UNIVERSIDAD
REY JUAN CARLOS

Free Libre Open Source Software Master

Academic Year 2012/2013

Master Thesis

Trade-Off Analysis of Open Source Web Mobile App Frameworks: The KuDo Project

Author: Esther Parrilla-Endrino

Tutors: Prof. Gregorio Robles, Prof. Daniel Izquierdo

Contents

Summary

In my daily work I play the role of Technical Manager [1] who coordinates the development of both standalone and web solutions for the Aerospace field, therefore I have no possibility of working in projects related to mobile apps implementation which is one of the most interesting activity nowadays.

Beginning this year I have started working on a personal project called 'KuDo', the idea is to setup a start-up focused in the development of web-based applications for educational purposes that can be easily run in mobile devices such as smartphones, tablets etc...

This Master Thesis is the baseline activity I have performed for the KuDo project in order to determine which technical solution could fit better for this purpose from the wide range of Web Mobile App Frameworks existing nowadays.

All the work done for this Master Thesis is Open Source material licensed

[1] http://www.linkedin.com/in/eparrilla

under a Creative Commons Attribution 3.0 License[2], it has been developed using GitHub repository and can be downloaded from the following url:

`https://github.com/eparrillae/eparrillae-mswl-thesis.git`

Also the different activities performed in this Master Thesis have been described in my personal blog and can be found under the following tag:

`http://eparrillae.net/wordpress/?tag=mswl-thesis`

I would like to thank my Master Thesis tutors Gregorio Robles[3] and Daniel Izquierdo[4] for guiding the whole process, always providing good advices and added-value to this work.

Finally I would like to specially thank my colleague Solange Molina Urrutia[5] who has been a reference for me in this work due to her huge background expertise in Mobile App developments.

[2]http://creativecommons.org/licenses/by/3.0/
[3]http://gsyc.urjc.es/ grex/
[4]http://libresoft.es/publications/author/17
[5]http://www.linkedin.com/in/smolina

Chapter 1

Overview and Objectives

Nowadays mobile devices such as smartphones and tablets have become the preferred tools for communicating with each other, for that reason there are lots of technical solutions for implementing applications that can be run under those devices and that offer a clear enhancement in the end-user experience. That is the reason why the KuDo project is focused in these kind of developments, there is a clear niche of new opportunities in this software development field.

Mobile Apps can be grouped in two categories:

- **Native Apps**: A native app is an app for a certain mobile device (smartphone, tablet, etc.) They are installed directly onto the device. Users typically acquire these apps through an online store or market-

place such as The Apple Store[1] or Android Apps on Google Play[2] .

- **Web-based Apps**: When we talk about mobile web apps we are referring to Internet-enabled apps (compliant with HTML5[3] , CSS3[4] and Javascript[5] standards) that allow web developers to quickly and easily create mobile apps that work on Android, iOS and BlackBerry devices, and produce a native-app-like experience inside a browser.

For the KuDo project I had to decide which type of Mobile Apps I would develop, table below summarize using a SWOT[6] analysis the advantages and disadvantages of Native Apps:

[1]https://www.apple.com/iphone/from-the-app-store/
[2]https://play.google.com/store/apps
[3]http://www.w3.org/html/wg/drafts/html/master/
[4]http://www.w3.org/TR/CSS/
[5]http://www.w3.org/standards/webdesign/script.html
[6]https://en.wikipedia.org/wiki/SWOT_analysis

	Strengths	**Weaknesses**
Internal	Standardized software development kits (SDKs) are often provided , Can interface with the device's native features (camera, accelerometer etc...), Installed and runs as a standalone application (no web browser needed), There are stores and marketplaces to help users find your app, Typically perform faster than mobile web apps (being native code), App store approval processes can help assure users of the quality and safety of the app	Each development platform (e.g. iOS, Android) requires its own development process, Each development platform has its own native programming language, Users must manually download and install app updates, Are typically more expensive to develop, Supporting multiple platforms requires maintaining multiple code bases, Users can be on different versions making your app harder to maintain, App store approval processes can delay the launch of the app
	Opportunities	**Threats**
External	Powerful apps that use all devices' potential and bring business opportunities, Developers have the ability to charge a download price and app stores will typically handle the payment process	Mobile-specific ad platforms (e.g. AdMob[7]) can include restrictions[8] set by the mobile device's manufacturer, Supported by less forges in the future

Table below summarize using a SWOT analysis the advantages and disadvantages of Web-based Apps:

