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

KUDO
PROJECT

Author: Esther Parrilla-Endrino

Tutors: Prof. Gregorio Robles, Prof. Daniel Izquierdo

Contents

1 Overview and Objectives **9**

2 Selected Frameworks **17**

3 OpenBRR Analysis **19**

3.1 What is OpenBRR? 19

3.2 Data Sources 21

4 Trade-Off Results **27**

4.1 Phase 1: Quick Assessment 27

4.1.1 Functionality Category metrics definition 28

4.1.2 OpenBRR improvements with new metrics 28

4.2 Phase 2: Target Usage Assessment 36

4.3 Phase 3: Data collection and processing 38

4.4 Phase 4: Data Translation 40

4.4.1 jQuery Mobile Results 40

4.4.2 Sencha Touch Results 46

5 Conclusions **55**

6 Future Work **59**

7 Apendix A: jQuery Mobile ”Hello World” Prototype 61

8 Apendix B: Sencha Touch ”Hello World” Prototype 63

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[1] :

- **Native/Hybrid 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

[1] https://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Deve

marketplace such as The Apple Store[2] or Android Apps on Google Play[3] .

- **Web-based Apps**: When we talk about mobile web apps we are referring to Internet-enabled apps (compliant with HTML5[4] , CSS3[5] and Javascript[6] 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[7] analysis the advantages and disadvantages of Native Apps:

[2] https://www.apple.com/iphone/from-the-app-store/

[3] https://play.google.com/store/apps

[4] http://www.w3.org/html/wg/drafts/html/master/

[5] http://www.w3.org/TR/CSS/

[6] http://www.w3.org/standards/webdesign/script.html

[7] 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[8]) can include restrictions[9] 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[10] 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.

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

- **How will this app be monetized effectively?** I have some experience in setting up Paypal[11] 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.

[11] 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[2] 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 evaluated:

- 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/secondary 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[3] and Sencha Touch[4] , they provide reliable information about their functionality, history, documentation and support.

One of the data sources we studied in the MSWL was Ohloh[5] 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[6] that provides very valuable information about the project for our trade-off analysis. Also the parent project jQuery[7] 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[8] 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 because its code reposiroty is not open to the public and we shall have to use other sources.

In the MSWL we studied other data sources like FLOSSMetrics[10] , FLOSSmole[11] and FLOSShub[12] 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[13] 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"[14], the "Mobile Software Development Group"[15] and the "Developers HTML5, Android, iOS, Windows, Java, BlackBerry"[16] groups which are very active. LinkedIn allows users to start discussions and polls in the groups they are subscribed to.

Another interesting tool I have discovered during my thesis is Survey Monkey[4] ,a tool for publising quite complete polls for free, the system is

[1] https://sourceforge.net/

[2] http://gforge.org/gf/

[3] https://www.linkedin.com/

[4] http://es.surveymonkey.com/

more complete than other solutions like LinkedIn polls system.

Also for searching information on some metrics included in this trade-off study I have used Quora[5] wuich is a question-and-answer website wich lots of information on the Mobile Web Apps field.

Also Google discussion groups[6] which contain a searchable archive of more than 700 million Usenet postings from a period of more than 20 years is a valuable source of information.

A pretty good resource for searching useful articles about Mobile Web Apps development (and in general any kind of development technologies) is IBM Developer Works site that has dedicated sections for different frameworks like Dojo JS[7] , jQuery Mobile[8] , PhoneGap[9] , Sencha Touch[10] etc...

Regarding books available for each project, Amazon[11] is the reference data source I have used.

Regarding demos/presentations I have used Slideshare as main data source, there are many resources available for jQueryMobile[12] and Sencha Touch[13] there.

