2012

Testing Mobile Web Applications for W3C Best Practice Compliance

Sean Mee
Technological University Dublin

Testing Mobile Web Applications for W3C Best Practice Compliance

Sean Mee

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
M.Sc. in Computing (Information Technology)

May 2012

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Information Technology), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the rest of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: ______________________________

Date: ***14 May 2012***

1 ABSTRACT

Adherence to best practices and standards when developing mobile web applications is important to achieving a quality outcome. As smartphones and tablet PCs continue to proliferate in the consumer electronics market, businesses and individuals are increasingly turning from the native application paradigm to HTML 5-based web applications as a means of software development and distribution. With an ever-increasing reliance by users on the correct functioning of such applications, the requirement for stringent and comprehensive quality assurance measures is also brought sharply into focus.

This research investigates the increasing trend towards mobile web application development in the mobile software domain, and assesses the requirement for an automated approach to best practice validation testing for mobile web applications. Contemporary approaches to automated web application testing are examined, with particular emphasis on issues relating to mobile web application tests. The individual guidelines proposed by the W3C Mobile Web Application Best Practices are analysed and where applicable automated conformance tests are implemented in a customised testing tool. A range of mobile web applications are tested using this tool in order to examine the extent to which implementation of the tested-for guidelines is detected. Automated tests were successfully implemented in respect of nearly 60% of the best practices.

Key words: *mobile web applications, mobile browsers, W3C, best practices, web standards, conformance testing*

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my thesis supervisor, Dr Susan McKeever. This dissertation would not have been possible without her incisive feedback, sound advice and unwavering encouragement. While I could not have completed this project without her support, any errors or omissions within this work are entirely of my own making. I would also like to express my appreciation to the faculty and staff, too numerous to mention individually, who have been wonderfully supportive to me throughout the entirety of my studies at the Dublin Institute of Technology.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

When developing mobile web applications, adherence to standards and best practice is very important to ensure that a quality outcome is achieved. This research investigates the extent to which mobile web application best practice, as outlined by the W3C guidelines, can be quantified and automated.

One of the significant challenges that this research must address is determining the extent to which each of the guidelines can be automated and implemented. For example, ensuring the appropriate use of the meta viewport element to identify the correct screen size is straightforward in its implementation, whereas assessing the extent to which consistency of application state between different devices is achieved is somewhat less straightforward.

Existing research has proposed a number of testing schemas for web applications and highlights the requirements for robust testing procedures for mobile web apps in particular, given the extensive range of devices and operating systems upon which they may be deployed.

This research focuses on identifying optimal methods to assess each of the best practice guidelines, through the development of a customized mobile web app testing suite. Where the implementation level of a particular guideline cannot be assessed using an automated test, alternative methods of assessment are investigated.

1.1 Background

Web Applications, as distinct from native applications that are designed to be compiled and run on a specific computing platform, represent one of the fastest growing classes of consumer software in recent years (Elbaum, Karre & Rothermel, 2009). Indeed, many companies have adopted Web Applications, which run solely within a browser environment, to support 'mission-critical' corporate and e-Commerce objectives, as well as scientific and health applications. However, as business and consumers become more reliant on Web Application technologies, the quality and reliability of these applications becomes ever more important (Chien-Hung et al, 2000). As Heiatt and Mee (2002) point out, the speed of web application development can sometimes be to the prejudice and detriment of complete and satisfactory quality assurance measures. An effective strategy to ensure the quality of web applications is to implement a comprehensive, yet efficient testing regime.

A growing subset within the web application domain is that of mobile web applications, designed to run on the browser of an Internet-enabled mobile device such as a smartphone, Personal Digital Assistance (PDA) or tablet Personal Computer (PC). The International Data Corporation (IDC) suggests in a 2011 report that smartphone shipments have increased by 87.2% year-on-year, with very strong growth predicted to continue into 2012. Cortimiglia, Ghezzi and Renga (2011) suggest that 40% of Americans currently access the Internet through their mobile phones. Despite, or perhaps because of the dramatic proliferation of mobile web users, and a corresponding increase in the number of mobile web applications which are in circulation, some commentators have suggested that the "fast paced Web Application development culture" has led to established software engineering techniques and principles often being less than fully implemented (Heiatt and Mee, 2002).

In an attempt to address the challenges posed by this "fast paced" development environment, much research has been conducted into optimal methods of developing and testing mobile web applications. The classic testing methodology for Web Applications consists of generating HTTP requests, sending these to the website under test and then analyzing the results (Bruns, Kornstadt and Wichmann, 2009). The current trend however of implementing "Rich Internet Applications", based around Asynchronous JavaScript and XML (AJAX) technologies has left this classic approach somewhat unsuited for comprehensive testing of Web 2.0 applications. Accordingly, several newer testing models, tailored specifically for mobile web applications have been proposed. Zhang and Xu (2008) suggest an optimal approach may be to customize existing testing tools and extend them to form a "mobile agent-based tool". The mobile agent in this context refers to using user agent settings to simulate a mobile device in a desktop testing environment. Qian, Miao and Zeng (2008) point out that due to the complexities associated with user navigation through a web application, a structured approach to developing "test paths" along the navigational tree of the application must be first instituted. They propose an algorithm to generate Page Flow Diagrams (PFD), which maps the hyperlinks between individual pages within a web application. The difficulties associated with automated navigation of dynamic web applications are also addressed by research conducted by Alshahwan et al (2009). This research proposes a framework for the functional testing of web applications through the use of a "crawler" to enumerate and download the various pages comprising the application, extracting the forms contained therein and generating inputs to test each form, thereby validating the navigation and user input functions of the application. The testing considerations mentioned here, while discussed primarily in the context of functional testing, also offer insight into appropriate potential conformance testing approaches.

While research into the field of mobile web application testing continues to progress, it is useful for developers and end-users to have a standard against which to base tests and guide development. The World Wide Web Consortium (W3C) Mobile Web Initiative has seen the Mobile Web Best Practices Working Group deliver its "W3C Recommendation of Mobile Web Application Best Practices" in December 2010. These 32 best practice guidelines provide recommendations "designed to facilitate development and delivery of Web applications on Mobile Devices" (http://www.w3.org/TR/mwabp/). While there are other independently developed web application quality assurance frameworks and best practices, such as Yahoo YSlow or Twitter Bootstrap, these approaches are not recognized as web standards to a comparable extent as the W3C Best Practices are.

1.2 Research Problem

As Internet enabled mobile devices have continued to proliferate in the consumer electronics market, many organizations have produced mobile web applications to support their operations/marketing objectives. Due to the requirements for cross-platform compatibility in such applications, given the range of different handsets/devices, as well as the unique interface methods that apply to such instruments, a requirement for developers to adhere to best practice is clearly identified. The question arises as to whether an objective measure of the implementation of such standards can be developed.

This research aims to address the gap that exists between the existence of such best practices and actually assessing the compliance of mobile web applications with these standards. This will be achieved by developing a series of generic test cases to assess each of the 32 Mobile Web Application Best Practice guidelines and evaluating the extent to which these tests can then be automated using an appropriate testing environment.

1.3 Intellectual Challenge

The intellectual challenges presented in the course of this dissertation span several domains and subject areas, including;

- Web application design considerations
- Mobile Web applications and mobile native applications
- Software Testing and Quality Assurance
- Best Practice and Web Standards formulation and publication processes
- Automated mobile application testing

These challenges are addressed through the presentation of a comprehensive literature review surrounding research into historical and current software quality assurance approaches and software testing processes, with particular reference to the mobile device context. This literature review is supplemented with independent research conducted in the form of a web application developer survey, as well as machine tests implemented in a Java based testing programme.

1.4 Research Objectives

The following objectives have been identified for this project:

- To explore definitions and concepts surrounding Web Application development and contextualize these concept within the sphere of Software Development and Web Engineering.
- To identify key literature and existing research related to Mobile Web Application testing and development.
- To define and assess the requirement for developers and designers to observe adherence to "best practice".

- To identify assessment criteria which are appropriate to the 32 W3C Best Practice guidelines.
- To design appropriate test cases that effectively assess best practice compliance levels.
- To develop a testing suite capable of automating identified test cases.

1.5 Research Methodology

Software testing is a well-established and critical function of the Software Development Lifecycle, to the extent that it is formalized in ISO 9126 (Standard for Software Engineering Product Quality). The relatively modern field of Mobile Web Application Development does not however readily lend itself to the strict application of traditional software testing methodologies (Nabil, Mosad and Hefny, 2011). While traditional testing approaches may not be wholly appropriate to mobile web applications, the requirement for quality and reliability are no less important in this growing subset of software development. This project investigates the current research into appropriate methodologies and approaches to testing mobile web applications, by conducting a comprehensive review of the relevant literature. Conformance testing for the W3C Mobile Web Application Best Practices is investigated by means of developing test cases based on the techniques and approaches outlined in the literature. Primary research, in the form of developing and evaluating automated tests for the best practice guidelines is conducted using a custom testing suite, developed using Open Source Software. This proposed testing software will be evaluated through the deployment of automated best practice conformance tests against a range of mobile web applications.

1.6 Resources

The resources that are utilized in the conduct of this research project are outlined hereunder:

Access to ACM Digital Library, IEEE Electronic Library and other academic libraries

Access to DIT Library service

Access to W3C Mobile Web Application Best Practice Guidelines

Access to open-source web application testing framework (Selenium 2/ Webdriver)

A suitable IDE to develop testing software (Eclipse IDE)

1.7 Scope and Limitations

This research aims to identify and define the key issues surrounding web application quality factors in general, and testing of Mobile Web Applications in particular. Concepts and approaches in Mobile Web Applications testing will be assessed, with a particular focus on conformance testing for compliance with established standards and best practices.

This study primarily examines the extent to which Mobile Web Applications compliance with W3C Best Practice guidelines can be automated, and proposes test cases and assessment criteria for those guidelines that can be tested for quantitatively. Those best practice guidelines whose implementation cannot be tested for in an automated fashion are also considered, and alternative assessment criteria proposed. Best practice conformance testing in this context focuses on client-side processes only. Server-side testing of mobile web applications is outside the scope of this study. This study will focus exclusively on mobile web applications as they pertain to smartphones and tablet devices, and will not consider internet-enabled feature phones. While this study will examine aspects of functional testing in the literature review, the primary purpose of this research is in assessing automated conformance testing of web applications.

1.8 Overview of Dissertation

Chapter 2 introduces the concepts associated with Web Application development and the characteristics particular to web applications that differentiate this field from traditional software development. It also outlines the web technologies used to design, implement and test web applications. This context within which this study considers Web Applications is outlined within this chapter.

Chapter 3 focuses on the growing subset of web applications which are targeted at mobile devices, and examines how the proliferation of this software has necessitated new approaches to web engineering tailored to the mobile paradigm. Web application development and testing considerations are also explored in this chapter. The characteristics of mobile web applications are contrasted to those of the native web application, with analysis of the different testing approaches necessitated by each. Existing testing methodologies employed as part of the software development life cycle for mobile web applications will also be examined.

Chapter 4 examines Mobile Web Standards and Best Practices, and records how web application quality factors have been proposed to provide a framework for developers to achieve high quality outcomes in terms of mobile web applications. The particular challenges associated with testing dynamically generated web content will be detailed. This chapter will also outline each of the 32 W3C Mobile Web Application Best Practice guidelines. A survey of mobile web application developers is carried out in this chapter that identifies current developer attitudes and approaches to mobile web application quality assurance and testing.

Having outlined the individual Best Practices, Chapter 5 proposes assessment criteria by which the implementation levels of the guidelines can be measured and will examine the Best Practices in terms of those which lend themselves to automated testing and those which require a qualitative assessment in terms of their implementation. Where appropriate, these assessment criteria are implemented as automated test cases utilized in the test software developed as part of this project.

The experimentation and evaluation of the automated test cases proposed in Chapter 5 are discussed in Chapter 6. In this section a wide range of mobile web applications are assessed using the software tests proposed earlier in the dissertation. The automated test tool, developed in Java as part of this project, is used to analyze and assess the best practice compliance levels of mobile web applications.

The study in its entirety is discussed and overall conclusions offered in Chapter 7, which will propose the contribution this research offers to the mobile web application testing body of knowledge. Future research and work in the field is also suggested in this chapter.

2 WEB APPLICATIONS

2.1 Introduction

Web Applications are software applications which are provided entirely through a client's web browser, or which are interacted with through a network (Parveen, Tilly & Gonzalez, 2007). A web application is differentiated from a website in that a user's interaction with a web application involves more than merely "navigational requests, including some form of data that needs further decomposition and analysis to be served" (Elbaum, Karre & Rothermel, 2009). This chapter discusses the advantages and disadvantages of web applications when compared with native apps, considered from an end user and business perspective. The general concepts that distinguish and define web application explored in this chapter are then considered and applied more specifically to the mobile platform in Chapter 3.

A web application represents a more dynamic mode of interaction between the user and device, which takes user input and subjects it to some process to achieve an output. The purchasing interfaces of e-commerce sites like Amazon.com are a common form of web application. Software that is provided as a native application executed on the underlying operating system remains the predominant means of software development and distribution. However, web applications have continued to make significant gains in terms of developer preference and user acceptance in the last number of years (GIA, 2010) The increases in Web Application market share are attributable in part to the manner in which this software production and consumption paradigm presents "convenient and inexpensive ways to provide product information and services on-line to anyone at any time" (Liu et al, 2000). Qian, Miao & Zeng (2008) observe that web applications are currently one of "the fast growing classes of system software" in usage, and that these applications are deployed widely in critical business, scientific and medical transactions. Several commentators suggest that the deployment of web applications as a software solution will continue to grow in popularity, due to the aligned demands of both consumers and businesses.

2.1.1 Business Case for Web Applications

Native applications generally execute faster than web applications and typically benefit from direct access to the underlying device hardware, such as GPS or other geolocation capabilities, while web applications generally do not. Firms that have nonetheless forgone production of a native application and chosen to offer only a web application to their customers have suggested several compelling reasons for doing so. In June 2011, the Financial Times withdrew its native application from Apple's iTunes Store and offered its customers a web application as an alternative. This circumvented the App Store model, rendering wholly redundant its role as intermediary in the distribution of software content (Golson, 2011). This strategy to shift from a native application to a web application proved to be a successful one for the Financial Times, with FT.com Managing Director Rob Grimshaw commenting that the HTML 5 based web application, depicted in fig 1.1, has drawn significantly more traffic than the native application previously achieved (Reuters, 2011). Bypassing the App Store as a distribution channel also yields financial benefits in that the profit sharing business model it relies on is effectively by-passed, usage/subscriber data is held solely by the content provider and perhaps most significantly, it facilitates cross-compatibility on multiple platforms in that the applications are run solely in a web browser environment. This cross-compatibility allows developers to simultaneously offer software to users of iPhone, Android and any other smart phone which uses a standards compliant web browser, without the costly and time-consuming requirement to adapt the code base for the individual platforms targeted.

Figure 2.1 FT.com web application (www.ft.com**, 2012)**

A 2010 Global Intelligence Alliance survey of firms (GIA, 2010) which provided either native or web applications to consumers found that while 44% of companies offered a native application only, 35% offered both native and web based applications, with 25% offering a web application only. When asked about the business case to develop only a web application, a majority of respondents pointed to the economies associated with developing a single application interface that was compatible across a number of platforms. Fig 1.2 highlights the other main reasons offered by those surveyed for deciding on exclusively providing a web application. Significantly the preference for circumventing the App Store distribution model is also cited as one of the top reasons for preferring web applications.

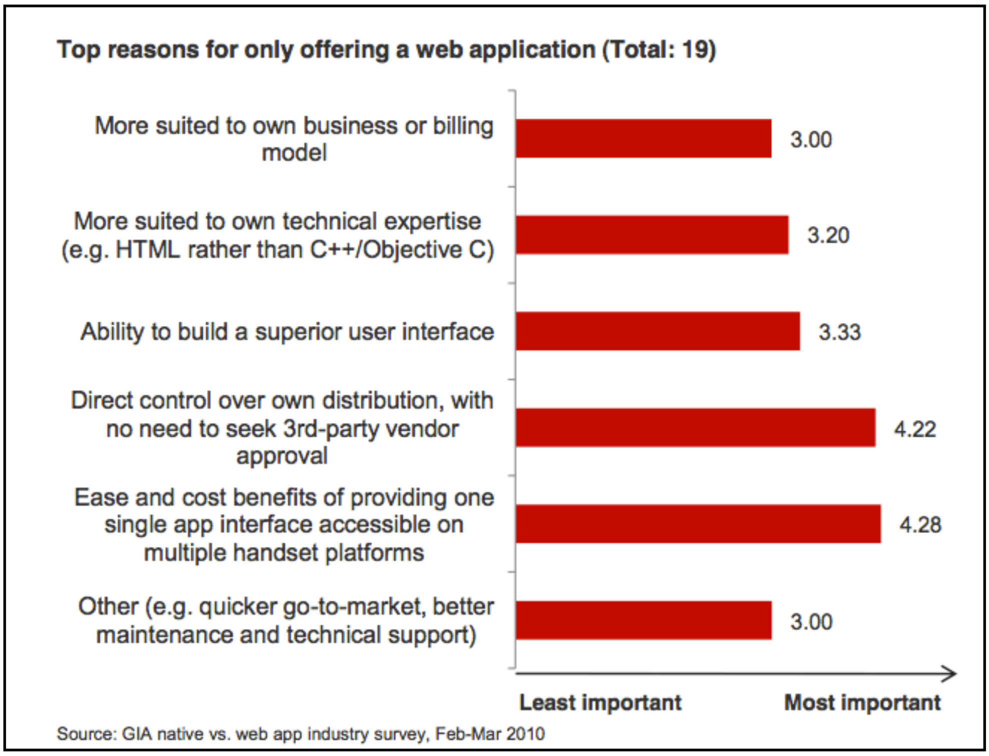

Figure 2.2 Top reasons for only offering a web application (GIA industry survey, 2010)

2.1.2 Users' Case for Web Applications

Web applications often provide a convenient and cost effective software solution to end-users, particularly in the domains of web based email, social networking and e-commerce. Gmail, Yahoo! Mail and Hotmail are widely used email web applications, with a combined share of 28.7% of the overall email client market (Campaign Monitor, 2011). Mesbah and Prassad (2011) even assert that "web applications pervade all aspects of human activity today" across an increasingly broad set of business and consumer functions. A prominent example of the growing ubiquity of web applications, Google offers a wide range of web applications in addition to webmail, in the form of the Google Apps suite, including web-based versions of traditional office information systems software. Wang & Jin (2010) note the significant advantages this Software-as-a-Service (SaaS) model affords the consumer, allowing "users spend less to build, to maintain, to use and to invest in software". Pemberton (2006) points out that a major advantage of web applications over native applications to the end user is that "everyone has always got the most recent version".

Figure 2.3 "Google Apps" Web Application suite (www.google.com/apps, 2012)

The advantages which web applications provide to users over traditional native applications, combined with the compelling business case for firms to embrace this software development and distribution model have proven to be significant drivers for advancing research and development in web application design and testing. As web applications continue to proliferate and play an increasingly important role in delivering critical software solutions, Qian, Miao & Zeng (2008) warn that poorly implemented and inadequately tested web applications can have "far reaching consequences on business, economies, scientific progress, [and] health".

2.2 Characteristics of Web Applications

Web applications differ significantly from native applications in terms of the planning, implementation and testing methodologies employed in their software development lifecycle. These differences arise from the unique characteristics of web applications which differentiate them from traditional software models. Di Lucca and Fasolino (2006) suggest that web applications display the following defining characteristics:

- Concurrently accessed by a wide number of users distributed globally.
- Run on heterogeneous execution environments (different operating systems, web browsers, etc.).
- Comprise several heterogeneous software components (different programming/scripting languages).
- Dynamically invoke different components at run time based on user inputs.

Al-Salem and Samaha (2007) differentiate web application development from traditional software development based on the significantly shorter development lifecycle, diverse and unknown end-users, integration with backend databases and the high visibility/immediate web presence which web applications deliver. The authors propose that these differences are of a significant magnitude to justify customized requirements gathering and testing methodologies instead of relying on traditional software development approaches. This view is corroborated by Li and Miao (2008), who suggest that due to the "different hardware and software platforms" and "several different programming languages" used in the production of web applications, new approaches to modelling and testing web applications should be developed.