	Strengths	**Weaknesses**
Internal	Mobile web apps use standard languages such as HTML5, CSS3 and Javascript, Accessed through a mobile device's web browser no need to install new software , Updates are made to the web server without user intervention, All users are on the same version no maintenance issues, Have a common code base across all platforms, Can be released in any form and any time as there is not an app store that has to approve the app, If you already have a web app, you can retrofit it with a responsive web design	Runs in the mobile device's web browser and each may have its own features and quirks, There are no standard software development kits (SDKs), Can access a limited amount of the device's native features and information, Since there is no app store for the Mobile Web, it can be harder for users to find your app
	Opportunities	**Threats**
External	Fast development of lightweight user-friendly apps, Mobile web apps can monetize through site advertisement and subscription fees	Always needs Internet connection, Charging users to use the mobile web app requires you to set up your own paywall or subscription-based system

To help me decide which type of solution I should take for my KuDo project I asked myself the following questions:

- **Does the mobile app require the use of any special device features (i.e., camera, the camera's flash, accelerometer, etc.)?** In priciple my idea is to implement simple apps with a high level of responsiveness[9] from the user's point of view but the access to the device hardware components is not a must.

- **What is my budget?** Very limited :)

- **Does the mobile app need to be Internet-enabled?** This is not a must for the first applications that I plan to develop in the context of KuDo project but for sure in future versions it would be interesting to be able to link the apps with external educational resources such as Wikipedia, museums, libraries etc...

- **Do I need to target all mobile devices or just certain devices?** Portability is a key issue, the apps should be supported by most of the devices and the idea of having several codebases does not fit in KuDo project.

- **What programming languages do I already know?** Here I have to say that my programming background expertise is more oriented to backend developments and therefore I am ready to start implementing low-level tools using Native apps but in the other side for me it is a good chance to improve my knowledge in web-developments which is a field I cannot explore in my daily professional work.

- **How important is speed and performance?** It is important both using Native and Web-based apps.

[9] https://en.wikipedia.org/wiki/Responsiveness

- **How will this app be monetized effectively?** I have some experience in setting up Paypal[10] systems but I would need some help with this issue, again this is an interesting challenge for me.

Summarizing, due to the type of developments I am planing to do I think the Web-based Apps solutions are the ones that best fit these purposes even though I am aware that I shall have to make an effort in learning more on HTML5, CSS3, Javascript and other web technologies but I see this as an interesting challenge that shall improve my professional background.

In this Master Thesis I am doing the exercise of performing a deep search in the existing FLOSS Web-based App frameworks for mobile devices, selecting a couple of the most popular ones and then comparing their capabilities using a well-formed quality system like OpenBRR.

The fact that all frameworks taken under consideration in this study are Open Source solutions is a consequence of different subjects I studied during the Master Thesis and the idea that this kind of solutions bring more benefits to the development community instead of closed solutions.

The objectives of this Master Thesis were defined and agreed with my tutors and can be summarized in the following list:

- Select a couple of representative Open Source web-based frameworks for developing Mobile Apps from the existing solutions currently available in the market.

[10] https://www.paypal.com/es/webapps/mpp/home

- Set a checklist of points to be analyzed for each solution in the form of well-defined metrics that can be properly measured.
- Analyze each solution using the OpenBRR methodology, setting different weights and scores for each category according to this model. Spreadsheets shall be used for automating this process.
- Summarize pros and contras of each solution.
- Perform a comparison of the results for each solution taking as a baseline the final version of each spreadsheet and also applying a SWOT comparision analysis on top of OpenBRR.

Chapter 2

Selected Frameworks

Chapter 3

OpenBRR Analysis

3.1 What is OpenBRR?

The Business Readiness Rating model (OpenBRR)[1] is intended to help IT managers assess which Open Source software would be most suitable for their needs. Open Source users can also share their evaluation ratings with potential adopters, continuing the virtuous cycle and "architecture of participation" of open source.

The initiative is lead by the Carnegie Mellon West University, Spike Source, Intel and O'Reilly's Code Zoo and offer proposals for standardizing different types of evaluation data and grouping them into categories.

The framework suggests the following metrics to be analyzed and evalu-

ated:

- Functionality
- Usability
- Quality
- Security
- Performance
- Scalability
- Architecture
- Support
- Documentation
- Adoption
- Community
- Professionalism

The model is composed of the following phases:

- Phase 1 - Quick Assessment: defining and ranking of metrics and categories according to their importance within the product that is going to be evaluated.

- Phase 2 - Target usage assessment: Set the necessary category and metric weights according to the project's goals.

- Phase 3 - Data collection and processing: Gather data for each metric used in each category rating, and calculate the applied weighting for each metric, spreadsheets are used for this purpose.