[5] https://www.quora.com/

[6] https://groups.google.com

[7] https://www.ibm.com/developerworks/topics/dojo

[8] http://www.ibm.com/developerworks/topics/jquerymobile

[9] https://www.ibm.com/developerworks/topics/phonegap

[10] http://www.ibm.com/developerworks/topics/sencha%20touch

[11] http://www.amazon.com

[12] http://www.slideshare.net/search/slideshow?searchfrom=header&q=jquery+mobile

[13] http://www.slideshare.net/search/slideshow?searchfrom=header&q=sencha+touch

Also for having a better idea on measuring some of the metrics related to the functionalities provided by each framework like how easy is the installation process, available IDEs, portability to different mobile devices etc... I have developed a couple of prototype examples one for jQuery Mobile and another for Sencha Touch, the source code of these examples can be found in the thesis Github repository under:

`https://github.com/eparrillae/eparrillae-mswl-thesis/tree/master/MasterThesis/prototypes`

Chapter 4

Trade-Off Results

As stated before, 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 I have selected: jQuery Mobile and Sencha Touch.

4.1 Phase 1: Quick Assessment

This first phase is focused in setting the number of categories used and the different metrics that shall be evaluated per category. The canonical OpenBRR model recommends to focus in not more than seven categories, but in order to provide a more general overview, I have considered the twelve categories present in the model.

Regarding the metrics, I have reviewed the list of metrics provided by the canonical OpenBRR model template, if these metrics fit well with the type of tools I have to analyze (Mobile Web App) I have left them as they are but in some cases I have substituted the default metrics with new ones that measure better the quality of the products.

4.1.1 Functionality Category metrics definition

The most important category within this study is the "Functionality" category, here I have defined a set of very specific criteria to evaluate precisely the selected frameworks, to be able to define a complete subset of metrics I have used two different sources:

- The experience analyzing the available solutions in the market for developing Mobile Web Apps done in previous section "Selected Frameworks".
- The experience developing jQuery Mobile and Sencha Touch prototypes.
- Wikipedia comparison tables between web-based Mobile App frameworks[17] and HTML5/CSS3/Javascript frameworks[18].

4.1.2 OpenBRR improvements with new metrics

The following new metrics have been added to this study:

Multiplatform support

I have added this metric in the **Usability** category, one of the key points of the development of KuDo projects is that the applications can be run in the most popular mobile platforms used nowadays, therefore, it is quite important to ensure that the salected framework runs well in different web browsers and mobile devices containing those web browsers.

The score assigned to this metric will be:

- Good Support: 5
- Regular Support: 3
- Poor Support: 1

Is there a dedicated information (web page, wiki, etc) for security?

I have added this metric in the **Security** category, even though security is not a key aspect for the KuDo project because the applications that shall be deployed are going to be very simple ones with no authentication procedures it is always interesting which support is provided by default regarding this aspect.

The score assigned to this metric will be:

- Yes, well maintained: 5

- Yes: 3
- No: 1

Architecture based on design patterns

I have added this metric in the **Scalability** category, the idea is to check if the selected framework is built on top of a robust architecture that can be extended in the future, for that reaosn the usage of design patterns is so important.

The score assigned to this metric will be:

- Yes, architecture fully based on standard design patterns: 5
- Yes, partially: 3
- No: 1

Are there any repositories of 3rd party UI Plug-ins

I have added this metric in the **Architecture** category, for any kind of User Interface development it is quite important to have available a good toolkit of UI widgets coming by default with the selected solution, but also if there are external repositories containing widgets implemented by 3rd party developers this helps a lot in extending the capabilities provided by the applications.

The score assigned to this metric will be:

- more than 5 repositories available: 5
- 1 to 5 repositories available: 3
- 0 repositories available: 1

Backend Services Support

I have added this metric in the **Architecture** category, the idea is to measure the availability of pluggable backend libraries for connecting with Databases, external File Systems, Web Services etc...

The score assigned to this metric will be:

- Yes, extensive: 5
- Yes: 3
- No: 1

Enable/disable features through configuration

I have added this metric in the **Architecture** category, the idea is to measure how easily the framwork can be configured.

The score assigned to this metric will be:

- Yes, during runtime: 5
- Yes, during runtime: 3
- No: 1

Programming Languages

I have added this metric in the **Architecture** category, as we saw during the "Communities" subject of the MSWL for a given development the support of more languages in the core source code of the framework means more flexiblity and also more people may get involved in the project, if the project is completely built based in just one programming language is possible that a developer who does not know about the language used on it cannot collaborate.