As previously alluded to, web applications differ from static webpages in that user interaction is not restricted to simply navigating through the hyperlinks in a website. Web applications provide dynamically generated content, which is delivered based on the users' inputs to the system. One of the most prevalent methods by which the dynamic content is produced is through the use of AJAX (Asynchronous Javascript and XML) technology. However, while AJAX offers the end user a very high degree of interactivity with the web application, it also presents a number of challenges to developers and testers (Mesbah, Bozdag & van Duersen, 2008).

2.3 Web Technologies

The traditional model by which users interact with websites involves a HTTP Get request being sent from the client to the web server hosting the website, which then serves the requested page. Subsequent requests from the client to the server follow a synchronous "click-and-wait" pattern (Matthews et al, 1997). This method of client-server interaction does not lend itself well to highly interactive and responsive web applications such as those mentioned above.

The AJAX enabled Internet model however facilitates asynchronous communication between the client and server, allowing parts of the user interface presented in the web application to be updated as the user interacts with page elements. The “AJAX engine” refers to the code executed on the client side that facilitates this asynchronous communication with the web server as shown in Figure 1.4(Richie, 2007).

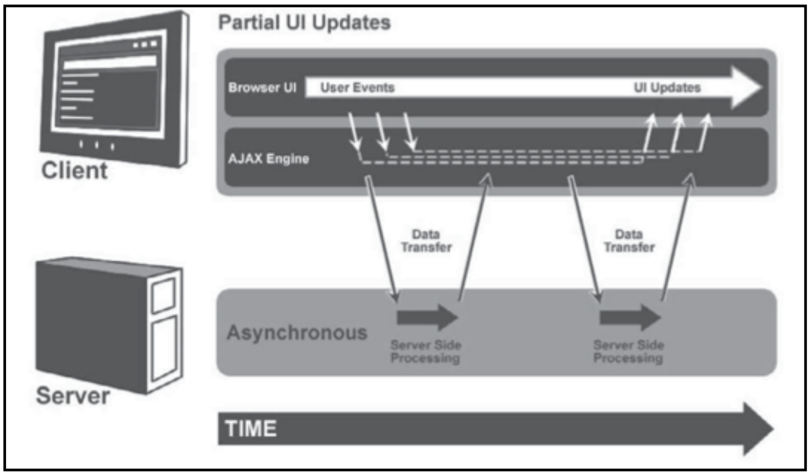

Figure 2.4 AJAX model (Richie, 2007)

While AJAX driven web applications can deliver highly interactive and responsive user interfaces, this web technology has also created a number of challenges to traditional methods of web testing using HTML validation applications (Artzi et al, 2010) and page coverage using “web crawling” (Alshahwan et al, 2009). Mesbah, Bozdag & van Duersen (2008) highlight three particular issues which AJAX enabled web applications present to developers:

- Difficulties in ensuring AJAX sites are indexed by search engines due to dynamically generated content.
- Difficulties in automatically testing dynamically generated user interfaces
- Difficulties in assessing whether “all states of an AJAX site meet certain accessibility requirements”

Web Engineering, defined as “the systematic, structured and quantifiable application of methodological proposals to the development, evaluation and maintenance of web application", has emerged as a software engineering approach designed to address these challenges (Escalona et al, 2006).

2.4 Web Application Testing

The testing of web applications is assessed as being significantly more complex in comparison with traditional native applications, due to the "heterogeneity, distribution, concurrence and platform independence" characteristics exhibited by web applications (Li & Miao, 2008). Established techniques for web based testing are also proving increasingly inadequate to cater for rich content web applications, commonly referred to as Web 2.0 applications, due to their reliance on "static analysis" and HTML validation (Huang and Chen, 2006). The requirement for thorough testing for web applications is no less pronounced however than in other software development domains, particularly in the context of the critical business and corporate functions performed by such applications.

2.4.1 Requirement for web application testing

Given the growing reliance by many users on correctly and consistently functioning web applications, it is essential that comprehensive testing is implemented during the development lifecycle. Sprenkle and Pollock (2007) underline this requirement when they note that the critical dependency of many users on this software model "motivates developing automated, accurate validation approaches, specialized for web applications." While much research has been conducted into web testing and HTML validation in a non AJAX environment, several modern approaches to automated coverage and interaction simulation in respect of Web 2.0 applications have been also been proposed in the literature.

2.4.2 Web Application coverage testing approaches

Montoto et al (2011) assert that an essential element of web application testing is the capability to automate the process of browsing through the website. This role was traditionally performed through the employment of a web crawler, which analyses the hyperlinks within a website and has as its output a linked map of all the possible execution paths through the site. Andrews et al (2010) posit that the use of a Finite State Machine (FSM), where each "screen" represents a state, to model the behaviour of a web application may be a useful method of providing maximum coverage for possible user paths through the application. This FSM approach is also propounded by Mesbah and Prassad (2011), whereby the finite state machine representation of the web application is generated through a crawler implementation which is capable of detecting and executing "doorways", i.e. user-clickable interface elements. The authors acknowledge however that a major difficulty in producing a workable FSM is the large number of allowable inputs to a web application, which can result in a "state space explosion".

In the case of modern AJAX enabled web applications, Alshahwan et al (2009) also observe that the traditional method of website coverage, crawling by hyperlink analysis, is unsuitable due to the inability of this approach to automatically interact with the application through the user interface. This method of crawling by analysis of static HTML links within a web application therefore does not deliver an adequate testing path through the possible user interactions with a Web 2.0 application. This is as a result of dynamic updating of page elements through AJAX without the use of embedded links in the page source, and any automated testing model for web applications must address this characteristic of their user interaction.

2.4.3 Testing AJAX based web applications

Mesbah and van Duersen (2009) point out the weakness of a static HTML analysis approach to web application testing, observing that while useful for validating correct use of syntax, HTML analysis does not adequately account for dynamic content behaviour. The authors propose a method of constructing a State Machine of the web application through the deployment of a crawler that identifies and activates "clickable" screen elements. This approach starts with the root state, essentially the start-screen of the web application, and incrementally adds additional states as the application is "crawled and state changes are analysed".

A further consideration for the automated testing of web applications is to define how the specified test cases are to be assessed. Having identified an approach to obtain coverage of the application, perhaps through generating a FSM representation of its possible states, the tester must then evaluate how the application performs against a specified benchmark. Sprenkle, Hill and Pollock (2007) expound the use of HTML based “oracle comparators” to provide an appropriate baseline of expected versus actual behaviour of the system under test. The expected response of a web application to user input and its subsequent output, called the “gold standard”, is compared to the actual behaviour of the web application.

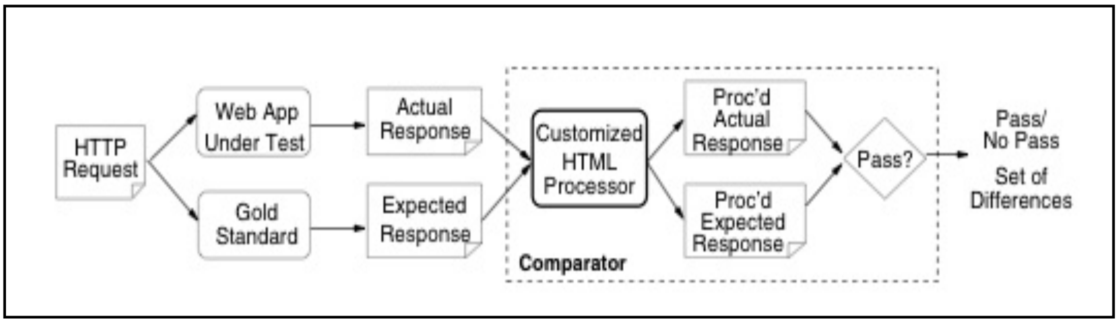

Figure 2.5 HTML Oracle Comparator validation (Sprenkle, Hill & Pollock, 2007)

When implementing an oracle comparator based validation approach however, the automated testing system must adequately account for dynamically updated content within the web application.

2.5 Conclusion

The perceived advantages of web applications as a software development and distribution model, to both business and consumers, have seen significant increases in both their quantity and scope over recent years. This upward growth trend is likely to continue as this software model continues to compete aggressively with traditional native application deployment paradigms. This increasing reliance on web applications has also highlighted a requirement for stringent quality control and testing, which must be conducted in a manner that accommodates the unique characteristics of web applications over native apps. An effective web application testing solution for Web 2.0 applications incorporates an efficient method of traversing and enumerating dynamically generated HTML content, through simulating a user’s interaction with the page elements. It will also ensure a well-planned range of oracle comparators are employed in order to validate the correctness of server responses to user input. A growing subset of the web application domain is the mobile web application, typically deployed on a smartphone or other modern Internet enabled mobile devices. Chapter 3 discusses this rapidly growing area within the wider web application sphere, and identifies the quality and testing issues that are associated with the development of mobile web applications.

3 MOBILE WEB APPLICATIONS

3.1 Introduction

In the last chapter, the general characteristics of web applications, as well as business and technical issues surrounding their implementation were discussed. This chapter examines a specific subset of the web application domain that deals with the appropriate methods to serve and test web applications on mobile devices. Internet enabled mobile devices, such as smartphones, Personal Digital Assistants (PDAs) and tablet Personal Computers (PCs), are rapidly becoming ubiquitous in modern society, due in large part to the continuing and fast-paced advances in mobile communication technology, aligned with dramatic production and sales efficiencies (Murugesan and Venkatakrishnan, 2005). The range of functions that such devices perform, in addition to voice telecommunication in the case of smartphones, has become increasingly diverse as the technology has matured and consumer demand developed.

Modern smartphones and tablet PCs provide a platform which provides many of the same software functions as a traditional desktop PC, such as office productivity, e-commerce applications, social networking and scientific tools, combined with the added benefit of device portability (Aggarwal and Yu, 2011). However, users of mobile devices often utilize this technology in a drastically different manner when compared with traditional PC utilization. Garofalakis and Stefanis (2008) highlight the "more immediate and goal-directed intentions" of mobile device users compared to desktop web users. Additionally restrictions on user input due to a small or absent physical keyboard, and restrictions on device output due to a relatively smaller screen place additional constraints on the use of mobile web applications in the place of desktop based web applications.

Despite these constraints, consumer demand for mobile software applications continues to experience strong growth and many businesses now provide either native or web-based applications for mobile devices. The reliance that many users now place on correctly functioning mobile software applications across a broad spectrum of usage scenarios motivates a structured and comprehensive approach to quality assurance and testing (Qian, Miao & Zeng, 2008).

3.2 Proliferation of Mobile Web Technology

Global smartphone and tablet device ownership has increased dramatically in most market segments over the past 12 months, with a Gartner (2011) study highlighting a 57% increase in worldwide smartphone sales to 468 million units from 2010 to 2011. Gartner also predicts that this upward trend will continue into future years, driven by decreasing unit costs and rapid improvements in mobile technology. Figure 3.1 outlines the striking increases in smartphone sales between 2010 and 2012, with a predicted total of 1,104,898,000 units sold by 2015.

Worldwide Mobile Communications Device Open OS Sales to End Users by OS (Thousands of Units)

OS	2010	2011	2012	2015
Symbian	111,577	89,930	32,666	661
Market Share (%)	37.6	19.2	5.2	0.1
Android	67,225	179,873	310,088	539,318
Market Share (%)	22.7	38.5	49.2	48.8
Research In Motion	47,452	62,600	79,335	122,864
Market Share (%)	16.0	13.4	12.6	11.1
iOS	46,598	90,560	118,848	189,924
Market Share (%)	15.7	19.4	18.9	17.2
Microsoft	12,378	26,346	68,156	215,998
Market Share (%)	4.2	5.6	10.8	19.5
Other Operating Systems	11,417.4	18,392.3	21,383.7	36,133.9
Market Share (%)	3.8	3.9	3.4	3.3
Total Market	**296,647**	**467,701**	**630,476**	**1,104,898**

Fig 3.1 Mobile Comms Device Global Sales (Gartner 2011)

A comScore (2012) White Paper reports that smartphone adoption in 2011 has surpassed 40% of the total mobile phone market in the United States, France and Italy and surpassed 50% in the UK and Spain. Their research attributes much of these gains to improved availability of 3G and 4G networks, as well as improvements in device functionality and the "aggressive pricing" strategies of manufacturers and distributors. Figure 3.2 highlights this strong growth in smartphone ownership as a percentage of total mobile phone audience across 7 major market segments, with each displaying significant increases in adoption within the 12-month timeframe between December 2010 and December 2011.

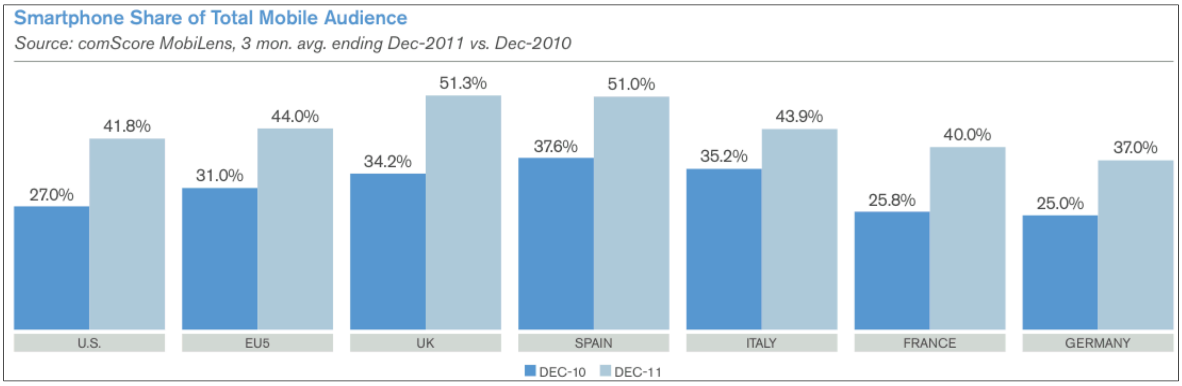

Fig 3.2 Smartphone share of total mobile audience (comScore 2012)

As users continue to migrate to the smartphone/tablet PC model of computing, the consumer demand for both personal and business related applications is also set to increase in tandem. In order to effectively meet this demand, mobile application developers must take cognizance of the essential differences in processor capabilities, user interaction methodologies and application usage patterns between traditional desktop and mobile computing platforms.

3.3 Characteristics and Limitations of Mobile Web Applications

Modern Internet enabled mobile devices provide users with a relatively powerful computing platform, with the advantages of portability, geo-location services and connectivity to the World Wide Web. The platform is also a very extensible one, in that developers can produce applications that leverage the sensory and computational capabilities of these devices to accomplish a wide variety of tasks. These smartphones/tablet PCs are not however without their drawbacks and limitations. Wessels, Purvis and Rahman (2011) highlight the following characteristics that limit the capabilities of such devices:

- *Connectivity* - data costs for 3G internet access and sub-optimal connection speed can have a negative impact on data-intensive web applications
- *Graphics and Hardware* - limited resources in comparison with full featured desktop computers
- *Screen Real Estate* - a smaller screen implies that traditional presentation techniques may not be appropriate to mobile
- *Input/Output functionality* - lack of mouse and keyboard affects the method of user interaction with web applications

- *Software Support* - different devices utilize varying mobile operating systems and web browser support

The fifth point, which emphasizes the complexities associated with varying degrees of browser support, is largely mitigated through the implementation of standards-compliant browsers on modern mobile devices. Issues with screen real estate present enormous challenges to developers of native applications for Android device, requiring them to consider a wide range of screen sizes and resolutions. Effective mobile web application implementations seek to ameliorate the other factors using code optimization to minimize HTTP requests between the client and server, as well as media compression techniques to reduce page load times. Such optimization can largely counteract the negative effects of unreliable or expensive connectivity associated with mobile devices. However it is also important for developers to employ practices that minimize the negative effects on user experience of the other limiting characteristics of this platform. This can be achieved through interface design based on touch gesture interaction and by limiting the amount of features presented to a user on the small screen (Nielsen, 2009).

The limitations and characteristics of mobile computing devices in terms of performance and interaction constraints apply to both native and web applications, though currently native applications tend to perform better in domains which require significant graphics processing. While native applications, such as those distributed through the Apple App Store or Google Play (formerly Android Market) continue to enjoy a larger market share in comparison to mobile web applications, the web application paradigm is rapidly shortening this gap (GIA, 2010).

3.3.1 Native Apps vs Mobile Web Applications

The introduction by the World Wide Consortium (W3C) of HTML 5, though not yet a fully finalized standard (McDonald, 2011), has contributed significantly to cross-platform compatibility and convenient multimedia access in mobile web applications. Indeed, one of the primary goals of this latest iteration of web markup language is to "move the web away from proprietary technologies", by providing developers with a unified standard for web engineering which provides enhanced functionality without the use of proprietary audio/video encoding or typefaces (Vaughan-Nichols, 2010).

HTML 5 offers developers built-in functionality such as location-based services, video embedding, advanced graphics rendering and off-line capabilities as standard, and is already in use by some major web companies, including YouTube and Google (Parr, 2010).

The cross-platform capabilities associated with mobile web applications also contribute to a "build once, deploy anywhere" paradigm of software development which is not present in the native application model. Charland and LeRoux (2011) point out that while web applications can be effectively developed for a multitude of browsers and hardware platform using single code base, native application distributors require capabilities in a multitude of proprietary SDKs and several programming languages, such as Java for Android application development and Objective-C in the case of iPhone/iPad programming. Figure 3.3 highlights the various skill-sets that are required in order to target nine of the most prominent mobile operating systems. Clearly there are economies to be achieved for developers in terms of localization and code-maintenance if they adopt the mobile web application development model.

Mobile OS Type	Skill Set Require
Apple iOS	C, Objective C
Google Android	Java (Harmony flavored, Dalvik VM)
RIM BlackBerry	Java (J2ME flavored)
Symbian	C, C++, Python, HTML/CSS/JS
Windows Mobile	.NET
Window 7 Phone	.NET
HP Palm webOS	HTML/CSS/JS
MeeGo	C, C++, HTML/CSS/JS
Samsung bada	C++

Fig 3.3 Skill sets required for nine mobile OSs (Charland and LeRoux, 2011)

Despite the advantages associated with mobile web applications over native applications, the native app model continues to outperform in terms of user engagement and retention. This is illustrated in Figure 3.4, with a majority of industry respondents to a 2010 survey indicating significantly longer usage time by users on native applications.

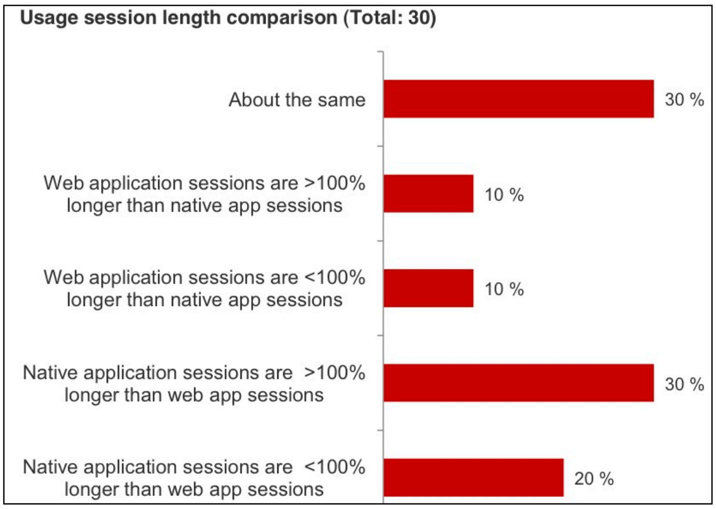

Fig 3.4 Native v Web App usage session comparison (GIA, 2010)

Charland and LeRoux (2011) attribute the continuing dominance of the native application model to the fact that the web "technology stack has not yet reached the level of performance we can attain with native code". The "hybrid" model allows developers to combine the strengths of native applications with the advantages provided by HTML5 in terms of cross-browser compatibility. The "Facebook" application for iPhone and Android is an example of one such hybrid application, which uses the app store model as its distribution mechanism, but is effectively a mobile web application packaged into a native executable (Gibbs, 2011). A major issue which presents native applications with a significant advantage over web applications is their access to device specific features such as accelerometer, some geolocation services and telephony/SMS access (Mims, 2011).