- Phase 4 - Data translation: Use category ratings and the functional orientation weighting factors to calculate the Business Readiness Rating score and publish the software's Business Readiness Rating score.

The Business Readiness Rating model offers a trusted and open framework for software evaluation, this model aims to accelerate the software evaluation process with a systematic approach, facilitate the exchange of information between IT managers, result in better decisions, and increase confidence in high-quality open source software.

3.2 Data Sources

For analyzing the different web-based Mobile App frameworks and provide a reliable comparison, data from different primary sources will be used. In this section I present the most important ones; other sources will be referenced or explained directly (for example using a footnote) when the corresponding fact, metric or argument is discussed.

The starting point for getting data on the selected solutions are the official

websites of jQuery Mobile[2] and Sencha Touch[3] , they provide reliable information about their functionality, history, documentation and support.

One of the data sources we studied in the MSWL was Ohloh[4] which is a free public directory of open source software and people, owned and operated by BlackDuck Software Inc., a consulting company specialized in gathering and providing information about open source software projects. Some metrics present in Ohloh site are provided by Ohloh specific tools and others are provided by gathering the information provided by Ohloh users for their own projects or the projects they are interested in.

Both jQuery Mobile and Sencha Touch are sub-projects of other parent projects (jQuery and Sencha), in Ohloh we can find information on both the parent and child projects:

- For jQuery Mobile there is a dedicated page in Ohloh[5] that provides very valuable information about the project for our trade-off analysis. Also the parent project jQuery[6] and other related extensions have their own pages in Ohloh data source.
- For Sencha Touch the situation is not the same, even though we can find an entry in Ohloh for the parent Sencha project[7] the page has no activity since a long time, therefore unfortunately we cannot take Ohloh as data source for finding information in the case of Sencha Touch and we shall have to use other sources.

In the MSWL we studied other data sources like FLOSSMetrics[8] ,

FLOSSmole[9] and FLOSShub[10] which provide centralized access to data analysis (charts, tables and other quantitative information) of free/libre/open source projects hosted in forges such as Sourceforge[1] , GForge[2] etc... Also we saw the FLOSSpapers project[11] allows to perform queries on papers published on these purposes.

Unfortunately because the technologies we are studying in this thesis are quite new there is no data for both jQuery Mobile and Sencha Touch project in none of the data sources cited above so I had to look for other sources of information.

One of the most interesting data sources nowadays is LinkedIn[3] a social networking website for exchanging profesisonal information, there are lots of groups devoted to Mobile Apps technologies, I have performed a seach in several of those groups looking for information on jQuery Mobile and Sencha, specially in the "iPhone, Android, iPad, Tablet & Mobile Application Development"[12], the "Mobile Software Development Group"[13] and the "Developers - HTML5, Android, iOS, Windows, Java, BlackBerry,..."[14] groups which are very active.

LinkedIn allows users to start discussions and polls in the groups they are subscribed to...

Also Google discussion groups[**?**] which contain a searchable archive of more than 700 million Usenet postings from a period of more than 20 years

[1] https://sourceforge.net/
[2] http://gforge.org/gf/
[3] https://www.linkedin.com/

is a valuable source of information.

Regarding documentation available for each project, Amazon[**?**] is the reference data source I have used.

Chapter 4

Trade-Off Results

I am going to follow the "Business Readiness Rating for Open Source (OpenBRR)" whitepaper[1] to apply this model to the evaluation of the different solutions.

4.1 Phase 1: Quick Assessment

This first phase has been applied by performing a deep search of the existing HTML5 frameworks for developing mobile apps through the Internet.

The following frameworks have been selected among the most popular existing ones:

- jQuery Mobile: `http://jquerymobile.com/`
- Sencha Touch: `http://www.sencha.com/products/touch`

4.2 Phase 2: Target Usage Assessment

This second phase consists in setting the category and metric weights according to our requirements. The canonical OpenBRR model recommends to focus in not more than seven categories, but in order to provide a more general overview, I will consider the twelve categories present in the model.

If we were considering all the categories equal in importance, we should weight each one of them with 8,33%. Our assessment will consider this number, in order to weight more than 8% the categories considered relevant for the company, and less than 8% the categories considered not so relevant for the company.

The most important selected categories have been **functionality**, **usability** and **community**. Each one of them have been given a weight of 12%, so together they reach 36% of the total evaluation.

The OpenBRR model provides no ready-to-collect metrics for **functionality**, allowing the evaluator to create them in a tailored way according to the customer's requirements.