But also is important the amount of code used in a concrete programming language, it would not be good if we say that a FLOSS project uses three different programming languages when one of them just has a few lines of code.

The score assigned to this metric will be:

- 80% or more of SLOC in three or more languages: 5

- 80% or more of SLOC in two languages: 3
- 80% or more of SLOC in one language: 1

Framework web site quality

I have added this metric in the **Documentation** category, a good web site should provide a centralized point for getting information on the project's objectives, license, access to source code, forums, mailing lists etc...

Also the information should be provided for the various end-users roles:

- users with no experience using Mobile App frameworks
- users with experience using Mobile App frameworks but with no experience using this particular framework
- developers of the framework
- project administrators

The score assigned to this metric will be:

- Superb web site: 5
- Acceptable web site: 3
- Poor web site: 1

Longevity

A new metric is introduced in the **Community** category. I think that the longevity of a given project must be measured combining two different criteria, in one hand the project's age, but also it has to be analyzed if during the whole project lifecycle there has been development activity, a very old project with just a few activity is really useless for the KuDo project purposes.

Taking into account the previous assertions the normalized scores to assign to this metric will be:

- Young project (more than 1 year old) and increasing lines of code production : 1
- Medium project (more than 5 years old) and increasing lines of code production : 3
- Long project (more than 10 years old) and increasing lines of code production: 5

Cloud Support

A new metric is introduced in the **Community** category. This metric is related with the feature to provide backend-as-a-service capabilities that allows developers to access a set of APIs that helps them to build, run and test their applications in a cloud environment.

The score assigned to this metric will be:

- Full support: 5
- Partial Support : 3
- No Support: 1

License

A new metric is introduced in the **Professionalism** category. For the KuDo project it is a must that the Mobile App solutions is licensed under an OSI-approved[1] FLOSS license. Also if the license allows to mix the product with proprietary code is an enhancement.

Taking into account the previous assertions the normalized scores to assign to this metric will be:

- Open Source but Non-OSI-Approved license: 1
- OSI-Approved and "GPL-like" license: 3
- OSI-Approved and "weak copyleft" license: 5

[1] http://www.opensource.org/licenses

4.2 Phase 2: Target Usage Assessment

This second phase consists in setting the category and metric weights according to the KuDo project requirements. The canonical OpenBRR model recommends to focus in not more than seven categories, but as I said before, 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 KuDo project needs, and less than 8% the categories considered not so relevant.

The most important selected categories have been **Functionality** and **Usability**. 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, in previous section "Functionality Category metrics definition" I have explained the criteria followed to define the list of metrics.

Support, **Documentation** and **Community** 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 categories have been weighted with 10%.

With the same arguments I have evaluated **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.

Architecture and **Scalability** are also key aspects of this trade-off, as I are looking for a framework that serves as a baseline for the developments done in the context of the KuDo project, this framework should provide an architectural design modular and flexible enough to allow the integration of new components to the "core" in an easy way.

About **Quality**, **Security** and **Performance** the given weight has been 5% as I have not defined yet strong requirements on this purpose, but they desirable features specially for the future of the project when the mobile apps to be developed become more and more complex.

Finally **Professionalism** has been given a weight of 5%, it is important that the selected framework is supported by a robust community and in the particular case of jQuery Mobile and Sencha Touch both are sub-projects that were created from a bigger parent project (jQuery and Sencha) which ensures its stability.

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	5%
5	Performance	5%
6	Scalability	8%
7	Architecture	8%
8	Support	10%
9	Documentation	10%
10	Adoption	10%
11	Community	10%
12	Professionalism	5%
	TOTAL WEIGHT	**100%**

Table 4.1: OpenBRR Target Usage Assessment for web-based Mobile App 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: Data Translation

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 is summarized in the following subsections:

4.4.1 jQuery Mobile Results

The OpenBRR spreadsheet containing the metrics final scores and comments justifying those scores can be found here:

`https://github.com/eparrillae/eparrillae-mswl-thesis/blob/master/MasterThesis/thesis/OpenBRR_Templates/BRR_Template_jQueryMobile.ods`

The final score for jQuery Mobile is 4.0015, which is a very high score, that means that this framework covers most of the needs required for the KuDo development project.