The strengths of native applications in terms of computational performance and their ready access to device specific APIs makes them more appropriate for computationally expensive tasks, such as 3D video games and other graphics-heavy applications. As HTML5 continues to deliver improvements to the capabilities of mobile web applications, as well as ever-improving 3G and mobile web access, this performance gap will be eroded over time. It is expected that by 2013, the majority of native device attributes, such as local file access, messaging and hardware control will be available to HTML5 powered mobile web applications, leading to "user experiences which will rival those of native applications" (GIA, 2010). While native applications currently enjoy the largest market share in terms of software executed on mobile devices, this project focuses primarily on the issues that surround web application implementation and testing on these devices.

3.4 Web Engineering for Mobile

Web development has evolved from a largely static-text based information presentation model to the modern rich Internet applications seen in Web 2.0 applications. This evolution has taken place without a significant alteration to the basic paradigm associated with the client-server model, yet presents challenges in terms of application "scalability, reliability and performance" for mobile developers (Gitzel, Korthaus and Schader, 2007). Deshpande and Hansen (2001) proposed the concept of Web Engineering as the "application of a systematic, disciplined, quantifiable approach to development, operation and maintenance of the web-based applications", and argue the unique characteristics of web applications necessitate a departure from preexisting software development techniques. The authors view Web Engineering as a "forward looking" undertaking which seeks to apply a multidisciplinary approach including elements of computer science, software engineering and multimedia design. Ginige and Murugesan (2001) emphasize that the "subtle differences in the nature and lifecycle of web-based software" necessitate comprehensive testing strategies to ensure web applications are fit-for-purpose and achieve quality standards.

Murugesan and Venkatakrishnan (2005) suggest that the most significant mobile web engineering challenges can be classified under three broad categories:

- Design and development
- Testing and validation
- Deployment and execution

While the design, development, deployment and execution of mobile web applications are vital elements in the software development lifecycle, this research project primarily focuses on the Testing and Validation aspects of web engineering, specifically in proposing an automated approach to conformance testing of web applications for the implementation of best practices and standards.

3.5 Testing Methodologies for Mobile Web Applications

Mansour and Houri (2006) assess web applications as being "complex, ever evolving and rapidly updated software systems", the testing of which is both challenging and critical. The challenges associated with testing derive primarily from inadequacy of traditional tools and techniques in addressing the distinctive design and implementation features of web applications, which are discussed earlier in this chapter and in Chapter 2. The authors also point to instances in which insufficient testing of web applications has had a severe impact on the business operations of the developers, citing the high profile failure of the amazon.com web application in 1998. Inadequate testing failed to identify an error in their e-commerce web application that caused the site to be in-operational for several hours, costing the firm an estimated $400,000 in lost revenue and expense. The case for implementing rigorous, comprehensive and where possible automated testing approaches for web application development is a compelling one.

Parveen, Tilly and Gonzalez (2007) assert however that software testing is "still too often treated as an afterthought" and highlight web applications in particular as prone to being neglected in terms of implementing a comprehensive testing process to ensure compliance with design specifications and quality expectations. Much research into mobile web application testing has been conducted in an attempt to address this perceived lack of focus on quality assurance. In assessing the different categories of web application testing, DiLucca and Fasolino (2006) consider it useful to consider this in the context of functional and non-functional testing. Functional testing, as its name suggests, concerns itself with assessing whether the web application produces output in accordance with its specifications, and the presence of syntactic and semantic errors in the code employed. Non-functional testing covers a much broader array of requirements that the web application should fulfill. DiLucca and Fasolino (2006) further subdivide these requirements into performance, load, stress, compatibility, usability, accessibility and security testing. The categories of performance, compatibility and usability testing fall within the scope of this dissertation, and provide a framework within which this research considers approaches to conformance testing of mobile web applications to established best practice standards.

In terms of designing automated tests which assess the performance, compatibility and usability of mobile web applications, much research highlights the suitability of an 'oracle comparator' based approach. Marin et al (2011) suggest the use of oracles to assess whether a particular test case passes or fails, where the oracle is a 'gold standard' or expected output that the System Under Test (SUT) should also produce if it is operating correctly. These oracle comparators are generally derived from the software specifications, however in the case of highly dynamic web applications it can become difficult to specify an adequate range of oracles.

Marin et al's (2011) proposed approach to developing a testing methodology for "future" web applications is presented in 4 steps. This logical approach consists of firstly specifying a precise theoretical framework used to represent the application. In this case of mobile web applications, the Document Object Model (DOM) which depicts how the individual components or 'nodes' in a web application interact, represents the theoretical framework within which tests can be conducted. The second step involves identifying an appropriate testing methodology to apply. Open source testing tools such as Selenium and Webdriver provide testers with a basic but extensible framework which facilitates DOM traversal within web applications and facilitate the implementation of automated tests through simulating user interactions within a browser (Bruns, Korstadt & Wichmann, 2009). The third step proposed involves defining an empirical framework against which to measure the outputs produced at the conclusion of step two, with the final step requiring implementation of the application-specific tools to administer the tests. Marin's approach to developing a three step approach to web application testing is extended and customized in Chapter 5 to consider conformance testing of mobile web applications. These steps involved in Marin et al's methodology are depicted visually in figure 3.5:

Fig 3.5 Components of testing methodology (Marin et al, 2011)

Sprenkle, Hill and Pollock (2007) point out that while there are a large number of programming and scripting languages used in the development of web applications, HTML remains the standard language by which web content is published and displayed in the browser. They assess HTML oracle comparators, applied to HTML produced either dynamically or statically by web applications, as providing a useful empirical framework by which test cases can be applied.

3.6 Conclusion

This chapter has traced the recent proliferation of smartphone and tablet PC devices in the consumer electronics market and examined how this trend is expected to continue into the future. Rapid and ongoing improvements to mobile Internet connectivity, in terms of pricing and quality of service for both Wi-Fi and 3G/4G are significant drivers of this strong growth. The competitive pricing structures surrounding mobile devices is also a major factor in explaining the smartphone ownership trends, and this factor will continue to work in the consumer's favour as strong competition amongst device manufacturers prevails.

The characteristics and limitations of mobile devices, largely associated with screen real estate and processing capabilities, have motivated producers to embrace new methods of application development and testing. Industry surveys indicate that the heretofore primacy of native applications is under growing threat from the recent improvements in web application capabilities provided by leveraging HTML 5 and implementation of emerging standards and best practices. The relatively modern field of web engineering seeks to provide a multidisciplinary approach to ensuring quality outcomes for mobile web applications, by incorporating aspects of computer science, software engineering and multimedia design. Software testing by its nature is a time-consuming and human capital-intensive process, yet is of paramount importance in validating software quality. In order to minimize the costs associated with software testing, test automation is an attractive proposition for application developers.

Several approaches to quality assurance have been proposed to provide mobile web applications developers with a framework within which to design and develop their software. Chapter 4 examines these various quality factors, and outlines the World Wide Web Consortium (W3C) best practices for mobile web applications. These standards and best practices form the basis for the automated testing approaches that are proposed in Chapters 5 and 6.

4 MOBILE WEB STANDARDS AND BEST PRACTICES

4.1 Introduction

Having discussed the characteristics and limitations of mobile devices that are applicable to mobile web application development in Chapter 3, this chapter endeavors to explore the issues surrounding best practice implementation and quality assurance approaches specific to mobile platforms. This topic is addressed by reference to both established and emergent software quality frameworks and traces the development of the current W3C Mobile Web Application Best Practices through the web standard publication process. A survey of web application developers to canvass attitudes to best practice and quality assurance was also administered as part of this research, the results of which are discussed in this chapter.

The W3C Best Practices are outlined and analyzed in the context of the preceding research conducted in the field of software quality assurance, but also in terms of contemporary work undertaken in other aspects of web standard implementation, including *inter alia* W3C Mobile Web Best Practices and the W3C Web Accessibility Initiative. The analysis of mobile software quality assurance and testing begins first however with an examination of the well established ISO standard for software quality assurance.

4.2 Quality Assurance and Best Practices

The issue of software quality assurance and the preparation of associated best practices and standards is not a recent development in software engineering, nor is it exclusive to web engineering. In development since 1985 and first issued in 1991 by the International Organization for Standardization, ISO 9126 is the primary industry standard for software quality assurance.

The ISO model is "applicable to every kind of software", and specifies six quality characteristics that should be addressed by all software implementations (Behkamal, Kahani and Abkari, 2009). The overall considerations in this software quality framework are Functionality, Reliability, Usability, Efficiency, Maintainability and Portability. These characteristics are further divided into 21 subcategories, which should be manifested in the finished software product. Figure 4.1 outlines these characteristics, and provides an overview of the quality factors proposed in ISO 9126.

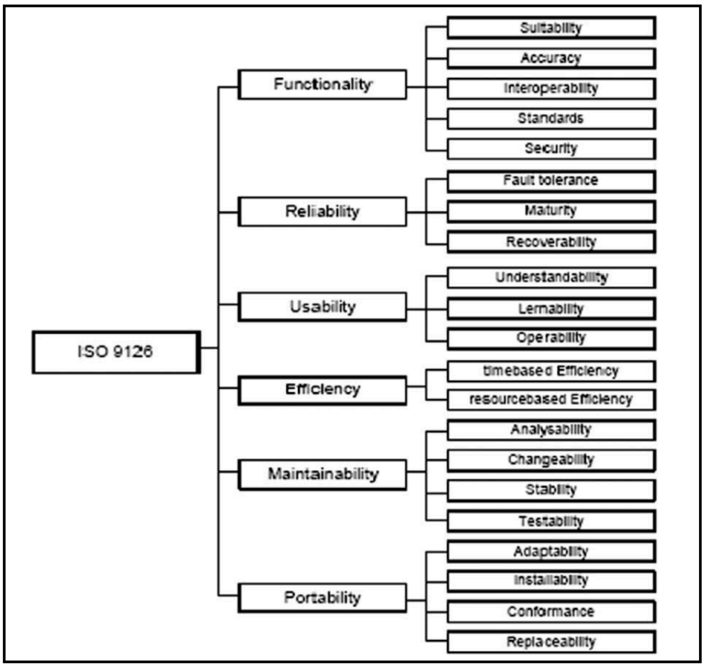

Fig 4.1 ISO 9126 Quality Model Attribute Tree (Behkamal, Kahani and Abkari, 2009)

While this standard provides a useful framework within which web application quality factors can be considered, it remains a generic approach that does not take many of the unique characteristics of web applications into account. This fact is hardly surprising given that web applications were not in general usage, if at all, at the time of the standard's drafting.

Much research has been conducted in recent years into identifying quality assurance approaches for web applications that use ISO 9126 as a template for a customized quality framework. Zulzalil et al (2008) suggest that when considering web application quality factors within the ISO 9126 framework, the factors of Functionality, Usability, Reliability and Efficiency are the most applicable from the end-user's view. Web applications quality factors are also considered from the user's view by Lew et al (2008), when they categorize these factors under the headings of Environment, Experience and Behavior. In this context, Environment refers to the factors associated with the operating system or browser environment in which the application runs, while Experience refers to the factors affecting user-interaction with the system, such as the "Look and Feel" of a web application. Behavior factors arise out of the interactions between Environment and Experience, and include user loyalty and a user community. The authors suggest that the "overall quality of the application" is derived from the

Experience and Behavior factors that the end user perceives. This relationship is depicted graphically in Fig 4.2.

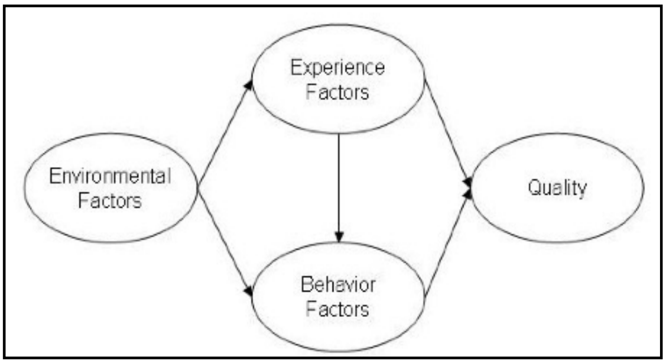

Fig 4.2 Quality Factor model (Lew et al, 2008)

Ultimately, the end user makes the determination of web application quality. A user's perception of application quality is often a subjective judgment call, influenced by the expectations and experience of the individual. Liu, Chen and Zhou (2006) propose a theoretical framework that aims to quantify the factors that influence end user satisfaction with information technology systems. In this framework, the user's expectations affect perceived ease-of-use and perceived usefulness of a system, while the perceived ease-of-use in turn affects both perceived usefulness and ultimately user satisfaction. These relationships are depicted in Fig 4.3, which illustrates how user expectation's drive perception of quality and satisfaction.

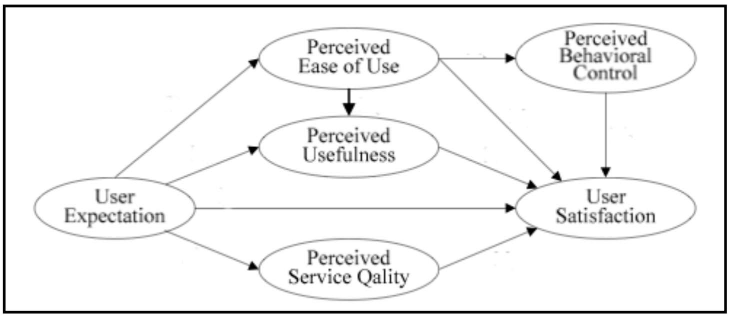

Fig 4.3 IT end user satisfaction framework (Liu, Chen, Zhou, 2006)

An effective quality assurance model for mobile web applications will therefore aim to implement measures that impact positively on the perceived and actual ease-of-use and actively manage user's expectations of the capabilities of the application. An efficacious method of expectation management is to ensure that web applications adhere to a set of best practices that provide a consistent standard of functionality with an intuitive user interface.

4.3 World Wide Web Consortium (W3C)

The World Wide Web Consortium (W3C) is an "international community" of member organizations, supported by a full time staff, whose mission is to "lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long term growth of the web". The Consortium is led by Sir Tim Berners-Lee, who is credited with inventing the world-wide-web and is administered jointly by three "host institutions" - the European Research Consortium for Informatics and Mathematics (ERCIM), the Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory (MIT CSAIL) and Keio University in Japan (W3C, 2009). As of 10 April 2012, there were 356 member organizations within the W3C, including many high profile technology companies such as AT&T, Cisco, Facebook and Microsoft Corporation (W3C, 2012a). Though the majority of member organizations are US or UK based, Irish members account for 2% of the overall membership figure. Fig 4.4 outlines the breakdown of W3C member organization distribution by country.

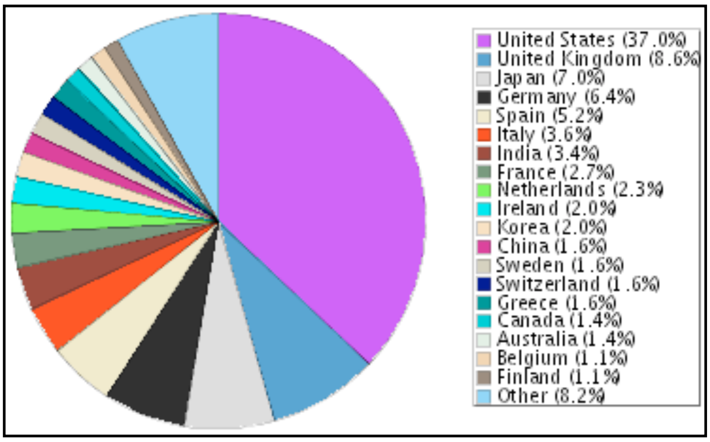

Fig 4.4 W3C membership by country (W3C, 2012b)

One of the core principles of the W3C is to promote a "Web on Everything" philosophy, which is focused on ensuring web interoperability across a large range of internetwork enabled devices, including smartphones, PDAs, televisions and many other consumer electronics. This is put into practical effect through the drafting and publishing of standards, protocols and best practices, in consultation with its member organizations.

4.4 W3C Standards Publication Process

The W3C standards are developed primarily through the efforts of specific Working Groups, formed to address a particular area of web technology standardization. Working Groups are responsible for researching and publishing Technical Reports, which form the basis for specifications and guidelines. The W3C defines a number of "maturity levels" which a Technical Report must reach in order to be eventually accepted as a W3C Recommendation, which is an analogous term for a Web Standard. In order for a Technical Report to be advanced to a Recommendation, it passes through the following stages, outlined in Table 4.1 (W3C, 2005a):

Stage in Process	Description
Publication of First Working Draft	Requests views and review from interested parties
Last Call announcement	Specifies a deadline for review comments by members and the public
Calls for implementation	Requests implementation proposals
Call for Review of Proposed Recommendation	Peer review of the proposals
Publication of W3C Recommendation	Proposal is formally released as a Recommendation

Table 4.1 W3C Recommendation publication process (W3C, 2005a)

A number of W3C Working Groups are currently grouped under the auspices of the Mobile Web Initiative, which advances the W3C efforts to promote and develop the "Web on Everything" philosophy mentioned above.

4.3.1 Mobile Web Initiative

The Mobile Web Initiative (MWI) of the W3C was initiated in 2005 in order to support software developers targeting mobile devices, in addition to researching and progressing "viable standards for the technological backbone of the mobile web" (Dick, 2010). The initiative receives its funding from W3C members, and the sponsors include firms such as Vodafone Group Services Ltd, Nokia, HP, France Telecom and Ericsson (W3C, 2005b). The MWI has overseen the publication of both the Mobile Web and Mobile Web Application Best Practices, as well as providing practical guidance and advice to developers on how to design accessible and mobile-friendly websites.

4.5 Mobile Web Application Best Practices

The Mobile Web Application Best Practices (MWABP) were released by the MWI as a W3C Recommendation on 14th December 2010, following a lengthy consultation and drafting process. The stated goal of the recommendation is to assist developers in the production of "rich and dynamic mobile Web applications", by promoting web engineering practices that contribute to an enhanced user experience and warning against those "which are considered harmful" (W3C, 2010c). The Recommendation is comprised of 32 Best Practice Guidelines, which the MWI describe as "forward looking" and therefore expected to remain valid for the foreseeable future. It is this web standard, and the development of associated automated conformance tests that this project is most concerned with.

Unlike the Mobile Web Best Practices, which outline a Default Delivery Context (DDC) as the minimum specifications and capabilities which a mobile device should possess in order to be able to effectively display web content (W3C, 2008), the Mobile Web Application Best Practices assume only that the mobile device provides support for standard XHTML, JavaScript and CSS capabilities. The BPWG do suggest however that "mid to high end devices" are likely to benefit most from the best practices (W3C, 2010c). It is the implementation and testing of these best practices that this dissertation is most concerned with. The previous quality models discussed, including the ISO approach, provide historical and technical context to facilitate the in-depth discussion of the W3C standards. In order to further subdivide the Mobile Web

Application Best Practices, the Working Group categorizes the recommendations under 6 subheadings. Table 4.2 enumerates and describes these subheadings:

Subcategory	Description
Application Data	Techniques to manage web application data, eg settings, preferences, emails,
Security and Privacy	Protect Personally Identifiable Information and reduce application security threats
User Awareness and Control	Allows user to control access to device capabilities(SMS, GPS, etc) as well as local storage
Conservative use of resources	Techniques to minimize use of device processor, memory and network bandwidth
User Experience	Enhance overall experience by considering factors such as latency, data consistency and user interaction
Handling Variations in the Delivery Content.	Web Applications should adapt to different Delivery Contexts (mobile devices) to ensure broad coverage of devices

Table 4.2 MWABP subcategories (W3C, 2010c)

The 32 Best Practices are listed in Tables 4.3 to 4.8 according to the appropriate subcategory, along with a brief description in respect of each guideline.

