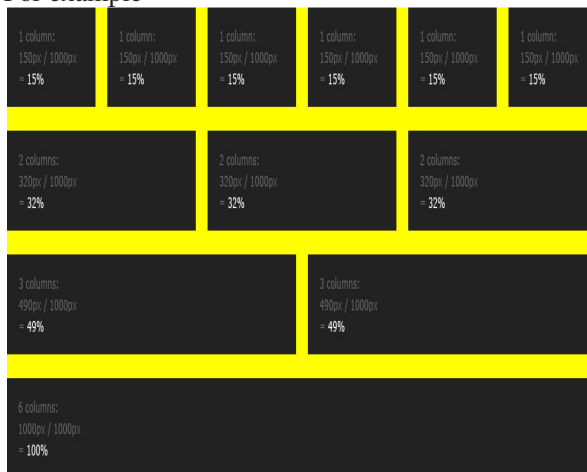
2.1.1 Apply “Mobile First”

Create a future friendly design not fixed pixel width interface keeping “Mobile First” interface to Large Screen. For Feature Phone just use vertical design with minimal use of images.

2.1.2 Grid System Design

As possible as use a grid system like 12 columns or 24 columns grid and arrange the elements as per the grid. A robust grid system helps to organize elements in very efficient manner.

For example



Source:

<http://cdn.harmonyapp.com/assets/4f199548dabe9d54ad00d558/articlesresponsivegrid01.jpg>

Navigation changes, reflowing columns, call to actions, branding, and white space changes all evolve the grid system design.

2.1.3 Identify Responsive Components

Identify all common elements or components that need to be on Mobile and design and organize them first for Mobile and then for large screen.

For Example: On desktop site we have Drop down Navigation and on Mobile site we have simple sliding navigation. By using some tweaks in CSS based on media queries we can make it Responsive with same HTML markup. No need to server two different components.

2.1.4 Identify Adaptive Components

Identify all elements or components that need to be on desktop but not on mobile or have different visual design and functionality so that during the development developer can easily take the decision to apply right approach and again that should be “Mobile First”.

For Example: One desktop site we have to show a Big five Images Carousel but on mobile site we have a small promo image. Here Carousel component can serve only to Desktop based on server side detection and for mobile we can serve Promo Image instead of Carousel.

3. Development Process

3.1.1 Identify Browser Matrix

Identify all supported browsers list and validate how well your browser supports the latest technology standard and related specifications. The matrix will give you a clear picture to take care of the use of fallback mechanisms.

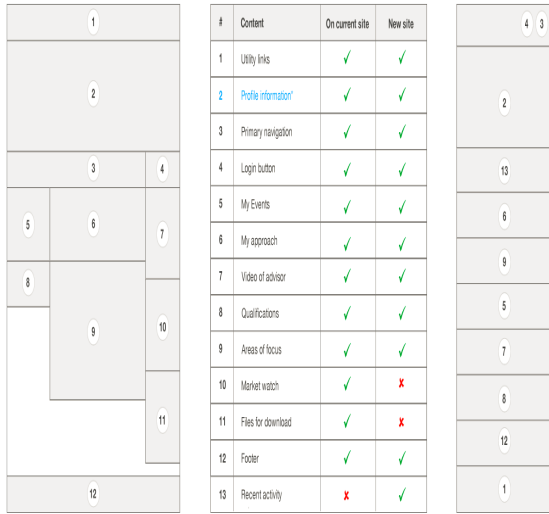
3.1.2 Identify Device Matrix

Identify all supported devices list and their browsers to identify viewport breakpoints and technology support. That will help to make a decision on URL redirection for Feature Phones.

3.1.3 Make Content Strategy

Content Strategy plays a very noteworthy role and it is top priority task to identify how well your devices

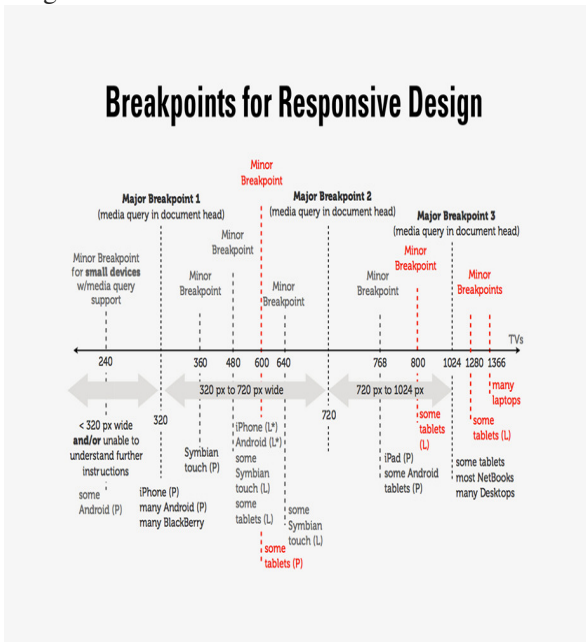
can display the same content with different layouts on different devices.