In the **Functionality** category we can see that most of the requirements are covered, the framework seems to be strong in areas like easy to install and use, multiplatform support, compliance with HTML, CSS3 standards for the web, themes customization and availability of 3rd party plugins/extensions. Also the development environment which is a key part of the implementation process is fully covered, even though there is no dedicated plugin for Eclipse IDE which is the one I use it is possible to do this integration through Aptana IDE[2] .

The weakest points in the case of functionality metrics are centralized in the access to native features of the mobile device that has to be done through other tools like [3] and the support of some advances features like media audio/video and look&feel stencils which is very poor at the moment.

In the **Usability** category again we can see that the framework is easy to install and use in a wide range of devices, this is very important for the KuDo project, portability problems could be an issue and the selected solution should cope with that in the proper way.

For measuring the **Quality** category metrics we have accessed jQuery Mobile GitHub repository[19] where we can find useful information about developers involved in the project, project milestones and roadmap, bugs/enhancements raised in the code etc...

In terms of major releases jQuery Mobile is in the range of 1 to 3 major releases which means that teh project is active but its community still is not very big, the latest release is from the 20th of Februarty 2013 (v1.3.0) so we have a very recent version to be downloadable and installed.

The activity of fixing bugfixes in the code repository that is measured

[2] http://stackoverflow.com/questions/4721124/how-to-enable-jquery-support-in-aptana-studio-3

[3] http://phonegap.com/

by the rest of metrics in this category clearly show that the jQuery Mobile community is alive, this is very important for the KuDo project as I would like to take a baseline solution which is alive and garantees the close future of teh project.

In the **Security** category we can see in Github repo[4] that there are no major open issues in terms of security, the framework is very simple and there are no specific packages dedicated to security. Also in this category we have the new metric that checks if there is a dedicated web page for security issues, we see that there are no specific documentations on this regard.

In the **Performance** category we see some of the weakest points of this framework, there are no formal reports about the performance of the framework in different devices, also jQuery does not provide any benchmark/sandbox for testing purposes, whihc means that the developer itself shall have to write from scratch the proper unitary/integration tests which is a clear pitfall of this solution. Also I could not find any wide documentaiton about tunning performance, for that reason this framework has slow scores in this category.

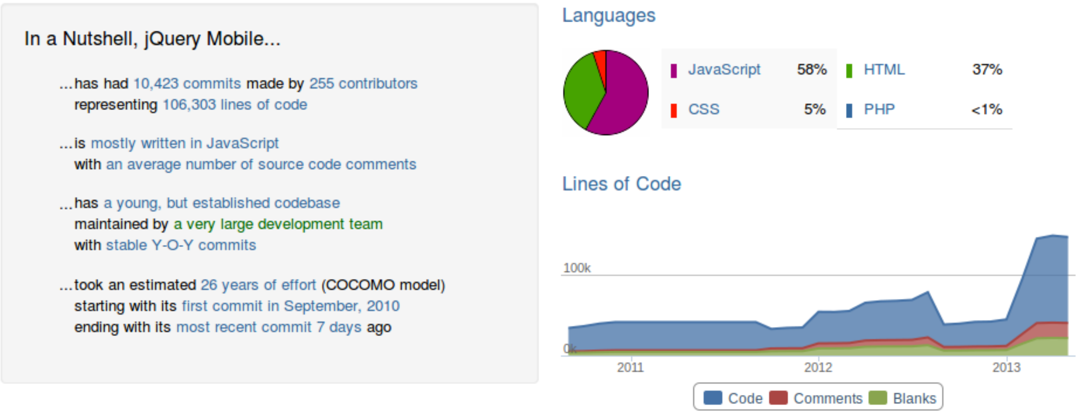
In the **Scalability** category it is clear that jQuery Mobile is based on a pluggable architecture which is quite extensible but the usage of standard design patterns is not provided by default just the Abstract Factory design pattern is used for its UI widgets toolkit, other patterns should be built by yourself from scratch like for example the implementation that the M

[4] https://github.com/jquery/jquery-mobile/issues?labels=4+-+High&page=1&state=open

Project[5] has done.