Application Data	Use Cookies Sparingly	Cookies may be unreliable or disabled in mobile devices so mobile web applications should be able to function without them.
Application Data	Use Appropriate Client Side storage technologies for Local Data	Client side storage reduces network bandwidth required and can accelerate application startup time to native app levels
Application Data	Replicate Local Data	Provides a consistent view of user data across multiple devices

Table 4.3 MWABP Application Data Guidelines (W3C, 2010c)

Security and Privacy	Do not execute untrusted or unescaped JSON data	Direct execution of JSON data using JavaScript's eval() function on mobile devices can represent a security risk

Table 4.4 MWABP Security/Privacy Guidelines (W3C, 2010c)

User Awareness & Control	Ensure user is informed about use of personal and device information	User should retain control over whether the application may use APIs should as Contacts, GeoLocation, etc.
User Awareness & Control	Enable Automatic Sign-in	Minimize requirement for data input by mobile users by allowing automatic sign-in

Table 4.5 MWABP User Awareness/Control Guidelines (W3C, 2010c)

Conservative Use of Resources	Use Transfer Compression	Web servers should be configured to deliver compressed data to reduce network bandwidth requirement on mobile
Conservative Use of Resources	Minimize Application and Data Size	Applications which are smaller in size will download and execute more efficiently
Conservative Use of Resources	Avoid Redirects	The network delays which are incurred by redirects impact more negatively on mobile devices
Conservative Use of Resources	Optimize Network Request	Establishing fresh network connections can take significantly longer on mobile devices
Conservative Use of Resources	Minimize External Resources	External CSS, scripts and other resources (eg pictures) should be consolidated to reduce the number of network requests.
Conservative Use of Resources	Aggregate Static Images into Single Composite resource (Sprites)	Each image requires a separate HTTP request, they should be aggregated into a single sprite

Conservative Use of Resources	Include Background images inline in CSS	Larger background images should also be handled by CSS as a base64 encoded string to reduce HTTP requests
Conservative Use of Resources	Cache Resources by Fingerprinting Resource References	Dynamic resources can be cached using a hash of the resource content in the URL. This also reduces network requests
Conservative Use of Resources	Cache AJAX Data	AJAX data should be cached similarly to external resources
Conservative Use of Resources	Do not send Cookie information unnecessarily	Repeatedly sending cookie information places additional burden on network performance of mobile devices
Conservative Use of Resources	Keep DOM size reasonable	Large or complex pages on devices with in-memory DOM size restrictions will cause performance degradation

Table 4.6 MWABP Use of Resources Guidelines (W3C, 2010c)

User Experience	Optimize for Application start-up time	By using local storage and caching, the web app can initiate without a network connection
User Experience	Minimize Perceived Latency	By preloading probable next pages and notifying users of loading activity, perceived latency is reduced
User Experience	Design for multiple interaction methods	Consider designing for touch, focus and pointer based interaction to increase interface options for users
User Experience	Preserve focus on dynamic page updates	JavaScript can be used to keep the page focus on the dynamically updated portion.
User Experience	Use Fragment IDs to drive application view	Web Apps should switch views without a full page reload by showing or hiding sections on content which have been already downloaded

User Experience	Make telephone numbers Click-to-Call	Users should be able to select phone numbers in web apps to initiate a phone call
User Experience	Ensure paragraph text flows	By not using pixels or other absolute measures in the CSS, paragraphs will flow correctly on different screen sizes
User Experience	Ensure consistency of states between devices	Application data should be stored on the server rather than locally to facilitate consistency of state
User Experience	Consider mobile specific technologies for initiating web applications	Network-initiated content delivery ("push" notifications) should be used if supported by the user agent
User Experience	Use Meta Viewport element to identify desired screen size	Instructs the mobile device how to scale the page, in order to optimize zoom levels

Table 4.7 MWABP User Experience Guidelines (W3C, 2010c)

Handling variation in Delivery Context	Prefer Server Side detection where possible	Server side detection of the user agent efficiently allows custom content to be delivered to mobile devices
Handling variation in Delivery Context	Use Client Side detection when necessary	If a web server is unable to identify a particular mobile device type, this may be identified on the client side
Handling variation in Delivery Context	Use device classification to simplify content adaptation	Deploy device specific applications for varying delivery contexts and deliver the appropriate app at runtime.
Handling variation in Delivery Context	Support a non-JavaScript variant if applicable	Not all mobile browsers implement JavaScript. Its absence should not cause a mobile web application to crash without notification
Handling variation in Delivery Context	Offer users a choice of Interface	Mobile users should be offered the alternative of the desktop version

Table 4.8 MWABP Delivery Context Guidelines (W3C, 2010c)

It is important to note that not all of the best practices are applicable to all types of mobile web application. The degree to which the best practices should be implemented is affected largely by the domain specific considerations of the mobile web application in question. For instance a mobile web application, such as a simple calculator or game, which does not require Automatic Sign-In will not implement the relevant best practice. It is therefore important for developers and testers to have an appreciation for the best practice guidelines that are applicable to the domain within which their mobile web application resides.

4.5.1 EU MobiWebApp Initiative

The MobiWebApp Initiative is a W3C project which is partly funded by the European Union through the Seventh Framework Programme and which has as its goals the promotion of standard's compliant mobile web application development and testing approaches to validate the best practices. In its most recent report on Best Practice Testing progress, the MobiWebApp Initiative concluded that having completed its deliberations on defining the scope and initial requirements for implementing automated conformance tests at the end of 2011, it now "needs to focus on developing the testing framework and gather test cases" (MobWebApp Initiative, 2011). Chapter 5 of this dissertation seeks to contribute to this objective, through the proposal of a conformance testing approach suitable for several of the Mobile Web Application Best Practices. The research undertaken in this dissertation is conducted independently of and is not affiliated with the efforts of the MobiWebApp Initiative.

4.5.2 Mobile Web Best Practices

In addition to its work on the Mobile Web Application Best Practices, the MWI has also facilitated the preparation of a set of best practices for the mobile web, under the auspices the Best Practice Working Group (BPWG). This group was formed in 2005 to develop a set of "technical best practices and associated materials in support of development of web sites that provide an appropriate user experience on mobile devices" (WC3, 2010a). This W3C Recommendation is intended to be complimentary to the Mobile Web Application Best Practices (W3C, 2010c). The work of the BPWG was accepted as a Web Standard in July 2008, though the group continued its work on Mobile Web Application Best Practices until 2010(W3C, 2010b).

The best practices that form this Web Standard consist of a set of 60 guidelines that are designed to improve the end user experience of browsing web sites from a mobile device. These guidelines focus largely on design issues and content formatting to overcome the limitations of screen size and user input associated with mobile devices. The guidelines therefore include recommendations that promote the use of short URLs, dissuade the use of pop-ups, tables and image maps, and suggest use of appropriate typefaces and colour combinations to achieve optimal contrast (W3C, 2008). These practices, while related closely with the Mobile Web Application Best Practices, are not tested for explicitly in the testing approaches proposed in Chapter 5. The Mobile Web Application Best Practices are not a subset of these practices, and form an independent quality assurance approach tailored for HTML 5 based applications rather than static sites.

The BPWG also partially based its work on the Web Content Accessibility Guidelines, a W3C standard first published as part of the Web Accessibility Initiative in 1999 which aims to ensure web content is accessible to users with a disability or other impairment (W3C, 1999). The Web Accessibility Guidelines, while originally designed to accommodate users with a disability/impairment, are an example of how Universal Design can benefit all users of a system. These guidelines promote document clarity, simplicity and the use of equivalent alternatives to audio and visual content.

4.5.3 MobileOK Basic Tests

The MWI provides an online tool which checks the level of "mobile friendliness" of a web site, by assessing the implementation of a subset of the Mobile Web Best Practices that can be automatically checked (http://validator.w3.org/mobile/). This tool, called the "mobileOK checker", is developed in the Java programming language and is accessed through a front end on the MWI website where a URL for the web site under test is entered (Daoust and Hoschka, 2004). This system defines a test case for each guideline, using an oracle comparator to predict the conditions that would indicate compliance. This process is outlined pictorially in Fig 4.5, which uses the example of the guideline that recommends against using a table for layout purposes. This best practice statement is used to propose a formal definition of the machine test to be conducted. This is then implemented in the Java based testing framework, and indicates a PASS/FAIL/WARNING depending on the outcome of the test.

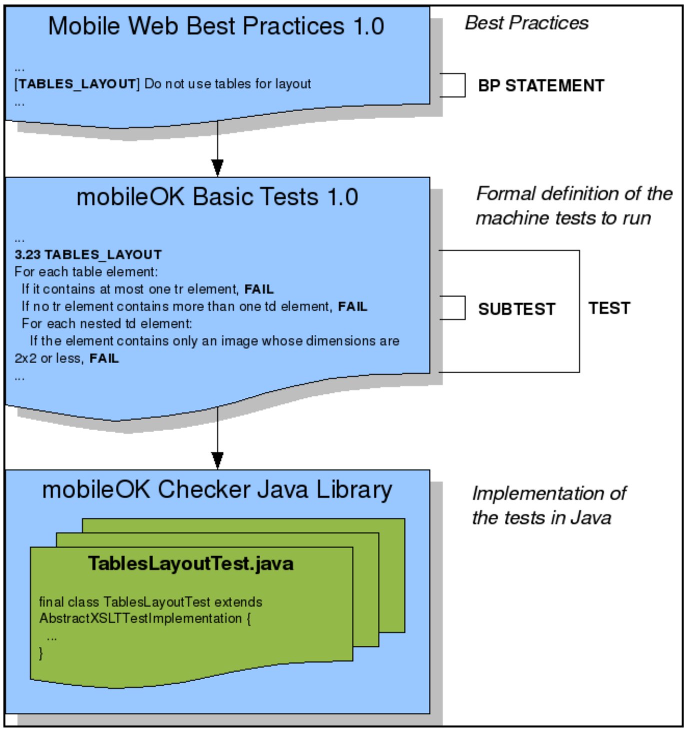

Figure 4.5 Relationship between mobileOK checker and best practices (Daoust & Hoschka, 2004)

The mobileOK model of implementing conformance tests, using the best practice guidelines to inform the definition of the machine test, is an approach while is built upon in this dissertation's treatment of testing the mobile web application best practices in chapter 5. While there is no direct overlap between the tests implemented in mobileOK and the testing software proposed in this project, the approach whereby a guideline is expressed as a machine test for validation is nonetheless one that is employed in this research.

Having reviewed three major initiatives undertaken by the W3C, namely the Mobile Web Best Practices, the Mobile Web Application Best Practices and the Web Accessibility Initiative, Figure 4.6 outlines how these 3 separate undertakings are interconnected. There are several validation tools available to assess a web site's adherence to the provisions of the Web Accessibility Initiative guidelines on accessibility for the broadest possible range of web users. The W3C maintains a listing of these tools, which is updated with additional validation tests as they are developed

(W3C, 2006). As previously alluded to, the W3C MobileOK Checker has been made available by the W3C to validate a web site's adherence to the Mobile Web Best Practices. This research focuses on developing automated conformance test approaches to validate implementation of the W3C Mobile Web Application Best Practices through the development of a Mobile Web App Checker (MWAC) tool, indicated in Fig 4.6 by the green arrow. This work is undertaken entirely separately from and independent of the efforts of the EU MobiWebApp Initiative.

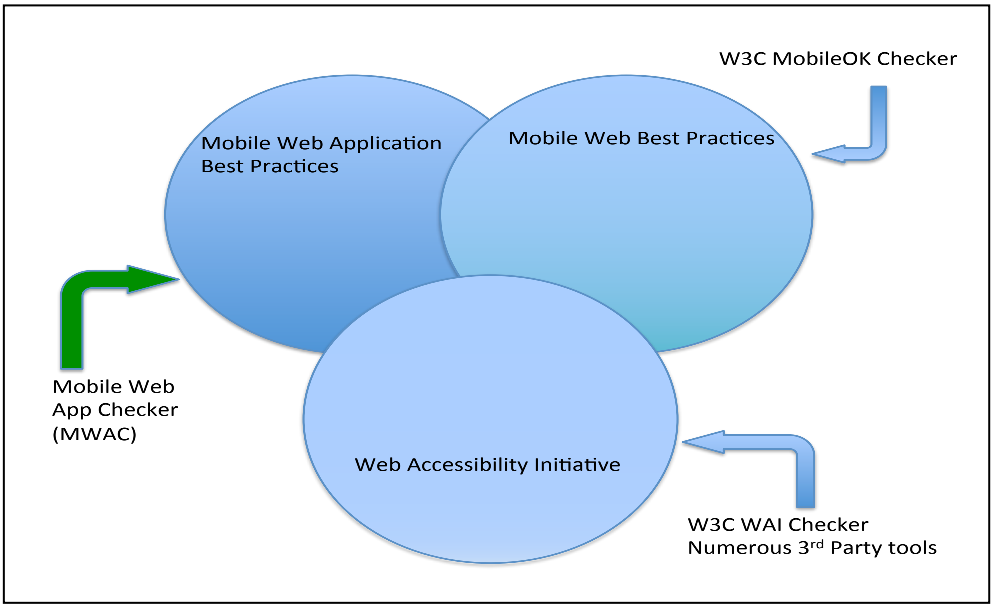

Fig 4.6 W3C Guidelines & Testing Tools

4.6 Mobile Web Application Developer Survey

In order to gauge the perceived utility of an automated testing system to validate implementation of the W3C Mobile Web Application Best Practices, a survey was administered to 11 web developers who completed the W3C Tech Course entitled "Mobile Web 1: Best Practices" in March 2012. This fully on-line course, completed by the author, is conducted over an 8-week period under the auspices of the MWI and instructs students in practical approaches to implementing W3C Mobile Web Application Best Practices and W3C Mobile Web Best Practices (W3DevCampus, 2012). The survey also sought to ascertain the current methodologies employed by developers when implementing quality assurance measures. Developer satisfaction with current approaches and software was also examined in an effort to identify gaps in the tools and techniques available for testing and quality assurance of mobile web applications. The full results of this survey, as well as the questions posed are included in Appendix A of this dissertation.

72.7% of those surveyed indicated that they were employed full-time in web application development, with a further 18.2% working part-time in the industry. Respondents were generally well established in the field, with 45.5% having worked in web application development for 5 years or longer. While a majority of those surveyed were familiar with the W3C Mobile Web Application Best Practices to some extent, only 18% had consciously implemented the guidelines in previous projects. This would suggest that while developers are aware of established best practices, further training and instruction in their deployment is important to achieve more widespread uptake by the development community. Figure 5.1 depicts graphically the level of familiarity with the guidelines that respondents claimed to possess prior to undergoing the Mobile Web 1 course.

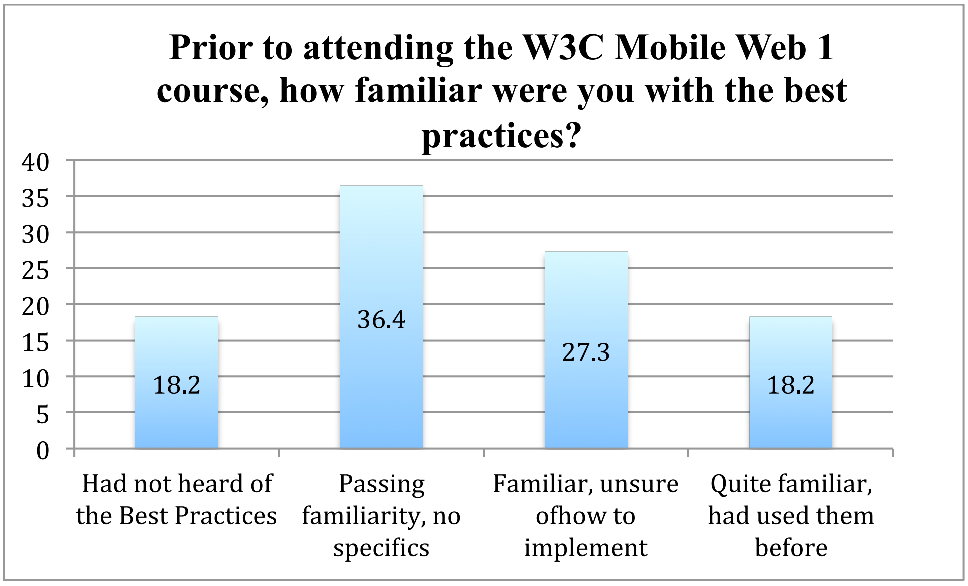

Fig 4.7 Respondents familiarity with MWABP prior to W3C Mobile Web 1 Course

Of those surveyed, 91% reported that they found the W3C Mobile Web Application Best Practices to be effective in assisting developers to ensure high quality mobile web applications, with one respondent noting that "ultimately it's still up to the developer to ensure rigorous testing" is conducted to achieve and maintain high quality web applications.

In terms of the proportion of total development time devoted to testing mobile web applications, the responses were rather more varied. 36.4% indicated that they spent between 11% and 20% of the total development time on application testing, though 18.2% suggested that between 41% and 50% of the total development time for a web application project is devoted to testing activities. The diverse range of responses to how much project time is devoted to testing is depicted in Fig 5.2.

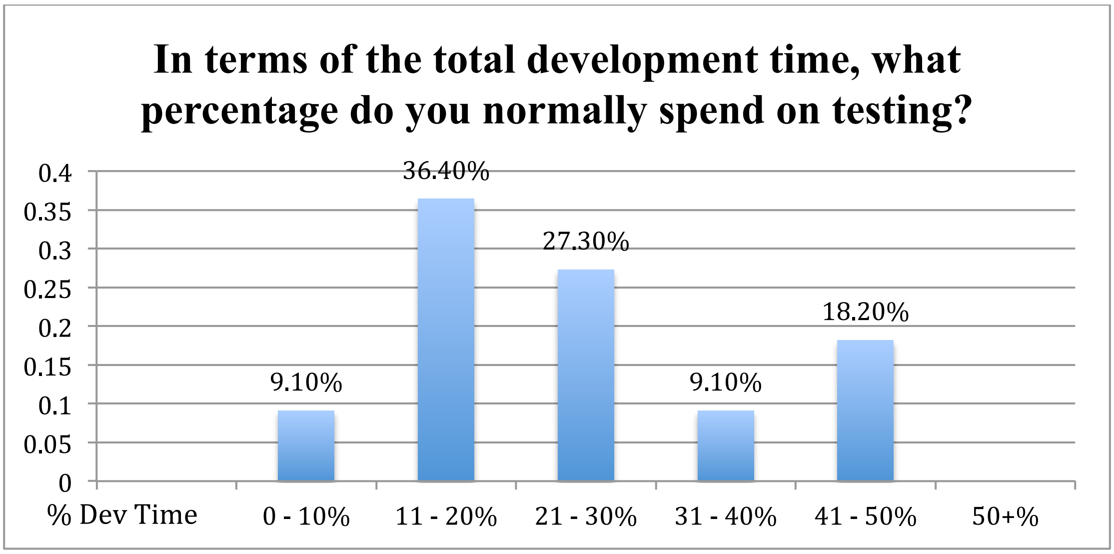

Fig 4.8 Respondents familiarity with MWABP prior to W3C Mobile Web 1 Course

When asked if they habitually implement any other forms of best practice or quality assurance approaches to mobile web application development, 27.3% of respondents indicated that they did not. Of the 45.5% of developers who have claimed to employ other quality assurance approaches, popular quality frameworks were Twitter Bootstrap and Yahoo! YSlow. Twitter Bootstrap provides a selection of prebuilt JavaScript libraries, HTML templates and CSS style sheets, which can be combined to build websites that employ many of the W3C best practices (http://twitter.github.com/bootstrap/). YSlow is an on-line testing application developed by Yahoo! which performs tests on web pages to assess their implementation of Yahoo!'s 34 best practices for high performance websites (http://yslow.org/). There is a certain amount of cross over between the W3C Mobile Web Application Best Practices and the standards developed by third party companies, however the W3C guidelines enjoy the status of being industry recognized standards and are tailored specifically to the mobile context.

The results of this developer survey would indicate a high level of acceptance of the benefits of the W3C Best Practice guidelines, as well as significant value being placed on the advantages of automated conformance testing of best practice implementation. An encouraging 100% of respondents intend to utilize the best practice guidelines in their future web application projects. Significantly, 90.9% of those surveyed indicated that an automated testing tool to validate compliance with the W3C MWABP would be a useful resource for developers.

While not all of the best practice guidelines are appropriate to every mobile web application domain, they do form a reliable basis for developing applications which offer the best possible experience to end users, across a broad range of mobile platforms and browsers. Several of the best practice guidelines require a subjective judgment call by developers in deciding whether they are appropriate for the particular web application under development. In order to assess which of the guidelines are suitable candidates for machine tests to validate their implementation levels, it is important to consider which best practices can be assessed in an automated manner and which ones require human verification.

4.7 Conclusion

This chapter examines the primary framework for software quality assurance, ISO 9126 and outlines how this approach has been tailored and customized for use in the web application sphere. While several web application quality models have been proposed, the W3C has assumed a leading role in advancing best practices for both mobile website and mobile web application development. The best practices are arrived at through a comprehensive process of consultation with interested parties and research conducted by the relevant W3C working groups. In order to support the deployment of best practices, automated conformance testing provides a means for developers to benchmark their own implementation of the accepted guidelines. The MWI mobileOK checker provides this facility for mobile web site validation, however there is as yet no publicly available equivalent framework available for mobile web applications.