Content Strategy Diagram between Existing Desktop Site and New Mobile Site

3.1.4 Identify Breakpoints

Breakpoints help us to make front end development architecture in terms of use of CSS, HTML and Images.



Legends Used: **P** – Portrait Mode, **L**- Landscape mode

Source: <http://www.thesteveropergroup.com/blog/wp-content/uploads/2013/04/Breakpoints1.jpg>

3.1.5 Identify Platform/ CMS limitations

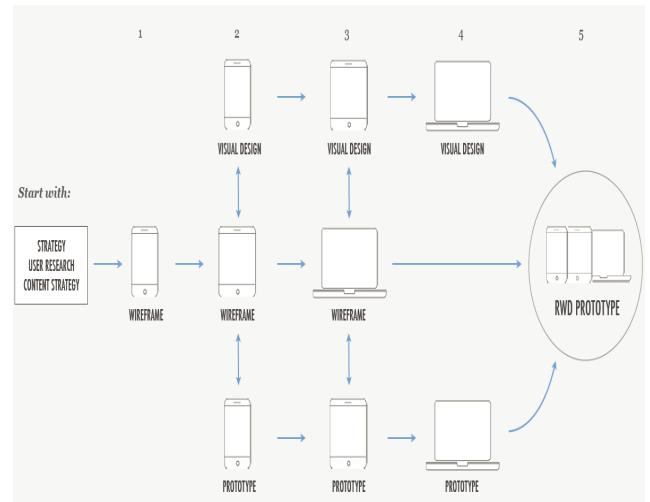
Any limitations of a CMS or Platform should be in place before development so that Tech and Front End team can make a collaborative strategy to display content.

3.1.6 URL Redirection Approach

Mainly this approach should only be considered for Feature Phone support. Where a server side device detection approach can easily redirects the user to m.dot website.

3.1.7 Identify Responsive/ Adaptive Components

If list of Responsive/ Adaptive components has not been provided by UX team. Tech and Front End team need to identify all components and make a strategy for development.



An ideal design and development process

3.1.8 Front End Architecture

There are the 3 technical ingredients for responsive web design

3.1.8.1 FLUID GRIDS

Based on the designer's grid system, we can opt for a fluid grid system.

Fluid grids respond when a layout is squeezed onto a tiny mobile device or stretched, all elements resize proportionally.

We have so many CSS grid system available over internet and free to use.

For example: Twitter Bootstrap, Foundation.js, Skeleton.js, 960.gs etc.

All grid system provides a breakpoint, which is a combination of resolution, screen and device at which your base design starts to break, for instance a 12-column design that works well at 960 pixels would break at 420 pixels.

3.1.8.2 FLEXIBLE MEDIA

Flexible media consist of usage of Image Handling, video management, navigation approach on different devices. So many approaches are there to server the media based on device context. As we are focusing on server side adaption that is perfect approach to serve flexible media.

3.1.8.3 MEDIA QUERIES

Responsive Design only relies on Media Queries, it is a conditional application of css styles based on a device and resolution context. We can define media queries guidelines and standards to adapt the css downward or upward based on our breakpoints and device matrix.

3.1.8.4 HTML/ JS / CSS Framework and Guidelines

This decision is solely depends upon the project requirement like identification of JS libraries, HTML templates and CSS framework. However we should go for latest technologies like **HTML5 and CSS3** with all fall back mechanism to support older browsers.

For JavaScript, we have so many mobile ready frameworks and libraries like

- JQuery
 - JQuery Mobile
 - Sencha Touch
- And many more

For HTML5 and CSS, there are very popular readymade templates like

- HTML5 Boilerplate
- HTML5 Mobile Boilerplate
- Twitter Bootstrap etc.

Do not forget to follow W3C recommendations for new features support and web standards.

4. Non-Functional Requirements

While making the solution architecture we have to close all NFRs given below that should be laid down in the beginning of the project

User Agent Detection
URL Requirement
Responsive Behavior
Content Management
URL Redirection
Social Media Integration
Web Analytics
Performance
SEO

5. CONCLUSIONS

The Conclusion of this document is that we have to take care of various techniques and apply best practices in order to server feature phones and smart phones separately. Technology is moving very fast and it would be very easy for to serve contextual data to user experience to mobile applications.

REFERENCES

- [1] <http://developer.nokia.com/Resources/Library/Web/#!nokia-browsers/common-elements-of-nokia-browsers/device-and-browser-detection.html>
- [2] <http://www.thesteveopergroup.com>
- [3] <http://www.w3.org/TR/mobile-bp/#OneWeb>.

Prepared By:	Anoop Kumar Gupta
Creation Date:	August 09, 2013
Last Update:	August 13, 2013
Version:	0.4
Author	Anoop Kumar Gupta