In the **Architecture** category we see that there are wide range of 3rd party plugins and extensions[20] available which is quite importante for the KuDo project, unfortunately most of those resources are related with the presentation layer of the applications and the support for plugins to connect backend services such as databases, web services etc... is not fully covered.

In terms of programming languages figure below shows the results of executing sloccount[6] over jQuery Mobile code in Ohloh statistics page:



In a Nutshell, jQuery Mobile...

...has had 10,423 commits made by 255 contributors
representing 106,303 lines of code

...is mostly written in JavaScript
with an average number of source code comments

...has a young, but established codebase
maintained by a very large development team
with stable Y-O-Y commits

...took an estimated 26 years of effort (COCOMO model)
starting with its first commit in September, 2010
ending with its most recent commit 7 days ago

Languages

JavaScript	58%	HTML	37%
CSS	5%	PHP	<1%

Lines of Code

Figure 4.1: *jQuery Mobile Programming Languages*

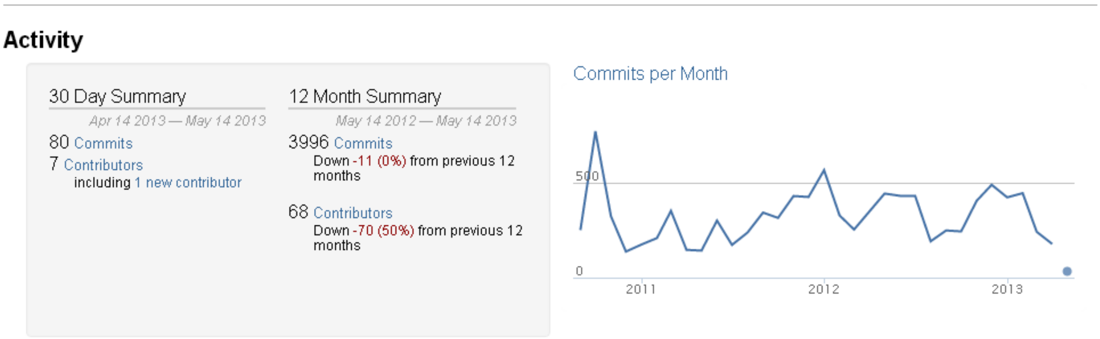
We see that most of the code is done using Javascript/HTML, which means that for implementing Mobile Apps using this framework requires a huge background experience in these languages, this could be a challenge for me as my experience is not so deep in that field.

[5] http://www.the-m-project.org/
[6] http://www.dwheeler.com/sloccount/

In the **Support** category we can clearly see that the jQuery Mobile community is very active and the mailing lists work very well but if we are looking for any kind of professional support that capability is missing within the project, for that reason we conclude that for using jQuery Mobile as baseline for KuDo developments it is necessary to have some background in web-based software development.

Regarding **Documentation** category I have to say that the project is really well-documented, there are lots of books about jQuery Mobile and the web site contains all necessary resources for both developers without experience and advanced ones, in this category all scores are high. This applies also for **Adoption** category, there are many real-world examples of Mobile Apps developed using jQuery Mobile framework.

In the **Community** category we see that average volume of messages in the mailing lists in the last six months is high, also the number of commits in the last month is good but the number of contributors in this period is still low as shown in figure below:



Activity

30 Day Summary
Apr 14 2013 — May 14 2013
80 Commits
7 Contributors
including 1 new contributor

12 Month Summary
May 14 2012 — May 14 2013
3996 Commits
Down -11 (0%) from previous 12 months

68 Contributors
Down -70 (50%) from previous 12 months

Figure 4.2: *jQuery Mobile Monthly Statistics in Ohloh*

The fact that the number od contributors to the 'core' is still low can be a problem to guarantee the sustainability of the project in the future and must be taken into account.

In terms of Longevity, jQuery is considered still a young project but taking into account that it is supported by its parent project jQuery[7] managed by the jQuery Foundation[21] which is very active I think the future of the project is ensured.