The results of a mobile web application developer survey, administered to developers who had recently completed the W3C Mobile Web 1: Best Practices course, clearly identified the value of an automated conformance testing solution to verify the correct implementation of the W3C MWABP. These findings also underlined the substantial time and effort that must be invested in manually testing and verifying the correct application of best practices by web application developers.

Chapter 5 examines the extent to which the Mobile Web Application Best Practices can be assessed using automated client side testing, and proposes implementation approaches for those guidelines which lend themselves to automated conformance checking. Automated conformance tests for the W3C Mobile Web Application Best Practices are then implemented in a Java based testing framework, which is also described in the next chapter.

5 ASSESSMENT CRITERIA FOR W3C BEST PRACTICES

5.1 Introduction

Whilst Chapter 4 addressed the issues surrounding the background to and requirement for a formal set of best practices for mobile web application development, this chapter focuses on developing automated testing approaches to validate best practice implementation in mobile web applications. Section 5.2 discusses the general approach adopted in arriving at proposed automated test cases in respect of the applicable best practices. Having considered the best practice in general, each of the 32 W3C Mobile Web Application Best Practices are then described and analyzed to ascertain the degree to which they can be tested for in a mobile web application. For those guidelines that are suited to automated conformance testing, implementations of machine tests are proposed and deployed in a Java based testing programme.

Testing all possible aspects of a mobile web application's behavior, while clearly desirable, is generally extremely difficult to achieve. This barrier to accurately and efficiently testing all possible execution paths through a web application arises in large part because of the "activity combinatorial explosion" problem, whereby the complexity and frequency of user inputs to the application results in a vast number of possible execution paths to examine and validate (Li, Miao and Qian, 2008). Even within a relatively straightforward web application which does not involve frequent server side updates, dynamic updates to the DOM brought about by client side validation of user inputs can result in significant variations to pre-defined test and execution paths within the application (Alshahwan et al, 2009). Conformance testing, which checks if the behavior of the System Under Test (SUT) is correct within predefined specifications, can be achieved however where discerning test cases definition is deployed when deciding the behavior and output to validate (Li, Qian & He, 2009).

While it is reasonable to operate on the basis that full testing coverage of mobile web applications is often extremely difficult to achieve, comprehensive testing is nonetheless an essential element in effective software quality assurance. By limiting the scope of automated testing to conformance testing of best practice implementation, the difficulties associated with activity combinatorial explosion can be largely avoided. Mobile Web Application developers and testers in particular stand to achieve significant economies in terms of resources and time expended on testing activities when such testing can be performed in an automated or semi-automated manner.

5.2 Automated Conformance Testing

In considering which best practices are amenable to automated conformance testing, a number of factors were taken into account. Automated test cases were successfully generated in respect of 19 of the 32 W3C Mobile Web Application Best Practices. These test cases were then implemented as machine tests in a Java application, which extended the Webdriver/Selenium 2 testing framework. This application, called "Mobile Web Application Checker (MWAC)", simulates a mobile user agent when interacting with the web server which hosts the mobile web application under test. The tests in respect of each guideline are effected using the HTTP response headers and an examination of the DOM structure of the mobile web application as inputs.

Test case verdicts in respect of each guideline are defined as either PASS, INCONCLUSIVE or alternatively an output of textual information that the tester can interpret in the context of the application under test. A PASS result indicates that the implementation of a particular guideline is actively detected by MWAC, while an INCONCLUSIVE result suggests that the implementation of a guideline has not been identified. In the case of 3 of the guidelines the results of a conformance test is presented as text, which outputs relevant assessment metrics to the screen. These metrics require a judgment call on behalf of the human tester to verify the degree to which the particular guideline has been observed; this will largely be dependent on the particular application under test. Since not all guidelines are applicable to every application domain, an INCONCLUSIVE result does not specify whether it is appropriate that a guideline implementation should be included in a particular mobile web application. The test case definition processes for these 19 best practices are outlined hereunder.

5.2.1 Application Data

This category encompasses 3 guidelines and refers to best practices relating to managing web application data efficiently on mobile devices.

Guideline 1: Use Cookies Sparingly

Cookies, while useful for storing personalization data and authentication tokens for automatic sign-in should not be relied on in the mobile context. The network overhead associated with passing cookie information between client and server is considered by the W3C Working Group to have a negative impact on application performance. The cookies used by a mobile web application are specified in the HTTP response headers sent to the mobile device by the web application server. The number and content of these cookies can be inspected by examination of these response headers. MWAC identifies the number of cookies sent to the client by the mobile web application, and presents this metric to the tester to indicate the reliance or otherwise of the application on cookies.

Guideline 2: Use Appropriate Client Side Storage Technologies for Local Data

This best practice prescribes client-side storage mechanisms, available on modern mobile devices, as an alternative to cookies for storing application data. This is achieved in effect through the use of the Local Storage API provided for in HTML5 (W3C, 2008). MWAC validates implementation of this guideline by examining the HTML structure of the mobile web application, and identifying the manifest and local storage attributes. The presence of these attributes in a mobile web application indicates the use of client-side data storage and yields a PASS result in respect of this guideline.

Guideline 3: Replicate Local Data to a Server if Necessary

This guideline states that while it is appropriate that certain data such as preference settings be stored locally, other data types should be replicated to a server to maintain a consistent view across a user's different devices accessing the same mobile web application. When transferring such data to a server, the mobile web application must first establish whether the mobile device is reporting that it is online. MWAC checks whether the "onLine" property of DOM navigator object is invoked in the mobile web application when assessing the implementation of this guideline.

5.2.2 Security and Privacy

The MWABP specify just one guideline relating to mobile web application security, but suggest that "mobile web applications are subject to the same security considerations and risks as desktop web applications" (W3C, 2010).

Guideline 4: Do not execute Unescaped or Untrusted JSON data

Using the JavaScript eval() function to execute JSON data which has been transferred to the client's mobile device, while a fast and powerful technique, can represent a significant security risk. The W3C has assessed that "inadvertently executing malicious JavaScript is particularly dangerous on mobile devices", due to the sensitive and personal information stored on such devices. This best practice suggests that a JSON parser be implemented to parse JSON data before execution. MWAC tests for the implementation of this guideline by searching the HTML structure of the mobile web application for the presence of a JSON parser. The presence and use of such a parser indicates that any JSON data required for the web application will be handled safely and will generate a PASS result in MWAC.

5.2.3 Conservative Use of Resources

As stated in previous chapters, a limiting characteristic of mobile devices is the relative lack of processing, memory and display resources compared with desktop environments. This category encompasses 11 separate guidelines, making it the single largest classification within the MWABP.

Guideline 7: Use Transfer Compression

This guideline calls for web application data to be compressed prior to transfer to the client device. MWAC tests the implementation of this best practice through examining the HTTP response headers sent by the web application HTTP server. If the server indicates that transfer compression, such as "gzip", is in use the guideline is assessed as being correctly implemented and yields a PASS result.

Guideline 8: Minimize Application and Data Size

In order to reduce the overhead associated with transfer of larger files to mobile devices, this guideline calls for applications to be "minimized" prior to transfer. "Minification" can be achieved by the removal of whitespace and comments from HTML, CSS and script used in a mobile web application. MWAC measures the sizes of these parameters before and after applying a "minification" algorithm to this data in order to assess whether this guideline is implemented. Successful detection of minification techniques such as white space or comment removal from the HTML and scripts associated with the application under test will generate a PASS result.

Guideline 9: Avoid Redirects

The network delays associated with HTTP 3xx responses, indicating a redirect of a HTTP request is deemed by the MWABP Working Group as being a significant performance issue for mobile web applications. MWAC examines the HTTP response headers when accessing a mobile web application and reports if the web server returns a HTTP 3xx response. Failure to detect a HTTP 3xx response will results in a PASS for this guideline.

Guideline 11: Minimize External Resources

Mobile Web Applications that rely on external scripts, images and stylesheets, incur additional network overhead when downloading these resources to the client device. The extent to which a web application depends on such resources varies widely and is largely based on the complexity and scope of the application. MWAC presents the tester with a summary of the number of external stylesheets, images and scripts that the mobile web application requires. This information is presented for comparison with other mobile web applications in similar categories.

Guideline 12: Aggregate Static Images into a single Composite Resource

Each image used in a mobile web application represents an additional HTTP request made by the device to the web server, which can be detrimental to the performance of the application. This guideline calls for images to be aggregated into a single sprite sheet, which would require a single HTTP request to transfer all of the images. MWAC tests this guideline by examining the CSS structure of the mobile web application to identify the use of CSS sprites. If CSS sprites are detected in the stylesheets associated with a mobile wen application, a PASS result will be returned.

Guideline 13: Include Background Images Inline in CSS

Encoding background images as a base64 encoded string in the CSS of a mobile web application also saves an additional HTTP request to fetch the image. This guideline is tested by examining the mobile web application's CSS to detect the use of base64-encoded strings. The presence of a base64-encoded string representing a background image will generate a PASS result.

Guideline 14: Cache Resources by Fingerprinting Resource References

Resources that are generally static, but can change infrequently can also be cached by the mobile browser. This is achieved by setting an appropriate resource cache policy, using the "Expires" HTTP header and using a hash of the content in the URL that references the resource. In this way, if the resource is changed, the hash will also change, thus indicating the newer version of the resource should be downloaded to the client device. This guideline is tested for by examining the HTTP response headers and resource URLs for the appropriate implementation of these measures.

Guideline 17: Keep DOM Size Reasonable

The capacity of mobile devices to accommodate large and complex page structures is less developed than that of desktop computers. Accordingly this guideline prescribes that the DOM size should be kept as small as possible. As the DOM size varies depending on the complexity and functionality of the mobile web application, MWAC displays the number of DOM nodes to the tester for comparison purposes, without quantifying what DOM size is considered "reasonable". The optimal DOM size required to most efficiently execute a given web application depends on the application domain and complexity.

5.2.4 User Experience

This subcategory outlines best practices that seek to enhance the fluidity and responsiveness of mobile web applications by implementing strategies to improve latency and user interaction methods on a mobile device.

Guideline 18: Optimize for Application Start-Up Time

By implementing measures to enable mobile web applications to start-up as rapidly as native applications, the user experience is significantly enhanced. Rapid web application initiation can be achieved through the use of off-line technologies such as the HTML5 App Cache. MWAC assesses the implementation of this guideline by searching the DOM structure of the mobile web application for the inclusion of the HTML5 App Cache. The detection of this App Cache will yield a PASS outcome in respect of this guideline.

Guideline 22: Use Fragment IDs to drive Application View

Fragment IDs can be used to switch views in a web application, without the need for a full-page reload. This can be implemented by hiding or showing parts of a page that has already fully downloaded as required, using the fragment ID in a URL to indicate which page sections should be displayed. MWAC tests for the use of fragment IDs by inspecting the links within a mobile web application and reports whether fragment IDs have been detected.

Guideline 23: Make Telephone Numbers Click-to-Call

Modern mobile devices allow users to initiate phone calls and send SMS messages from within a web application. This is achieved through the use of standardized URL schemes which include "<a href ='tel:" or "<a href='sms:". MWAC checks for these URLs schemes by examining the hyperlinks within a mobile web application and reports PASS if these schemes are detected.

Guideline 27: Use Meta Viewport Element to Identify Desired Screen Size

Many mobile browsers use automatic zooming to display the correct screen size of a desktop site. By using the viewport meta elements, the mobile web application can specify the most appropriate zoom level to display a page on a mobile browser. MWAC tests the implementation of this guideline by checking the HTML structure of the mobile web application for the inclusion of this viewport element and reports if it is PASS.

5.2.5 Handling Variations in the Delivery Context

The best practices listed under this category relate to providing an optimized user experience across a broad range or delivery contexts, i.e. different mobile devices, by tailoring the web application content for the particular client device.

Guideline 28: Prefer Server Side Detection Where Possible

This guideline recommends using server side processes to identify the device type that is accessing the mobile web application. This server side detection allows the correct content type to be served without requiring the client device to actively request alternative device specific content. MWAC assesses the implementation of this guideline by examining the HTTP response headers of the mobile web application and checks if the web server uses the User Agent String of the mobile browser to identify the device type. If the Vary: User-Agent response header is detected by MWAC, a PASS result is returned.

Guideline 29: Use Client Side Detection Where Necessary

In cases where server side detection of the device type is not implemented, the next preferred method of device detection is to use a client side process. This is commonly achieved through the use of JavaScript or CSS media queries to classify mobile devices by screen width. MWAC checks the HTML header element to assess if such a JavaScript or CSS media query is in place and reports PASS if either is detected.

Guideline 31: Support a non JavaScript Variant if Appropriate

This guideline suggests that mobile web applications should handle the unavailability on the device of JavaScript gracefully. This is achieved through including a "<noscript>" element in the HTML of the mobile web application and outputting an appropriate notification to the user instead of the application exiting/failing to initiate without informing the user. MWAC checks the DOM structure of the mobile web application and assesses whether the "<noscript>" element is present. Successful detection of this element within the HTML structure of a mobile web application will generate a PASS result.

5.2.6 MWABP Not Tested for Automatically

Of the remaining 13 best practices, 6 are assessed as requiring a subjective judgment call on behalf of a human tester and are therefore unsuited to machine testing. These 6 guidelines, as well as proposed manual assessment criteria are examined in Section 5.3. The implementation levels of outstanding 7 best practices, which are not tested for using an automated process, are outlined below:

Guideline 10: Optimize Network Requests

This guideline calls for fewer, larger HTTP requests in place of more frequent requests between client and server and well as leveraging device awareness of connectivity (WiFi, 3G, UMTS, HSDPA, etc) to determine if requests should be throttled or delayed. The degree of implementation of this guideline is specific to both the web application domain and networking characteristics of the mobile device. These factors suggest that this best practice is unsuited to a generic automated conformance test.

Guideline 15: Cache AJAX Data

Whilst it is relatively straightforward to assess to degree to which caching techniques and resource fingerprinting are implemented in a mobile web application through an examination of the associated HTTP response headers and HTML structure, reliably measuring the extent to which these techniques are applied specifically to AJAX data across the range of possible web applications is not practicable.

Guideline 16: Do Not Send Cookie Information Unnecessarily

The degree to which cookie information is passed between client and server depends largely on the functionality, scope and domain of the mobile web application under consideration. As there is no reliable method of determining which cookie information should be considered “necessary” across the spectrum of web applications, this guideline is not assessed using automated tests in this project.

Guideline 21: Preserve Focus on Dynamic Page Updates

This best practice calls for the judicious application of the JavaScript “focus” method to move the focus of the web application to sections that have been changed dynamically. The implementation of this guideline is therefore subject to a judgment call on behalf of the web application developer as to what constitutes an essential use of this function, and cannot be reliably measured through automated conformance testing.

Guideline 25: Ensure Consistency of State between Devices

Achieving consistency of state across multiple devices requires a server side process to synchronize application data associated with a user's account or other identifying attributes. As this process is conducted on the server side, the associated conformance testing is outside the scope of this project.

Guideline 26: Consider Mobile Specific Technology to initiate Web Applications

This guideline promotes the use of "network initiated content delivery", commonly referred to as "Push" notifications on mobile devices. The implementation and testing of this technology constitute server side processes and therefore fall outside the scope of this project.

Guideline 30: Use Device Classification to Simply Content Adaptation

Device classification is a process carried out on the server side to identify the appropriate data to serve to a particular class of mobile device. As this is a server side activity, it falls outside the scope of the client side testing that this project addresses.

5.3 Visual Conformance Testing / Qualitative Assessment Criteria

The remaining 6 guidelines of the 32 Mobile Web Application Best Practices are most suited to direct visual verification and checking by a human tester. These guidelines promote practices that necessarily involve a subjective judgment call in terms of both their implementation and conformance testing, and are outlined below.

Guideline 05: Ensure User is informed About Use of Personal and Device Information

Many mobile web applications require access to device specific functions such as geo-location services or a user's address book. When mobile web applications attempt to access such API's, a native dialog box is normally presented, allowing the user to actively grant consent for such access. The appearance of such a notification is best verified by visual inspection by a tester, and can be checked visually using the browser interface on MWAC.

Guideline 06: Enable Automatic Sign-In

Automatic Sign-In is applicable to a limited subset of mobile web applications, the implementation of which is immediately apparent from a visual check of the web application interface. Such a visual check is facilitated through the browser interface provided by MWAC.

Guideline 19: Minimize Perceived Latency

The perception of latency in a mobile web application is a function of the expectations of the end user. Various presentation techniques, such as progress bars and intermediate loading screens, can reduce the perception of unfavorable latency without affecting the speed at which the application executes. Accordingly, a human tester most appropriately assesses the extent of an application's perceived latency.

Guideline 20: Design for Multiple Interaction Methods

In the context of mobile web applications, multiple interaction methods refer to utilizing either touch based, pointer based or focus based interaction styles. As this is a stylistic consideration, the most appropriate interaction method for a given web application domain is unsuited to an automated checking process and is most effectively assessed by visual testing.

Guideline 24: Ensure Paragraph Text Flows

Paragraph text should appear consistent and well formatted across a wide range of screen sizes associated with various mobile devices. While setting the appropriate viewport meta element should ensure the correct zoom level is implemented for a given screen size, the extent to which paragraph text appears to flow correctly is a subjective metric, which is best assessed by a human tester.

Guideline 16: Offer Users a Choice of Interface

A choice of interfaces for a web application involves permitting the user to select which version of the application they prefer, regardless of the device characteristic detection performed with locally or on the server side. The manner in which this choice is presented to the end user varies considerably between web applications, and is therefore best suited to a visual check by the tester.

5.4 Mobile Web Application Checker (MWAC)

The 19 best practice guidelines that were successfully incorporated into the automated conformance testing software (MWAC) are summarized below:

- Use Cookies Sparingly
- Use Appropriate Client Side Storage Technologies for Local Data
- Replicate Local Data
- Do not Execute Unescaped or Untrusted JSON Data
- Use Transfer Compression
- Minimize Application and Data Size
- Avoid Redirects
- Minimize External Resources
- Aggregate Static Images into a Single Composite Resource (Sprites)
- Include Background Images inline in CSS
- Cache Resources by Fingerprinting Resource References
- Keep DOM Size Reasonable
- Optimize Application for Start Up Time
- Use Fragment IDs to Drive Application View
- Make Telephone Numbers Click-to-Call
- Use Meta Viewport Element to Identify Desired Screen Size
- Prefer Server Side Detection Where Possible
- Support Client Side Detection When Necessary
- Support a non-JavaScript variant if appropriate

The implementation of the above mentioned 19 W3C Mobile Web Application Best Practices is assessed using the processes described in section 5.2 which are incorporated using the Mobile Web Application Best Practice Checker (MWAC) software.The initial screen that a tester is presented with when using this programme is depicted in Fig. 5.1. A list of user agents is available to the tester, and MWAC will use the appropriate user agent setting in its HTTP request headers while simulating the correct screen size based on the device selected at this point. Having selected a user agent, the tester must then input the URL of the mobile web application under test.

Fig 5.1 MWAC Initial Screen

The appropriate mobile browser is modeled in this application by instantiating a "webdriver" object based on the Selenium 2 API, with the appropriate configuration according to the tester's device choice. An example of the browser display that is generated by MWAC is seen in Fig. 5.2, which shows the output based on the selections made in Fig 5.1.