Support and **documentation** are also desirable aspects, that ensure

the liveness of the community of any piece of software, and also guarantee usability since good instructions and advices smooth out the difficulties of any tool. For this reason these two categories have been weighted with 10%. With the same arguments we could consider **adoption**, but we also need to know that there are two influent factors in adoption: on one hand, we need time for any tool to be widely used. On the other hand, "trends" have also influence in the IT world; and certain companies or tools come in a particular time to the crest of the wave, but quickly sink into obscurity due to the dynamism of the technologies environments. So adoption have been scored with 9%, still over the mean, but not so much.

About **security**, the given weight has been 8% as we have not defined specific requirements on this purpose, but it is a desirable feature specially for the future when new developers come to the community.

Performance and **architecture** are two categories weighted under the mean (6%)

The less important categories for this evaluation are **quality**, **scalability** and **professionalism**. These categories have been weighted with 5%, which makes a sum of 15% of total evaluation.

In conclusion, in table 4.1 I present the categories and their resulting weights for our evaluation.

Rank	Category	Weight
1	Functionality	12%
2	Usability	12%
3	Quality	5%
4	Security	8%
5	Performance	6%
6	Scalability	5%
7	Architecture	6%
8	Support	10%
9	Documentation	10%
10	Adoption	9%
11	Community	12%
12	Professionalism	5%
	TOTAL WEIGHT	**100%**

Table 4.1: OpenBRR Target Usage Assessment for HTML5 mobile apps frameworks

4.3 Phase 3: Data collection and processing

For filling the differents scores assigned to each category defined previously I have used the OpenBRR baseline spreadsheet provided to the students of Master on Libre Software 2011-2012 located at:
`http://docencia.etsit.urjc.es/moodle/mod/resource/view.php?id=4350`

For more information about this topic you can visit the MSWL Project

Evaluation Subject's Moodle site in:
`http://docencia.etsit.urjc.es/moodle/course/view.php?id=125`.

This spreadsheet has an initial set of metrics for each OpenBRR category, allowing to ponderate each metric and providing a normalized score according to the possible values obtained in measurements.

Category weights have been introduced in the sheets. Each metric within each category should have a weighting factor to differentiate the metric's importance withing that particular category.

Each metric has been measured searching the Internet and getting the needed information from official mailing lists or websites and referencing that link in the corresponding "Raw score" cell with a "comment" in the cell. When a reference is not provided, it means that that metric could not be found or the own tool command line help or main website announces that aspect so it is easy to find.

For the unknown data, I have assigned the worst possible normalized score to the corresponding metric, so the results is not biased by unreliable information.

4.4 Phase 4: Representative Metrics and their Scoring

After collecting all the data and normalizing using the OpenBRR spreadsheet, scores for each category and a global score is automatically calculated. The resulting work can be downloaded from this URLs:

- jQuery Mobile spreadsheet: `https://github.com/eparrillae/eparrillae-mswl-the tree/master/MasterThesis/thesis/OpenBRR_Templates/BRR_Template_jQuery.ods`
- Sencha Touch OpenBRR spreadsheet: `https://github.com/eparrillae/eparrillae-mswl-thesis/tree/master/MasterThesis/thesis/OpenBRR_Templates/BRR_Template_Sencha.ods`

Chapter 5

Conclusions

Chapter 6

Future Work

Appendix A

Appendix A

Appendix A

Appendix B

Bibliography

[1] **OpenBRR: Business Readiness Rating for Open Source (White paper)**

http://docencia.etsit.urjc.es/moodle/mod/resource/view.php?id=4343

[2] **jQuery Mobile**

http://jquerymobile.com/

[3] **Sencha Touch**

http://www.sencha.com/products/touch

[4] **Ohloh**

http://www.ohloh.net/

[5] **Ohloh jQuery Mobile**

https://www.ohloh.net/p/jquerymobile

[6] **Ohloh jQuery**

https://www.ohloh.net/p/jQuery

[7] **Ohloh Sencha**

https://www.ohloh.net/p/sencha

[8] **FLOSSMetrics**

http://flossmetrics.org/

[9] **FLOSSmole**

http://flossmole.org/

[10] **FLOSShub**

http://flosshub.org/

[11] **FLOSSpapers**

http://flosshub.org/biblio

[12] **LinkedIn1**

http://www.linkedin.com/groups/iPhone-Android-iPad-Tablet-Mobile-2013391

[13] **LinkedIn2**

http://www.linkedin.com/groups/Mobile-Software-Development-Group-69893

[14] **LinkedIn3**

http://www.linkedin.com/groups?home=&gid=54723&trk=anet_ug_hm