In the **Professionalism** category the role of the jQuery Foundation is crucial for organizing conferences and workshops that promote the framework among its users community (next jQuery Conference to be held in June 2013[8]).

Also the community around jQuery Mobile is very welcome to receive new contributors[9] which makes easier the process of participating in the project which is something I would really like to do.

Finally jQuery Mobile is licensed under a MIT license[10] which is perfect for KuDo project purposes, my idea is to integrate in jQuery Mobile other open source plugins and also implement on top of the framework my own developments.

[7] http://jquery.com/

[8] http://events.jquery.org/2013/portland/

[9] http://forum.jquery.com/topic/how-can-i-contribute-to-jquery-mobile-core-development

[10] http://opensource.org/licenses/MIT

Table below summarize using a SWOT analysis the advantages and disadvantages of jQuery Mobile after the OpenBRR analysis:

	Strengths	**Weaknesses**
Internal	Easy of install and use, Wide range of 3rd party plug-ins/extensions, Weak-copyleft MIT license, Strong Open Source community behind, Pretty good documentation, books, articles etc..., Code repository available in Github contributors can join to the core team	No Performance Testing Reports and Benchmarks available, Access to native features through PhoneGap, No support for some advanced features video/audio/stencils/backend services, No design patterns by default, No professional support
	Opportunities	**Threats**
External	Fast development of lightweight user-friendly responsive apps, Good change to join an active community of developers	Huge background in Javascript/HTML needed, Low number of contributors that could compromise the future of the project

4.4.2 Sencha Touch Results

The OpenBRR spreadsheet containing the metrics final scores and comments justifying those scores can be found here:

`https://github.com/eparrillae/eparrillae-mswl-thesis/blob/master/MasterThesis/thesis/OpenBRR_Templates/BRR_Template_SenchaTouch.ods`

The final score for Sencha Touch is 3.723, which is a lower score than the one we got for jQuery Mobile, that means that this framework covers many of the needs required for the KuDo development project but it is not as complete as jQuery Mobile.

In the **Functionality** category we can see that most of the requirements are covered, the framework seems to be strong in areas like easy to install and use, compliance with HTML, CSS3 standards for the web, themes customization and availability of 3rd party plugins/extensions.

Also the development environment which is a key part of the implementation process is fully covered, in the case of Sencha there is a dedicated plugin for Eclipse IDE[30] which makes easier the development as this is the IDE I am used to use. Also Sencha Touch provides Sencha Architect[31] which is a tool that provides sketching and drag&drop graphical desing features.

One important pitfall for Sencha Mobile is that it is focused on running well in Webkit[11] based web browsers, for that reason multiplatform support in Black Berry and Nokia devices is not guaranteed.

Also the access to native features of the mobile device that has to be done through other tools like [12] which is teh same case we had in jQuery Mobile.

[11] https://www.webkit.org/

[12] http://phonegap.com/

On the other hand, the support of some advances features like media audio/video and look&feel stencils is fully supported in Sencha Mobile.

Sencha Mobile supports by default a MVC architecture[38] that makes easier the development of well-structured Mobile Apps.

In the **Usability** category we get similar scores that the ones for jQuery Mobile, we can see that the framework is easy to install and use.

For measuring the **Quality** category metrics we have checked that in Ohloh we do not have specific information about Sencha Touch[9], this is because the code repository of Sencha Touch is public, this is a big difference with jQuery Mobile where there is a GitHub project where we can check the source code improvements, bugs/enhancements raised and their status etc... This point is one of the reasons that have made Sencha Touch to be evaluated lower that jQuery Mobile in this category, without the repository it has been impossible to score the required metrics, for that reason those metrics have the lowest score possible.

In terms of major releases Sencha Touch is in line with jQuery Mobile both are in the range of 1 to 3 major releases which means that the project is currently active, the latest release is from the 15th of April 2013 (v2.2.1) so we have a very recent version to be downloadable and installed. Here we have to say that Sencha Touch has delivered more stable releases than jQuery Mobile, therefore it seems that the number of companies/developers behind the project could be bigger than the jQuery Mobile community.