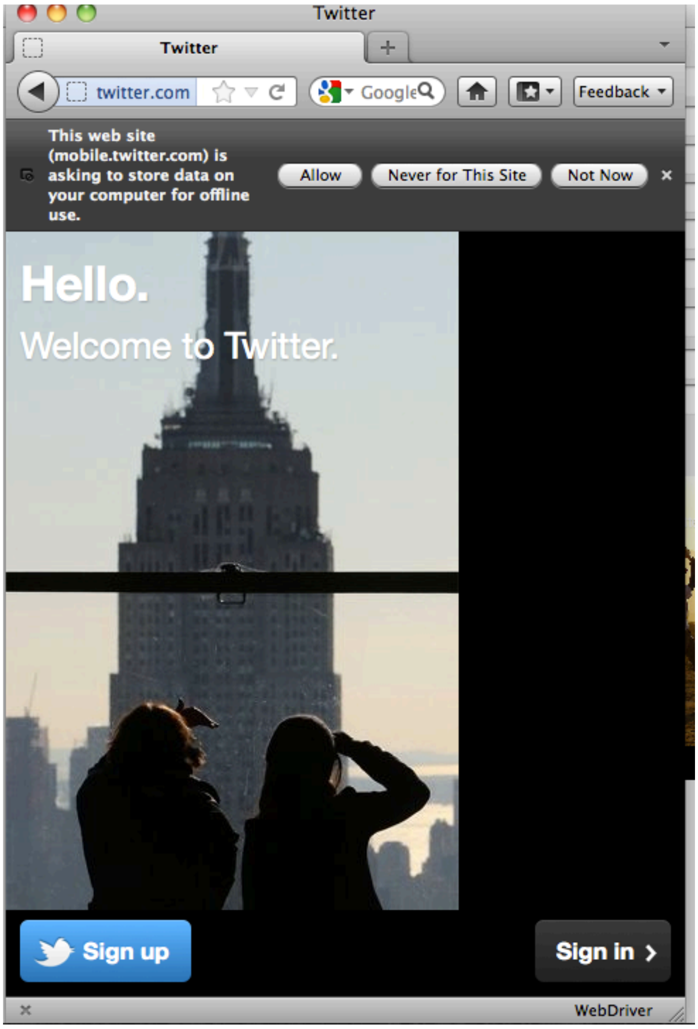

Fig 5.2 MWAC Mobile Browser Simulator

Having established the HTTP connection between the user-agent represented by MWAC and the server hosting the mobile web application under test, the automated conformance tests outlined above are executed. A Test class controls the actual conduct of each conformance test, and sets a Boolean flag of either true or false depending on the computed outcome of the machine tests. Once the tests have completed execution, the user is presented with the interface depicted in Fig 5.3. This screen outputs the HTTP response headers received from the web server in question, as well as a screenshot of the web application under test. The user is presented with a menu that allows the tester to exit the system, restart a test, or output the test results to a PDF file. This PDF is generated using the *iText PDF* Java API, and is configured to incorporate meta data such as the time and data of testing, the URL and user-agent utilized, as well as a result of PASS or INCONCLUSIVE in respect of each of the tested best practices.

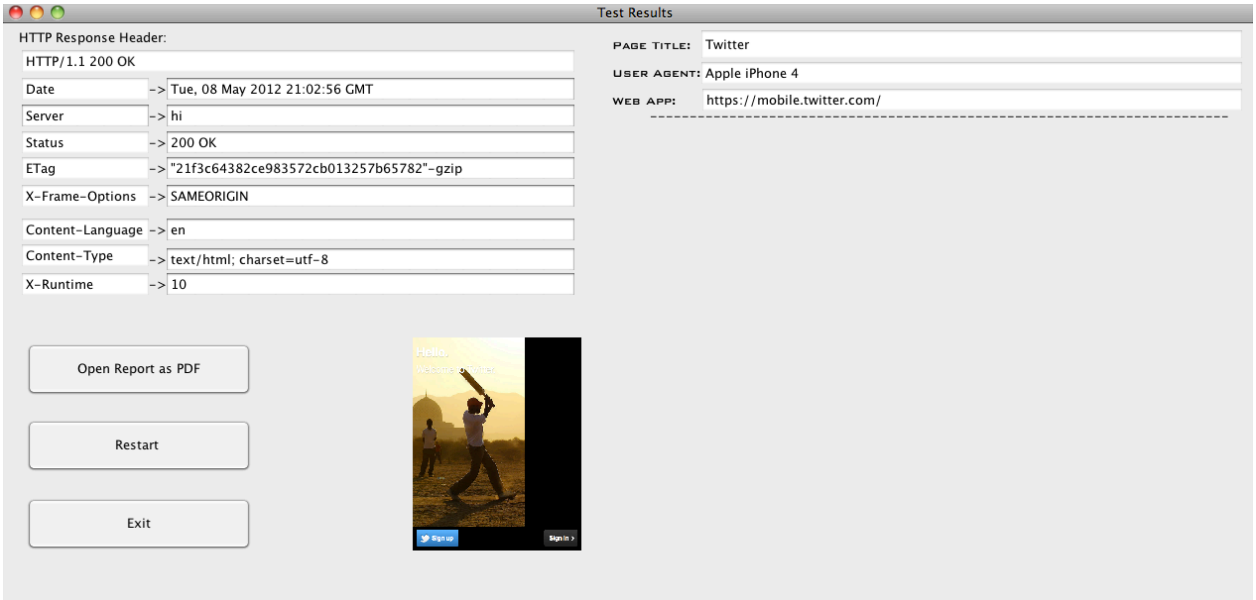

Fig 5.3 MWAC Results Screen

5.5 Conclusion

In attempting to address the requirement identified in previous chapters for a largely automated approach to conformance testing, this chapter proposes automated validation methods for 19 of the 32 best practice guidelines. 6 of the best practices were found to be best tested through the use of visual confirmation by a human tester, while 7 proved to present substantial difficulties in facilitating automated client side conformance testing.

Chapter 6 utilizes the Mobile Web Application Checker (MWAC) software, developed as part of this dissertation, to conduct conformance tests for 19 of the W3C Mobile Web Application Best Practices across a wide range of mobile web applications.

6 EXPERIMENTATION AND EVALUATION

6.1 Introduction

Whilst Chapter 5 describes the processes used to derive automated tests for verifying mobile web application conformance with the W3C best practices, this chapter focuses on evaluating the test tool by deploying these tests against a broad range of mobile web applications. Adherence to standards and best practices facilitates developers in producing high quality, cross-platform mobile web applications that take into account the specific capabilities and limitations of mobile devices. The aim of this chapter is to assess the extent to which the implementation of the W3C Mobile Web Application Best Practices can be effectively assessed using the proposed automated testing software. This work does not consider implementation of other third party quality assurance approaches or recommendations, such as those enumerated in the previous chapter. These third party systems, while undoubtedly valuable to many in the development community, do not represent an established and generally accepted web standard to the same extent as the W3C Best Practices.

As discussed in Chapter 5, it is neither appropriate nor necessary to implement every best practice guideline across every application domain. Accordingly, the mobile web applications that are selected to undergo automated conformance testing are grouped together by functional domain. This is to ensure that comparisons of best practice implementation are drawn in a consistent and meaningful manner across broadly similar categories of mobile web application.

While not every guideline applies to every application, there are certain best practices that should be observed in most, if not all, mobile web applications. Best practices relating to how content should be adapted and served for display on mobile devices, such as the use of device-type detection and viewport settings are appropriate to all categories of mobile web application. It is important to differentiate between the domain-specific guidelines that should be implemented as necessary to a particular web application category, and those best practices that should be deployed in respect of all mobile web applications, regardless of the specific functional domain.

6.2 Mobile Web Application Test Candidates

As outlined in Chapter 3, mobile web applications are currency enjoying strong growth in terms of end-user and developer preference, despite the continuing dominance of native mobile applications and the associated "App Store" paradigm. The strong growth in respect of mobile web applications has given rise to an increasingly large quantity of such applications in circulation, not all of which consistently apply best practices in their implementation. With the publication of the W3C Mobile Web Application Best Practices as an established web standard, it is important that mobile web applications adhere closely to its provisions to facilitate maximum cross compatibility and quality across a wide range of mobile devices and browsers.

As it is infeasible to conduct conformance testing in respect of every mobile web application, this study focuses instead on testing a representative subset of 20 mobile web applications. In order to allow for a consistent analysis of mobile web application best practice implementation, test candidate applications are classified under the below mentioned headings. These categories are selected on the basis of a recent App Consumer Behavior Pattern Report (Neilson, 2011), which found that these categories are amongst the most commonly accessed mobile web applications by US consumers, with News and Social Media being the most popular at 30% and 24% respectively.

1. Social Media/Chat
2. Photo/Video Sharing
3. News & Weather
4. Office/Productivity

Tables 6.1 to 6.4 below outline the 20 mobile web applications, categorized under the above-mentioned 4 headings, which are tested using the Mobile Web Application Checker (MWAC) tool described in Chapter 5.

Mobile Web Application	***Uniform Resource Locator (URL)***
Facebook	http://www.facebook.com
Twitter	http://www.twitter.com
LinkedIn	http://www.linkedin.com

Tumblr	http://www.tumblr.com
Morse Code Converter	http://www.pjnation.com/iphone/morse-code/

Table 6.1 Social Media/Chat Mobile Web Applications

Mobile Web Application	***Uniform Resource Locator (URL)***
YouTube	http://www.youtube.com
Flickr	http://www.flickr.com
Shozu	http://www.shozu.com/
iTrailers	http://itrailers.info/
Photobucket	http://www.photobucket.com

Table 6.2 Photo/Video Sharing Mobile Web Applications

Mobile Web Application	***Uniform Resource Locator (URL)***
Financial Times	http://m.ft.com
Weather Report	http://www.weather.com
Yelp	http://www.yelp.com
Weather Network Mobile	http://www.theweathernetwork.com/weatherapps/mobileweb
Pocket Market	http://www.pocketmarket.info/

Table 6.3 News/Weather Mobile Web Applications

Mobile Web Application	***Uniform Resource Locator (URL)***
Google Maps for Mobile	http://www.google.com/gmm/index.html
4G Speed Test	http://www.4g-speed.info
Color Mail	http://www.touchapp.co.uk/iphone/colormail/index.htm
Ultimate Thesaurus	http://www.1webapps.com/apps/utilities/thesaurus/
CheckList	http://alexgibson.github.com/checklist/

Table 6.4 Productivity/Office Mobile Web Applications

The 20 mobile web applications listed above were selected as a representative cross-section of current HTML 5 web applications across 4 broad domain categories. Each mobile web application was subjected to automated best practice conformance testing using MWAC, as described in Chapter 5. MWAC was configured to simulate an Apple iPhone 4 as the mobile user agent and the results reported relate to the 19 best practices in respect of which automated tests were generated. The full test reports produced by MWAC on each of the mobile web applications assessed are presented in Appendix B. The 19 best practice guidelines that are reported on using MWAC are outlined hereunder:

- Use Cookies Sparingly
- Use Appropriate Client Side Storage Technologies
- Replicate Local Data to a Server
- Do not Execute Unescaped or Untrusted JSON Data
- Use Transfer Compression
- Minimize Application and Data Size
- Avoid Redirects
- Minimize External Resources
- Aggregate Static Images into a single Composite Resource (Sprites)
- Include Background Images inline in CSS Stylesheets
- Cache Resources by Fingerprinting Resource IDs
- Keep DOM Size Reasonable
- Optimize Application for Start-up Time
- Use Fragment IDs to drive Application View
- Make Telephone Numbers Click-to-Call
- Use Meta Viewport Element to Identify Screen Size
- Prefer Server Side Detection Where Possible
- Use Client Side Detection where Necessary
- Support a non-JavaScript Variant where Appropriate

6.3 Testing Outcomes and Analysis

Table 6.5 highlights the 10 Mobile Web Application Best Practices most frequently implemented across the entire set of tested applications. The overwhelming majority of applications tested did not utilize HTTP 30X redirects, thus avoiding an unnecessary network overhead. Likewise the correct use of transfer compression while serving the application content in 15 of the 20 web applications examined reduces the burden that these applications place on the limited network resources available to most mobile devices, thus contributing to a responsive end-user experience with reduced content loading times.

Significantly, 85% of mobile web applications implemented the Viewport Meta element, ensuring that the correct screen size and "zoom" level appropriate to the particular mobile device was identified and catered for in these applications. This guideline is particularly important given the wide range of screen resolutions and orientations that may be found in modern mobile devices and smartphones.

Best Practice	*Percentage of Compliant Apps*
Avoid Redirects	95%
Use Viewport Meta Element	85%
Use Transfer Compression	75%
Aggregate Static Images into Sprites	50%
Use Server Side Detection where Possible	40%
Cache Resources by Fingerprinting Resource References	35%
Use Fragment IDs to drive Application View	35%
Support a non-JavaScript variant if appropriate	35%
Include Background Images inline in CSS Stylesheets	25%
Optimize Application for Start-up Time	20%

Table 6.5 Most Frequently Implemented Mobile Web Application Best Practices

The 5 best practices that were found to be least frequently implemented across the 20 tested mobile web applications are presented in Table 6.6.

Best Practice	*Percentage of Compliant Apps*
Make Telephone Numbers Click-to-Call	0%
Use Client Side detection where necessary	0%
Replicate Local Data to Server	5%
Minimize Application and Data Size	15%

Do not Execute Unescaped or Untrusted JSON Data	15%

Table 6.6 Least Frequently Implemented Mobile Web Application Best Practices

While on initial inspection the figures presented in Table 6.6 appear to indicate a general disregard for several important best practices, this is not necessarily the case. As previously alluded to, the extent to which particular guidelines are implemented in a given mobile web application depends largely on the functional domain of the application. A majority of web applications do not include nor do they require "clickable" telephone numbers as part of the application content, so implementing measures to allow Click-to-Call functionality would be redundant in all but a minority of mobile web applications. Similarly, the implementation of client-side device detection is obviated if this process is conducted on the server-side, which is the preferred method of content adaptation for the different mobile devices.

Fig 6.1 illustrates that mobile web applications tested in the Social Media/Chat category were found to have implemented marginally more of the guidelines overall than the other 3 categories. News/Weather applications and Productivity/Office applications were both found to have successfully deployed 27 guidelines, with the Photo/Video category implementing 26 best practices across the 5 web applications tested.

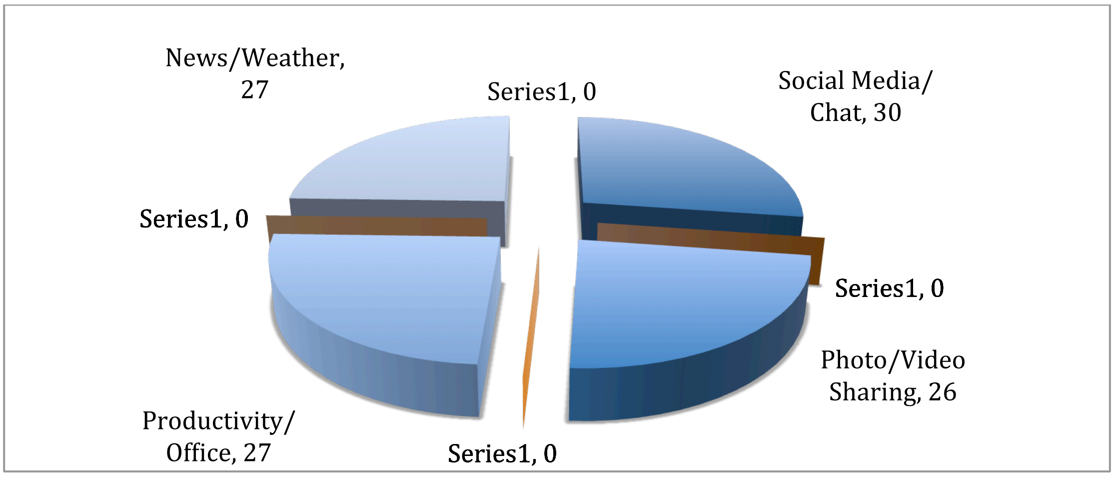

Fig 6.1 Number of Best Practices Detected in each Mobile Web Application Category

The consistency with which the implementation of best practices was detected across the 4 categories suggests that developers are taking cognizance of adherence to the W3C guidelines. The best practices that were found to be generally unimplemented largely relate to domain-specific functions, which are appropriate only in specific circumstances and for specific application types.

40 % of Mobile Web Applications in the Social Media/Chat category were found to have deployed a JSON parser to ensure that unescaped JSON data could not be executed from within the application, while none of the applications in the Photo/Video or News/Weather categories were found to have implemented this guideline. Similarly 80% of the Social Media/Chat web applications used CSS sprites to reduce the number of HTTP requests required to fetch image data from the server, while the other categories were far less compliant with this guideline. Given that CSS sprites are appropriate primarily for static images, while news, weather and photo-sharing applications would tend to rely on dynamically served images, this finding further lends credence to the assertion that the appropriateness of a particular guideline implementation is a function of the application domain.

6.3.1 Best Practices appropriate to all mobile web application categories

The most frequently implemented best practices identified in this chapter provide an insight into the W3C guidelines that are most appropriate for consideration in respect of all mobile web applications. Regardless of the web application domain or functionality, there is a universal requirement for data transfer to be optimized for the mobile context and for the device-appropriate screen dimensions to be catered for. This is reflected in the 3 most observed best practices identified in this study, two of which relate to minimizing network overhead associated with serving web application content, with the other focusing on the use of the Viewport Meta element to identify the correct screen size.

40% of mobile web applications tested were found to successfully utilize server-side detection of the mobile device type, which further reduces the computational requirement on the client device to request device specific content. All mobile web applications, regardless of functionality, should make provisions for server-side device detection through the appropriate configuration of the web/application server. Many mobile web applications include static images as part of the content served to the client, and half of those applications tested utilized CSS sprites to reduce the number of network requests to service this requirement. This guideline is appropriate to all mobile web applications that include static images as a key component of the application.

6.3.2 Domain-specific Best Practices

Table 6.6 outlines those guidelines that were found to be implemented least frequently or not adopted at all across the mobile web applications tested. As stated, these best practices are appropriate to specific application types, and should not be considered necessary in the case of every mobile web application. In addition to the 5 guidelines mentioned in table 6.6, the best practices listed below are considered most suited for inclusion only in mobile web applications that leverage the relevant technologies or functions.

- Ensure the User is informed about Personal/Device information
- Enable automatic sign-in
- Design for multiple interaction methods
- Ensure consistency of state between devices
- Offer users a choice of interface

6.4 Extent to which W3C MWABPs can be assessed using automated client-side processes

This dissertation set out to investigate the requirement for and feasibility of developing automated conformance tests to validate the implementation of the W3C Mobile Web Application Best Practices. The results of the survey administered to developers of mobile web applications pointed clearly to the benefits that could be achieved through an automated approach to conformance testing in results of the guidelines. The benefits of automated software testing are further corroborated by reference to the literature examined in chapters 3 and 4.

The testing strategies outlined in the previous chapter were implemented as machine tests in respect of 19 of the 32 best practices, representing 59.38% coverage of the total number of guidelines. The remaining 13 guidelines were divided conceptually into 2 categories – those that were assessed as being most appropriate to a qualitative judgment call by a human tester and those that were assessed as being too broad in scope to be accurately quantified by client-side processes.

6.5 Conclusion

The results of the automated conformance testing in respect of the 20 mobile web applications considered in this study indicate a significant variance in terms of best practice implementation levels. A majority of Mobile Web Applications across the 5 categories successfully incorporates Guideline 27, which mandates the use of the "Viewport" Meta element to detect the correct device screen size. The correct use of this guideline is essential in ensuring that mobile web applications are displayed correctly in the user device browser. Guidelines that are not so uniformly observed across the 5 categories include Minimizing Application and Data Size, Replicating Local Data to the Server and Client-side Device Detection. These guidelines, while important in assuring maximum interoperability across a range of mobile platforms, are not critical to the functionality or presentation of the mobile web applications under test.

The 32 W3C Mobile Web Application Best Practices can therefore be considered as consisting of two separate categories of guideline; those appropriate for implementation in all mobile web applications and those which should be reserved for specific application types. In general terms, those guidelines relating to network optimization and the correct presentation of material for the limited screen real estate of mobile devices should be deployed across all mobile web applications. The best practices which deal with issues such as automatic sign-in, correct handling of AJAX data and enabling click-to-call for telephone numbers should be implemented only as required and can be safely ignored for many web application types.

As the range and diversity of mobile handsets continues to expand, so too will the necessity for developers to adhere to established web standards come into sharper focus. By incorporating the W3C Mobile Web Application Best Practices into the design, implementation and testing phases of the Software Development Lifecycle, developers can ensure that their mobile web applications execute correctly and to specification across all standards-compliant mobile web browsers. Conversely, failure by application developers to observe the best practices can result in web applications which do not perform correctly across the full spectrum of mobile devices, thus negating one of the key advantages offered by mobile web applications over native apps, namely effective cross-platform compatibility.