In the **Security** category again we cannot check the number of critical bugs open in the Sencha Touch code and its status, therefore those metrics are scored with the lowest possible value. Like in jQuery Mobile, Sencha Touch does not have a dedicated package for security.

In the **Performance** category there are differences with jQuery because Sencha does provide tools for benchmarking and testing our Mobile Apps and also provides formal performance comparisons[32]. This is a strong point in favour of Sencha because jQuery Mobile does not provide anything on this regard.

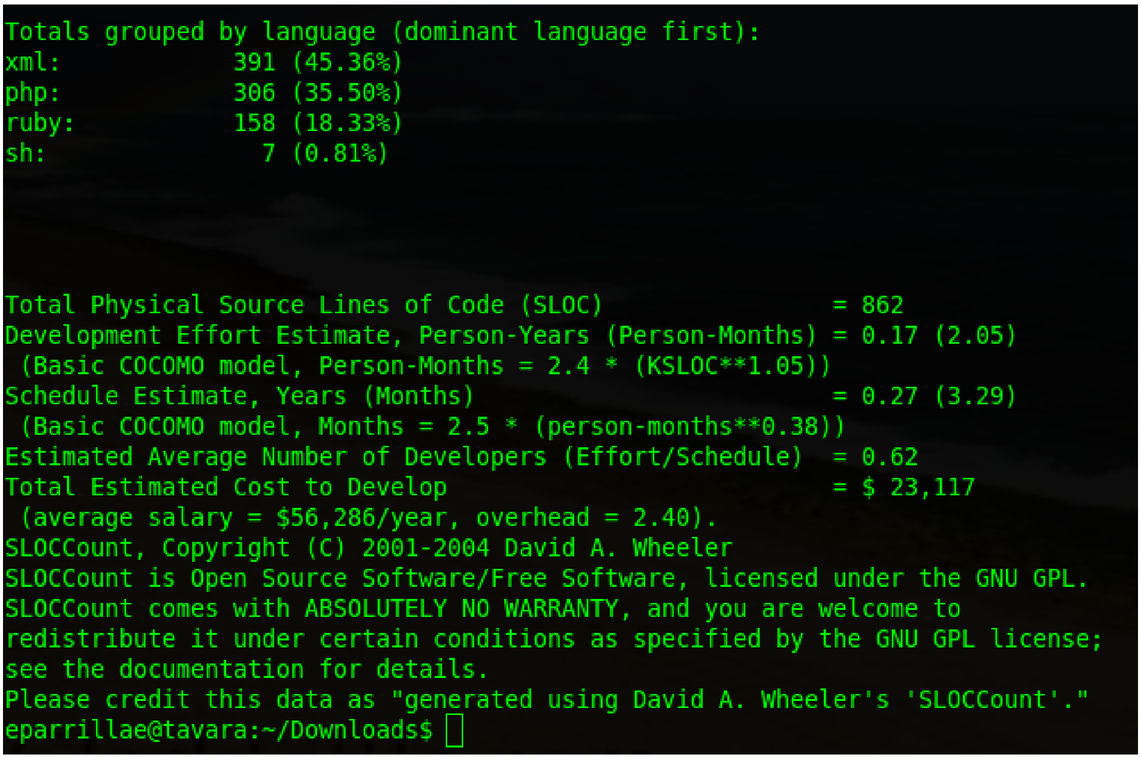
In the **Scalability** category Sencha Touch is based on a pluggable architecture which is quite extensible and also the usage of standard design patterns is provided by default[38] which is a difference with jQuery, so here the scores for the different metrics are the highest possible.

In the **Architecture** category we see that there are wide range of 3rd party plugins and extensions[36] available which is quite importante for the KuDo project, some of those resources are related with the presentation layer of the applications and others with the support for plugins to connect backend services such as databases, web services etc... here we have another difference with jQuery that does not provide such wide range of solutions.