7 CONCLUSION

7.1 Introduction

The purpose of this project was to explore the issues surrounding best practice and quality assurance when designing mobile web applications, and to investigate whether the implementation of a particular set of best practices could be assessed using automated client-side testing processes. The requirement for robust and comprehensive web standards governing mobile web application development, due mainly to the vast array of different devices, browsers and screen size in circulation, was identified by reference to the large volume of supporting literature and a survey of mobile web application developers.

The W3C Mobile Web Application Best Practices were analyzed and automated test cases suggested in respect of 19 of the 32 guidelines. These 19 test cases were then implemented as automated machine tests in a conformance-testing framework developed in support of this dissertation. A broad range of mobile web applications were assessed using this testing tool, with conclusions offered in the preceding chapter in respect of the results generated.

7.2 Research Definition and Research Overview

This dissertation undertook to research the implications on software quality and reliability in the context of the recent proliferation of mobile devices and the associated mobile web applications. The specific software quality assurance model that was subjected to detailed analysis was the W3C Mobile Web Application Best Practices. The following research objectives were initially defined at the commencement of this project:

- To explore definitions and concepts surrounding Web Application development and contextualize these concept within the sphere of Software Development and Web Engineering.
- To identify key literature and existing research related to Mobile Web Application testing and development.
- To define and assess the requirement for developers and designers to observe adherence to “best practice”.

- To identify assessment criteria which are appropriate to the 32 W3C Best Practice guidelines.
- To design appropriate test cases that effectively assess best practice compliance levels.
- To develop a testing suite capable of automating identified test cases.

The historical, business and technical background relating to this paradigm of software development and distribution was explored through the relevant literature. Primary research in the form of a web application developer survey, and the development of an automated mobile web application-testing suite was also conducted.

7.3 Contributions to the Body of Knowledge

This dissertation seeks to contribute to the body of knowledge associated with mobile web applications and best practice testing chiefly in two ways. Quantitative primary research in the form of a web application developer survey highlights a number of interesting findings in relation to the time and effort devoted to manual testing of mobile web applications. The requirement for automated approaches to best practice conformance validation was identified initially through the literature review and was reinforced and corroborated by reference to the conclusions drawn from this survey.

A second contribution proposed by this dissertation is a general conformance testing approach that is capable of achieving coverage of 59.38% of the total W3C Mobile Web Application Best Practices. While work in this regard is also underway under the auspices of the EU MobiWebApp Initiative, described previously in Chapter 4, this dissertation aims to contribute to the discussion surrounding issues relating to the automated validation of W3C best practices and standards. If the results of the best practices identified in Chapter 6 as being mostly commonly implemented are applicable across mobile web applications more generally, it can be concluded that some guidelines are favoured more strongly by developers and suggests that certain guidelines are applicable to all mobile web applications, while others relate to specific categories of application only.

7.4 Experimentation, Evaluation and Limitations

A survey of web application developers who had recently completed the W3C Mobile Web 1: Best Practices course was administered in the course of this research project. The results of this survey indicate clearly the requirement that web application developers have identified for an automated approach to validating mobile web application best practice implementation. Manual validation and testing was found to occupy a significant portion of the total development time for many mobile web applications, further corroborating the contention that manual software testing is a time and human-capital intensive activity.

The results of the automated conformance tests performed in the course of this dissertation using the MWAC tool indicated a general level of conformance in several specific guidelines across a majority of mobile web applications. Those guidelines relating to implementing data transfer compression and the identification of the correct screen size and essential to the effective functioning of practically all mobile web applications. The uniformly high levels of implementation of these guidelines, particularly when compared with low implementation levels in respect of some domain specific best practices, then suggests that several guidelines are considered more generally relevant than others.

The investigation of automated testing processes in this project was restricted to client-side testing only. Whilst this was an apparently fruitful approach in respect of a majority of the best practices, there remain others that are not effectively assessed using a client-side only approach.

7.5 Future Work and Research

While this dissertation proposes automated conformance testing approaches in respect of 59.38% of the W3C Mobile Web Application Best Practices, further work in this area should be able to achieve greater coverage of the remaining guidelines. This research focused on client-side testing strategies, which may not be sufficient to fully address testing of guidelines that may be more appropriately assessed using server-side processes. Specifically guidelines relating to single sign in and ensuring consistency of state across different user devices are not reliably assessed relying on client-side tests

alone. Future work in this area could focus on addressing implementing server-side test strategies to ensure more complete coverage of the best practices.
In terms of the guidelines found to be more suited to a subjective judgment on behalf of a human tester, further work could be valuable in proposing a structured approach to guiding the testers' qualitative decision-making process. It is suggested however that the perceived quality of this decision-making process may be primarily a function of the experience and skill of the human tester.

7.6 Conclusion

Mobile web applications as an alternative channel for mobile application distribution continue to increase in popularity, in line with the growing ubiquity of smartphones, PDA's and other Internet enabled mobile devices. Several companies, notably the Financial Times, have abandoned the traditional App Store distribution model and released HTML 5 based mobile web applications independently. As consumers grow ever more reliant on the correct and reliable functioning of these applications across the varied domains within which mobile web applications now operate, the necessity for observance of best practices and standards comes ever sharper into focus.

The W3C Mobile Web Application Best Practices were developed by the W3C Mobile Web Working Group, in consultation with several interested parties. Broad industry acceptance of and agreement with the provisions of W3C web standards provides mobile web application developers with a set of guidelines that facilitate and encourage increased performance and cross-compatibility. The capability to automatically test for the implementation of these guidelines within a given mobile web application provides the developer with a powerful tool to ensure correctness and reduce overall development times. In assessing the extent to which such automated processes could be brought to bear on the W3C Mobile Web Application Best Practices, this dissertation has considered several approaches to web application testing. The testing approach for best practice validation proposed in this project was found to be effective in 19 of the 32 Best Practices. Given the economies of effort and improvements in quality assurance that can be achieved through automated testing, this is a field of study that is sure to attract further research and work as mobile web applications continue to increase in both quantity and sophistication.

BIBLIOGRAPHY

Aggarwal, C. and Yu, P. S. On the network effect in Web 2.0 applications. Electron. Comm. Res. Appl. (2011), doi:10.1016/j.elerap.2011.11.001

Al-Salem A and Ala Abu Samaha. 2007. Eliciting Web application requirements - an industrial case study. *J. Syst. Softw.* 80, 3 (March 2007), 294-313. DOI=10.1016/j.jss.2006.05.005 http://dx.doi.org/10.1016/j.jss.2006.05.005

Alshahwan, N.; Harman, M.; Marchetto, A.; Tonella, P.; "Improving Web Application Testing using testability measures," *Web Systems Evolution (WSE), 2009 11th IEEE International Symposium on* , vol., no., pp.49-58, 25-26 Sept. 2009 Doi:10.1109/WSE.2009.5630393URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=5630393&isnumber=5629685

Andrews J, Anneliese A. , Jeff Offutt, Curtis Dyreson, Christopher J. Mallery, Kshamta Jerath, and Roger Alexander. 2010. Scalability issues with using FSMWeb to test web applications. *Inf. Softw. Technol.* 52, 1 (January 2010), 52-66. DOI=10.1016/j.infsof.2009.06.002 http://dx.doi.org/10.1016/j.infsof.2009.06.002

Artzi, S.; Kiezun, A.; Dolby, J.; Tip, F.; Dig, D.; Paradkar, A.; Ernst, M.D.; , "Finding Bugs in Web Applications Using Dynamic Test Generation and Explicit-State Model Checking," *Software Engineering, IEEE Transactions on* , vol.36, no.4, pp.474-494, July-Aug. 2010doi: 10.1109/TSE.2010.31http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=5416728&isnumber=5532334

Behkamal B, Kahani M, Akbari. 2009, Customizing ISO 9126 quality model for evaluation of B2B applications, Information and Software Technology, Volume 51, Issue 3, March 2009, Pages 599-609, ISSN 0950-5849, 10.1016/j.infsof.2008.08.001.

Bruns, A.; Kornstadt, A.; Wichmann, D.; "Web Application Tests with Selenium," *Software, IEEE* , vol.26, no.5, pp.88-91, Sept.-Oct. 2009
doi: 10.1109/MS.2009.144
URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=5222802&isnumber=5222779

Campaign Monitor (2011). Email Client Popularity Report. Available from http://www.campaignmonitor.com/stats/email-clients/ Accessed 15 March 2012

Charland A and LeRoux B. 2011. Mobile Application Development: Web vs. Native. *Queue* 9, 4, Pages 20 (April 2011), 9 pages. DOI=10.1145/1966989.1968203 http://0-doi.acm.org.ditlib.dit.ie/10.1145/1966989.1968203

Charu C. Aggarwal, Philip S. Yu, On the network effect in Web 2.0 applications, Electronic Commerce Research and Applications, Available online 10 November 2011, ISSN 1567-4223, 10.1016/j.elerap.2011.11.001.
(http://0-www.sciencedirect.com.ditlib.dit.ie/science/article/pii/S156742231100069X)

comScore, 2012, Industry White Paper, *Mobile Future in Focus.* Accessed 15 March 2012:
http://www.comscore.com/layout/set/popup/Request/Presentations/2012/2012_Mobile_Future_in_Focus_Download?req=slides&pre=2012+Mobile+Future+in+Focus

Cortimiglia, M.N.; Ghezzi, A.; Renga, F.; , "Mobile Applications and Their Delivery Platforms," *IT Professional* , vol.13, no.5, pp.51-56, Sept.-Oct. 2011
doi: 10.1109/MITP.2011.84
URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=6028560&isnumber=6028548

Daoust F, Hoschka P, 2004. The W3C mobileOK Checker. W3C-ERCIM. Available at http://validator.w3.org/mobile/The-W3C-mobileOK-Checker.pdf. Accessed 12 April 2012

Deshpande, Y.; Hansen, S.; "Web engineering: creating a discipline among disciplines," *Multimedia, IEEE* , vol.8, no.2, pp.82-87, Apr-Jun 2001
doi: 10.1109/93.917974

Di Lucca G. and Fasolino A.R. 2006. Testing Web-based applications: The state of the art and future trends. *Inf. Softw. Technol.* 48, 12 (December 2006), 1172-1186. DOI=10.1016/j.infsof.2006.06.006 http://dx.doi.org/10.1016/j.infsof.2006.06.006

Dick M. 2010. Weaving the "Mobile Web" in the context of ICT4D: a preliminary exploration of the state of the art. In *Proceedings of the 73rd ASIS\&T Annual Meeting on Navigating Streams in an Information Ecosystem - Volume 47* (ASIS\&T '10), Vol. 47. American Society for Information Science, Silver Springs, MD, USA, , Article 35 , 7 pages.

Elbaum, S.; Karre, S.; Rothermel, G.; "Improving web application testing with user session data," *Software Engineering, 2003. Proceedings. 25th International Conference on* , vol., no., pp. 49- 59, 3-10 May 2003
doi: 10.1109/ICSE.2003.1201187 http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=1201187&isnumber=27042

Escalona, M J. Torres, M. Mej, J. J. Guti, and D. Villadiego. 2007. The treatment of navigation in web engineering. *Adv. Eng. Softw.* 38, 4 (April 2007), 267-282. DOI=10.1016/j.advengsoft.2006.07.006
http://dx.doi.org/10.1016/j.advengsoft.2006.07.006

Garofalakis J and Stefanis V. 2008. MokE: a tool for Mobile-ok evaluation of web content. In *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)* (W4A '08). ACM, New York, NY, USA, 57-64. DOI=10.1145/1368044.1368058 http://0-doi.acm.org.ditlib.dit.ie/10.1145/1368044.1368058

Gartner, 2011, Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012 [Internet] Accessed 10 March 2012: http://www.gartner.com/it/page.jsp?id=1622614

Gitzel R, Korthaus A, and Schader M. 2007. Using established Web Engineering knowledge in model-driven approaches. *Sci. Comput. Program.* 66, 2 (April 2007), 105-124. DOI=10.1016/j.scico.2006.09.001 http://dx.doi.org/10.1016/j.scico.2006.09.001

Global Intelligence Alliance (GIA), Industry White Paper 2/2010. *Native or Web Application - how best to deliver content and services to your audiences over the mobile phone, http://www.globalintelligence.com/insights-analysis/articles/mobile-applications-native-v-web-apps-what-are-the/read-online* Accessed 05 Feb 2012.

Gibbs, C. 2011; Why ‘Web Vs Native’ Isn’t a Black and White Battle, Gigaom article [Internet] Available at: http://gigaom.com/mobile/why-web-vs-native-isnt-a-black-and-white-battle/ .Accessed 29 March 2012.

Ginige, A.; Murugesan, S.; "Web engineering: an introduction," *Multimedia, IEEE* , vol.8, no.1, pp.14-18, Jan-Mar 2001
doi: 10.1109/93.923949
URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=923949&isnumber=19981

Golson, J. 2011. MacRumours News [Internet]. http://www.macrumors.com/2011/06/07/financial-times-wont-give-apple-a-cut-drops-ios-for-web-app/ Accessed 06 Feb 2012

Hieatt, E.; Mee, R.; "Going faster: testing the Web application," *Software, IEEE* , vol.19, no.2, pp.60-65, Mar/Apr 2002
doi: 10.1109/52.991333

URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=991333&isnumber=21378

Holmes, A.; Kellogg, M.; "Automating functional tests using Selenium," *Agile Conference, 2006* , vol., no., pp.6 pp.-275, 23-28 July 2006
doi: 10.1109/AGILE.2006.19
URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=1667589&isnumber=34909

Huang C; Chen H; "A Tool to Support Automated Testing for Web Application Scenario," *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on* , vol.3, no., pp.2179-2184, 8-11 Oct. 2006
doi: 10.1109/ICSMC.2006.385184
URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=4274190&isnumber=4274116

International Data Corporation (IDC), 2011. Smartphone Market Hits All-Time Quarterly High Due To Seasonal Strength and Wider Variety of Offerings, According to IDC. Available at http://www.idc.com/getdoc.jsp?containerId=prUS23299912. Accessed 12 April 2012

International Organisation for Standarization (ISO), 1991. ISO 9126 Software Engineering Product Quality – Part 1: Quality Model. Available from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22749. Accessed 15 March2012

Jia Zhang; Di Xu; , "A Mobile Agent-Supported Web Services Testing Platform," *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on* , vol.2, no., pp.637-644, 17-20 Dec. 2008
doi: 10.1109/EUC.2008.194

URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=4755297&isnumber=4755183

Lew, P.; Li Zhang; Shouxin Wang; Wei Jiang; , "Guidelines to Determine Quality for Web-Based Software Applications," *Computational Intelligence for Modelling Control & Automation, 2008 International Conference on* , vol., no., pp.709-714, 10-12 Dec. 2008
doi: 10.1109/CIMCA.2008.191

Li L and Miao H. 2008. An Approach to Modeling and Testing Web Applications Based on Use Cases. In *Proceedings of the 2008 International Symposium on Information Science and Engieering - Volume 01* (ISISE '08), Vol. 1. IEEE Computer Society, Washington, DC, USA, 506-510. DOI=10.1109/ISISE.2008.265 http://dx.doi.org/10.1109/ISISE.2008.265

Li L; Zhongsheng Q; He T; "Test purpose-based test generation for web applications," *Networked Digital Technologies, 2009. NDT '09. First International Conference on* , vol., no., pp.238-243, 28-31 July 2009
doi: 10.1109/NDT.2009.5272184
URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=5272184&isnumber=5272056

Liu Y; Chen Y; Zhou C.; "Determinants Affecting End-User Satisfaction of Information Technology Service," *Service Systems and Service Management, 2006 International Conference on* , vol.1, no., pp.478-481, Oct. 2006
doi: 10.1109/ICSSSM.2006.320509

Liu C, Kung D, Pei Hsia, and Chih-Tung Hsu. 2000. Structural Testing of Web Applications. In *Proceedings of the 11th International Symposium on Software Reliability Engineering* (ISSRE '00). IEEE Computer Society, Washington, DC, USA

Mansour N, Houri M. 2006, Testing web applications, Information and Software Technology, Volume 48, Issue 1, January 2006, Pages 31-42, ISSN 0950-5849, 10.1016/j.infsof.2005.02.007.
(http://0-www.sciencedirect.com.ditlib.dit.ie/science/article/pii/S0950584905000297)

McDonald, M, 2011. HTML 5 : The Missing Manual, O'Reilly Media Inc. Sebastopol, California, USA.

Matthews R, Jones P, Magid J, Ball D, Hammel M (1997) The UNIX Web Server Book, Ventana Communications Group, USA

Marin, B.; Vos, T.; Giachetti, G.; Baars, A.; Tonella, P.; , "Towards testing future Web applications," Research Challenges in Information Science (RCIS), 2011 Fifth International Conference on , vol., no., pp.1-12, 19-21 May 2011 doi: 10.1109/RCIS.2011.6006859
URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=6006859&isnumber=6006819

Mesbah, A.; Bozdag, E.; van Duersen, A.; , "Crawling AJAX by Inferring User Interface State Changes," *Web Engineering, 2008. ICWE '08. Eighth International Conference on* , vol., no., pp.122-134, 14-18 July 2008 doi: 10.1109/ICWE.2008.24
URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=4577876&isnumber=4577855

Mesbah A and Prassad M. 2011. Automated cross-browser compatibility testing. In *Proceedings of the 33rd International Conference on Software Engineering* (ICSE '11). ACM, New York, NY, USA, 561-570. DOI=10.1145/1985793.1985870 http://0-doi.acm.org.ditlib.dit.ie/10.1145/1985793.1985870

Mesbah A and van Duersen A. 2009. Invariant-based automatic testing of AJAX user interfaces. In *Proceedings of the 31st International Conference on Software*

Engineering (ICSE '09). IEEE Computer Society, Washington, DC, USA, 210-220. DOI=10.1109/ICSE.2009.5070522 http://dx.doi.org/10.1109/ICSE.2009.5070522

Mims, C. 2011; Why Mobile Apps will soon be Dead, MIT Technology Review [Internet] Available at: http://www.technologyreview.com/blog/mimssbits/26778/ .Accessed 29 March 2012.

MobWebApp Initiative, 2011, Testing Report Year 1. Available at http://mobiwebappw3c.files.wordpress.com/2011/10/testing_report-year1.pdf Accessed 08 April 2012

Montoto P, Pan A, Raposo J, Bellas F, and Javier 2011. Automated browsing in AJAX websites. *Data Knowl. Eng.* 70, 3 (March 2011), 269-283. DOI=10.1016/j.datak.2010.12.001 http://dx.doi.org/10.1016/j.datak.2010.12.001

Murugesan, S.; Venkatakrishnan, B.A.; , "Addressing the challenges of Web applications on mobile handheld devices," *Mobile Business, 2005. ICMB 2005. International Conference on* , vol., no., pp. 199- 205, 11-13 July 2005 doi: 10.1109/ICMB.2005.16 URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=1493609&isnumber=32116

Nabil D, Abeer Mosad, Hesham A. Hefny, Web-Based Applications quality factors: A survey and a proposed conceptual model, Egyptian Informatics Journal, Volume 12, Issue 3, November 2011, Pages 211-217, ISSN 1110-8665, 10.1016/j.eij.2011.09.003. (http://www.sciencedirect.com/science/article/pii/S1110866511000405) Keywords: Web Based Applications; Quality; Quality assurance; Quality models

Nielsen, Jakob (2009). Mobile Usability.[Internet] Accessed 14 March 2012: http://www.useit.com/alertbox/mobile-usability.html

Neilson (2011). App Consumer Behaviour Patterns. Available from http:// dk.nielsen.com/site/documents/appconsumerbehaviourpatterns.pdf. Accessed 20 April 2012

Parr, B, 2010, YouTube launches support for HTML 5, Mashable.com [Internet]. Accessed 21 March 2012: http://mashable.com/2010/01/20/youtube-html5/

Parveen, T.; Tilley, S.; Gonzalez, G.; 2007 "On the Need for Teaching Web Application Testing," *Web Site Evolution, 2007. WSE 2007. 9th IEEE International Workshop on* , vol., no., pp.51-55, 5-6 Oct. 2007
doi: 10.1109/WSE.2007.4380244 http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=4380244&isnumber=4380229

Pemberton (2006); The Future of Web Applications. CWI and W3C, Amsterdam. Available at http://www.w3.org/2006/Talks/11-23-steven-origo/. Accessed 20 March 2012

Qian Z; Miao H; Zeng H . 2008; "A Practical Web Testing Model for Web Application Testing," *Signal-Image Technologies and Internet-Based System, 2007. SITIS '07. Third International IEEE Conference on* , vol., no., pp.434-441, 16-18 Dec. 2007

Reuters, 2011; "FT Web Based App more popular than App sold in Apple Store." Available at http://www.reuters.com/article/2011/09/22/us-ft-idUSTRE78L49Q20110922. Accessed 15 March 2012

Ritchie P., The security risks of AJAX/web 2.0 applications, Network Security, Volume 2007, Issue 3, March 2007, Pages 4-8, ISSN 1353-4858, 10.1016/S1353-4858(07)70025-9.
(http://0-www.sciencedirect.com.ditlib.dit.ie/science/article/pii/S1353485807700259)

Sprenkle S, Hill E, and Pollock L. 2007. Learning Effective Oracle Comparator Combinations for Web Applications. In *Proceedings of the Seventh International Conference on Quality Software* (QSIC '07). IEEE Computer Society, Washington, DC, USA, 372-379

Vaughan-Nichols, S.J.; "Will HTML 5 Restandardize the Web?," *Computer* , vol.43, no.4, pp.13-15, April 2010
doi: 10.1109/MC.2010.119.URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=5445161&isnumber=5445153

Wang Y; Jin B; "The application of SaaS model in network education-take google apps for example," *Education Technology and Computer (ICETC), 2010 2nd International Conference on* , vol.4, no., pp.V4-191-V4-194, 22-24 June 2010
doi: 10.1109/ICETC.2010.5529703
URL: http://0-ieeexplore.ieee.org.ditlib.dit.ie/stamp/stamp.jsp?tp=&arnumber=5529703&isnumber=5529615

Wessels A, Purvis M, and (Shawon) Rahman. 2011. Usability of Web Interfaces on Mobile Devices. In *Proceedings of the 2011 Eighth International Conference on Information Technology: New Generations* (ITNG '11). IEEE Computer Society, Washington, DC, USA, 1066-1067. DOI=10.1109/ITNG.2011.190 http://dx.doi.org/10.1109/ITNG.2011.190

W3DevCampus, 2012. [Internet] Available at http://www.w3devcampus.com/mobile-web-and-application-best-practices-training/ Accessed 14 April 2012

W3C, 1999 [Internet] Web Content Accessibility Guidelines 1.0. Available at http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/ Accessed 06 April 2012.

W3C, 2005a [Internet] W3C Process Document. Available at http://www.w3.org/2005/10/Process-20051014/cover.html Accessed 05 April 2012.

W3C, 2005b [Internet] W3C Mobile Web Initiative. Available at http://www.w3.org/2005/Talks/200509Berlin/all.htm Accessed 10 April 2012

W3C, 2006 [Internet] W3C Web Accessibility Evaluation Tools. Available at http://www.w3.org/WAI/RC/tools/complete Accessed 10 April 2012

W3C, 2008 [Internet] W3C Mobile Web Best Practices 1.0. Available at http://www.w3.org/TR/mobile-bp/ Accessed 07 April 2012.

W3C, 2008. Offline Web Applications, Working Group note, Available from http://www.w3.org/TR/offline-webapps/. Accessed 20 April 2012

W3C, 2009 [Internet] W3C Mission Statement. Available at http://www.w3.org/Consortium/mission Accessed 05 April 2012

W3C, 2010a [Internet] Mobile Web Best Practice Working Group. Available at http://www.w3.org/2005/MWI/BPWG/ Accessed 05 April 2012

W3C, 2010b [Internet] Available at http://www.w3.org/blog/BPWG/2010/12/14/mobile_web_application_best_practices_is_2/ Accessed 07 April 2012

W3C, 2010c, [Internet] Available at http://www.w3.org/TR/mwabp/ Accessed 07 April 2012

W3C, 2012a [Internet] W3C Member List. Available at http://www.w3.org/Consortium/Member/List Accessed 05 April 2012

W3C, 2012b [Internet] W3C Membership Statistics. Available at http://www.w3.org/Consortium/Member/stats.html Accessed 07 April 2012

Zulzalil, H.; Zain, Z.M.; Ghani, A.; Selamat, M.H.; Mahmod, R.; , "Relationships analysis between quality factors for Web applications," Information Technology, 2008. ITSim 2008. International Symposium on , vol.3, no., pp.1-5, 26-28 Aug. 2008
doi: 10.1109/ITSIM.2008.4631988

APPENDIX A

Survey Administered to Web Application Developers who had graduated from W3C Mobile Web 1: Best Practices course

1. **Do you work professionally in web application development?**

Response	Response Percentage	Response Count
Yes	**72.7%**	8
No	9.1%	1
Part-Time	18.2%	2

2. **For how long have you been involved in web application development?**

Response	Response Percentage	Response Count
0 – 1 Year	18.2%	2
1 – 3 Years	18.2%	2
3 – 5 Years	18.2%	2
5+ Years	**45.5%**	5

3. **Prior to attending the W3C Mobile Web Best Practice course, how familiar were you with the best practices?**

Response	Response Percentage	Response Count
I had not heard of them before	18.2%	2
I knew they existed, but no specific details	**36.4%**	4
I was familiar with them, but unsure of how to use them	27.3%	3
I was quite familiar with them, and had used them before	18.2%	2

4. **How would you rate the Effectiveness of Best Practices to assist developers ensure quality sites/apps?**

Ineffective	Not Very Effective	Adequate	Quite Effective	Very Effective
0	0	9.1%	**45.5%**	**45.5%**

5. **How would you rate the Level of ease in implementing Best Practices in sites/apps?**

Very Difficult	Quite Difficult	Not Difficult	Quite Easy	Very Easy
0	27.3%	**36.4%**	27.3%	9.1%

6. **Do you currently use any software/websites to validate/test your mobile apps conformance with best practice**

Response	Response Percentage	Response Count
Yes	**45.5%**	5
No	27.3%	3
Sometimes	27.3%	3

7. **In terms of the total development time, what percentage do you normally spend on testing your site/app**

Response	Response Percentage	Response Count
0 – 10%	9.1%	1
11 – 20 %	**36.4%**	4
21 – 30 %	27.3	3
31 – 40 %	9.1%	1
41 – 50 %	18.2%	2
50 + %	0	0

8. Do you plan to implement the W3C Best Practices in your future projects?

Response	Response Percentage	Response Count
Yes, all of them where applicable	**63.6%**	7
Yes, as many as I can	36.4%	4
Some of them	0	0
None of them	0	0

9. Do you use any other quality assurance/best practices in developing your sites/apps?

Response	Response Percentage	Response Count
Yes	**45.5%**	5
No	27.3%	3
Sometimes	27.3%	3

10. Do you feel an automated software tool to check compliance with best practices would be useful to you as a developer?

Response	Response Percentage	Response Count
Yes	**90.9%**	10
No	0	0
I don’t know	9.1%	1

APPENDIX B

Mobile Web Application Checker v0.9 Best Practice Report

Report on Facebook at URL http://www.facebook.com

Tested on 2 May 2012 17:43:18 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies 5
2. Use Appropriate Client Side Storage Technology: Inconclusive
3. Replication Local Data to Server: Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information: Visual Check Required
6. Enable Automatic Sign In: Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression PASS
8. Minimize Application and Data Size: Inconclusive
9. Avoid Redirects No Redirect Found
10. Optimize Network Requests: Not Tested
11. Minimise External Resources:
    - No of Scripts:9
    - No of Images :0
    - No of CSS :1
12. Aggregate Static Images (CSS Sprites): PASS
13. Background images inline in CSS: Inconclusive
14. Cache Resources by Fingerprinting: PASS
15. Cache AJAX Data: Not Tested
16. Do not send Cookie Information Unnecessarily: Not Tested
17. Keep DOM size reasonable: Number of DOM Nodes:110

User Experience

18. Optimize Start Up Time (appcache): Inconclusive
19. Minimize Perceived Latency: Visual Check Required
20. Design for multiple Interaction Methods: Visual Check Required
21. Preserve Focus on Dynamic Page Updates: Not Tested
22. Use Fragment IDs to drive Application View: PASS
23. Make telephone numbers Click-to-Call: Inconclusive

24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	Inconclusive
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	PASS
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Twitter at URL http://www.twitter.com

Tested on 2 May 2012 17:49:45 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	2
2. Use Appropriate Client Side Storage Technology:	PASS
3. Replication Local Data to Server:	PASS

Security and Privacy

4. Do not execute unescaped/untrusted JSON data:	JSON Parser Detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	PASS
9. Avoid Redirects	HTTP 30X Detected
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:4 No of Images :12 No of CSS :1	
12. Aggregate Static Images (CSS Sprites):	PASS
13. Background images inline in CSS:	PASS
14. Cache Resources by Fingerprinting:	PASS
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:738

User Experience

18. Optimize Start Up Time (appcache):	PASS
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	PASS
23. Make telephone numbers Click-to-Call:	Inconclusive
24. Ensure Paragraph Text Flows:	Visual Check Required

25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	Inconclusive
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	PASS
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on LinkedIn at URL http://www.linkedin.com

Tested on 2 May 2012 17:52:37 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	0
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data:	JSON Parser Detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:3 No of Images :0 No of CSS :1	
12. Aggregate Static Images (CSS Sprites):	PASS
13. Background images inline in CSS:	PASS
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:40

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive
24. Ensure Paragraph Text Flows:	Visual Check Required

25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	Inconclusive
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Tumblr at URL http://www.tumblr.com

Tested on 2 May 2012 17:55:53 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	1
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources:	
No of Scripts:3	
No of Images :1	
No of CSS :0	
12. Aggregate Static Images (CSS Sprites):	Inconclusive
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:38

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive

24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	Inconclusive

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	Inconclusive
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Morse Code Converter at URL http://www.pjnation.com/iphone/morse-code/

Tested on 2 May 2012 17:58:37 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	1
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data:	JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	Inconclusive
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:18; No of Images :7; No of CSS :1	
12. Aggregate Static Images (CSS Sprites):	PASS
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:113

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive

24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	Inconclusive
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on YouTube at URL http://www.youtube.com

Tested on 2 May 2012 18:06:12 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies 4
2. Use Appropriate Client Side Storage Technology: PASS
3. Replication Local Data to Server: Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information: Visual Check Required
6. Enable Automatic Sign In: Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression PASS
8. Minimize Application and Data Size: PASS
9. Avoid Redirects No Redirect Found
10. Optimize Network Requests: Not Tested
11. Minimise External Resources:
    - No of Scripts:3
    - No of Images :10
    - No of CSS :3
12. Aggregate Static Images (CSS Sprites): PASS
13. Background images inline in CSS: PASS
14. Cache Resources by Fingerprinting: PASS
15. Cache AJAX Data: Not Tested
16. Do not send Cookie Information Unnecessarily: Not Tested
17. Keep DOM size reasonable: Number of DOM Nodes:214

User Experience

18. Optimize Start Up Time (appcache): PASS
19. Minimize Perceived Latency: Visual Check Required
20. Design for multiple Interaction Methods: Visual Check Required
21. Preserve Focus on Dynamic Page Updates: Not Tested
22. Use Fragment IDs to drive Application View: PASS
23. Make telephone numbers Click-to-Call: Inconclusive

24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	Inconclusive
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	PASS
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Flickr at URL http://www.flickr.com

Tested on 2 May 2012 18:07:35 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	2
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources:	
No of Scripts:4	
No of Images :4	
No of CSS :2	
12. Aggregate Static Images (CSS Sprites):	Inconclusive
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	PASS
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:134

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	PASS
23. Make telephone numbers Click-to-Call:	Inconclusive

24. Ensure Paragraph Text Flows: Visual Check Required
25. Ensure Consistency of state between Devices: Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications: Not Tested
27. Use Meta Viewport Element for Screen Size: PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection: Inconclusive
29. Client Side Detection where appropriate: Inconclusive
30. Use Device Classification to simplify Content Adaptation: Not Tested
31. Support non-Javascript variants: Inconclusive
32. Offer Users a choice of interface: Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on ShoZu Web service at URL http://www.shozu.com

Tested on 2 May 2012 18:08:52 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies 5
2. Use Appropriate Client Side Storage Technology: Inconclusive
3. Replication Local Data to Server: Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information: Visual Check Required
6. Enable Automatic Sign In: Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression Inconclusive
8. Minimize Application and Data Size: Inconclusive
9. Avoid Redirects No Redirect Found
10. Optimize Network Requests: Not Tested
11. Minimise External Resources:
    - No of Scripts:0
    - No of Images :2
    - No of CSS :1
12. Aggregate Static Images (CSS Sprites): Inconclusive
13. Background images inline in CSS: Inconclusive
14. Cache Resources by Fingerprinting: PASS
15. Cache AJAX Data: Not Tested
16. Do not send Cookie Information Unnecessarily: Not Tested
17. Keep DOM size reasonable: Number of DOM Nodes:38

User Experience

18. Optimize Start Up Time (appcache): Inconclusive
19. Minimize Perceived Latency: Visual Check Required
20. Design for multiple Interaction Methods: Visual Check Required
21. Preserve Focus on Dynamic Page Updates: Not Tested
22. Use Fragment IDs to drive Application View: PASS
23. Make telephone numbers Click-to-Call: Inconclusive

24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	Inconclusive

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	Inconclusive
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on iTrailers at URL http://itrailers.info/

Tested on 2 May 2012 18:10:08 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	0
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:3 No of Images :1 No of CSS :1	
12. Aggregate Static Images (CSS Sprites):	Inconclusive
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:39

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive

24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	Inconclusive
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Mobile Photobucket at URL http://www.photobucket.com

Tested on 2 May 2012 18:11:32 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	6
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources:	
No of Scripts:23	
No of Images :14	
No of CSS :3	
12. Aggregate Static Images (CSS Sprites):	Inconclusive
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:168

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive

24. Ensure Paragraph Text Flows: Visual Check Required
25. Ensure Consistency of state between Devices: Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications: Not Tested
27. Use Meta Viewport Element for Screen Size: PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection: Inconclusive
29. Client Side Detection where appropriate: Inconclusive
30. Use Device Classification to simplify Content Adaptation: Not Tested
31. Support non-Javascript variants: PASS
32. Offer Users a choice of interface: Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on World business, finance and political news from the Financial Times - FT.com at URL http://m.ft.com

Tested on 2 May 2012 18:13:25 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	0
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data:	JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	Inconclusive
8. Minimize Application and Data Size:	PASS
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:4 No of Images :5 No of CSS :1	
12. Aggregate Static Images (CSS Sprites):	Inconclusive
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	PASS
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:414

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	PASS

23. Make telephone numbers Click-to-Call: Inconclusive
24. Ensure Paragraph Text Flows: Visual Check Required
25. Ensure Consistency of state between Devices: Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications: Not Tested
27. Use Meta Viewport Element for Screen Size: PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection: Inconclusive
29. Client Side Detection where appropriate: Inconclusive
30. Use Device Classification to simplify Content Adaptation: Not Tested
31. Support non-Javascript variants: Inconclusive
32. Offer Users a choice of interface: Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Mobile Web Page at URL http://www.weather.com

Tested on 2 May 2012 18:14:42 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	0
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:14 No of Images :0 No of CSS :1	
12. Aggregate Static Images (CSS Sprites):	PASS
13. Background images inline in CSS:	PASS
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:50

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive

24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	Inconclusive
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on PocketMarket at URL http://www.pocketmarket.info/

Tested on 2 May 2012 18:20:58 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	0
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data:	JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:2, No of Images :10, No of CSS :2	
12. Aggregate Static Images (CSS Sprites):	PASS
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:132

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive

24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	PASS
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Google Maps at URL http://www.google.com/gmm/index.html

Tested on 2 May 2012 18:49:52 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	1
2. Use Appropriate Client Side Storage Technology:	PASS
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data:	JSON Parser Detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:19; No of Images :19; No of CSS :11	
12. Aggregate Static Images (CSS Sprites):	PASS
13. Background images inline in CSS:	PASS
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:572

User Experience

18. Optimize Start Up Time (appcache):	PASS
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive
24. Ensure Paragraph Text Flows:	Visual Check Required

25. Ensure Consistency of state between Devices: Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications: Not Tested
27. Use Meta Viewport Element for Screen Size: PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection: PASS
29. Client Side Detection where appropriate: Inconclusive
30. Use Device Classification to simplify Content Adaptation: Not Tested
31. Support non-Javascript variants: PASS
32. Offer Users a choice of interface: Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on 4G Network Load Balance and Speed Test - Measure your Broadband, T1, WiFi, 3G and 4G Network at URL http://www.4g-speed.info

Tested on 2 May 2012 19:12:45 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	0
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources:	
No of Scripts:1	
No of Images :0	
No of CSS :1	
12. Aggregate Static Images (CSS Sprites):	Inconclusive
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:26

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive

23. Make telephone numbers Click-to-Call:	Inconclusive
24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	PASS
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Color Mail at URL http://wsidecar.apple.com/cgi-bin/nph-reg3rdpty2.pl/product=23215&cat=68&platform=osx&method=sa

Tested on 2 May 2012 19:15:40 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	0
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	Inconclusive
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources:	
No of Scripts:10	
No of Images :0	
No of CSS :2	
12. Aggregate Static Images (CSS Sprites):	PASS
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:117

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive

23. Make telephone numbers Click-to-Call:	Inconclusive
24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	Inconclusive

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	PASS
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Thesaurus at URL http://www.1webapps.com/apps/utilities/thesaurus/

Tested on 2 May 2012 19:17:12 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	1
2. Use Appropriate Client Side Storage Technology:	Inconclusive
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	Inconclusive
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:22 No of Images :36 No of CSS :1	
12. Aggregate Static Images (CSS Sprites):	PASS
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:272

User Experience

18. Optimize Start Up Time (appcache):	Inconclusive
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive

24. Ensure Paragraph Text Flows:	Visual Check Required
25. Ensure Consistency of state between Devices:	Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications:	Not Tested
27. Use Meta Viewport Element for Screen Size:	PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection:	PASS
29. Client Side Detection where appropriate:	Inconclusive
30. Use Device Classification to simplify Content Adaptation:	Not Tested
31. Support non-Javascript variants:	Inconclusive
32. Offer Users a choice of interface:	Not Tested

Mobile Web Application Checker v0.9 Best Practice Report

Report on Checklist (0) at URL http://alexgibson.github.com/checklist/

Tested on 2 May 2012 19:18:55 GMT using Apple iPhone 4

Application Data

1. Use Cookies Sparingly:#of Cookies	0
2. Use Appropriate Client Side Storage Technology:	PASS
3. Replication Local Data to Server:	Inconclusive

Security and Privacy

4. Do not execute unescaped/untrusted JSON data: JSON Parser NOT detected

User Awareness and Control

5. Use of Personal/Device Information:	Visual Check Required
6. Enable Automatic Sign In:	Visual Check Required

Conversative Use of Resources

7. Use Transfer Compression	PASS
8. Minimize Application and Data Size:	Inconclusive
9. Avoid Redirects	No Redirect Found
10.Optimize Network Requests:	Not Tested
11. Minimise External Resources: No of Scripts:15; No of Images :0; No of CSS :0	
12. Aggregate Static Images (CSS Sprites):	Inconclusive
13. Background images inline in CSS:	Inconclusive
14. Cache Resources by Fingerprinting:	Inconclusive
15. Cache AJAX Data:	Not Tested
16. Do not send Cookie Information Unnecessarily:	Not Tested
17. Keep DOM size reasonable:	Number of DOM Nodes:50

User Experience

18. Optimize Start Up Time (appcache):	PASS
19. Minimize Perceived Latency:	Visual Check Required
20. Design for multiple Interaction Methods:	Visual Check Required
21. Preserve Focus on Dynamic Page Updates:	Not Tested
22. Use Fragment IDs to drive Application View:	Inconclusive
23. Make telephone numbers Click-to-Call:	Inconclusive

24. Ensure Paragraph Text Flows: Visual Check Required
25. Ensure Consistency of state between Devices: Not Tested
26. Consider Mobile Specific Technology to initiate Web Applications: Not Tested
27. Use Meta Viewport Element for Screen Size: PASS

Handling Variations in the Delivery Context

28. Prefer Server Side Detection: PASS
29. Client Side Detection where appropriate: Inconclusive
30. Use Device Classification to simplify Content Adaptation: Not Tested
31. Support non-Javascript variants: Inconclusive
32. Offer Users a choice of interface: Not Tested