In terms of programming languages figure below shows the results of executing sloccount[13] over the latest version of Sencha Touch downloaded

[13] http://www.dwheeler.com/sloccount/

from its web site[14] :



```
Totals grouped by language (dominant language first):
xml:              391 (45.36%)
php:              306 (35.50%)
ruby:             158 (18.33%)
sh:                 7 (0.81%)




Total Physical Source Lines of Code (SLOC)                = 862
Development Effort Estimate, Person-Years (Person-Months) = 0.17 (2.05)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 0.27 (3.29)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 0.62
Total Estimated Cost to Develop                           = $ 23,117
 (average salary = $56,286/year, overhead = 2.40).
SLOCCount, Copyright (C) 2001-2004 David A. Wheeler
SLOCCount is Open Source Software/Free Software, licensed under the GNU GPL.
SLOCCount comes with ABSOLUTELY NO WARRANTY, and you are welcome to
redistribute it under certain conditions as specified by the GNU GPL license;
see the documentation for details.
Please credit this data as "generated using David A. Wheeler's 'SLOCCount'."
eparrillae@tavara:~/Downloads$ 
```

Figure 4.3: *Sencha Touch Programming Languages*

We see that most of the code is done using PHP and XML but also there is some code in Ruby and shell scripts, which means that for implementing Mobile Apps using this framework requires knowledge in those languages, this is good for me as I have used them extensively in the past.

In the **Support** category we can clearly see that the mailing lists work very well and also it is possible to have professional support[33] which is a missing capability in jQuery Mobile, so for this category Sencha is better than jQuery.

[14] http://www.sencha.com/products/touch/download/sencha-touch-2.2.1/2283

Regarding **Documentation** category I have to say that the project is really well-documented, there are lots of books about jQuery Mobile and the web site contains all necessary resources for both developers without experience and advanced ones, in this category all scores are high. This applies also for **Adoption** category, there are many real-world examples of Mobile Apps developed using jQuery Mobile framework.

In the **Community** category we see that average volume of messages in the mailing lists in the last six months is high, unfortunately the number of commits in the last month and the number of contributors in this period cannot be checked, as we said before, there is no public access to Sencha Touch code repository.

In terms of Longevity, Sencha Touch is considered still a young project but taking into account that it is supported by its parent project Sencha[15] managed by a private company[16] which has several private partners I think the future of the project is ensured.

In the **Professionalism** category we can see that there is a dedicated conference for Sencha users[41] that promotes the framework among its users (next Sencha Conference to be held in June 2013[17]).

A big difference is that there is no way to contribute to Sencha Touch in the same way it is done in jQuery, i.e., contacting directly with its community and join them, I consider this another important pitfall, the only information

[15] http://sencha.com/

[16] http://www.sencha.com/company/

[17] http://senchacon.com/rockstar-promotion/

I have found related to developers is in the support pages, see[?].

Finally jQuery Mobile is licensed using a dual licensing schema[18] with a commercial license and an Open Source GPLv3 license [19]. For the KuDo project I was thinking about using a weak-copyleft license like MIT, BSD, APL, EPL etc... this is because my idea is to integrate several tools within my own development and in some cases GPL does not allow me to do so.

Table below summarize using a SWOT analysis the advantages and disadvantages of Sencha Touch after the OpenBRR analysis:

[18] http://www.oss-watch.ac.uk/resources/duallicence2

[19] http://gplv3.fsf.org/

	Strengths	**Weaknesses**
Internal	Easy of install and use, Wide range of 3rd party plug-ins/extensions, Strong company and private partners behind, Pretty good documentation, books, articles etc..., Performance Testing Reports and Benchmarks available, Support for some advanced features video/audio/stencils/backend services, Design patterns by default, Professional support available,	License schema, Access to native features through PhoneGap, No Open Source community behind it is not possible to become a volunteer contributor, Software quality cannot be measured (access to code repo is not available), No visibility of improvements done and future roadmap
	Opportunities	**Threats**
External	Professional support, Better integration with other external tools	License schema, No community, no open sopurce development

Chapter 5

Conclusions

In this trade-off study I have used OpenBRR as QA methodology, I have to say that I believe this methodology covers most of the main aspects of a given project in terms of functionality, usability, scalability etc... also the fact of having templates that automate the calculation of the overall score for each solution is really useful, so I think for me it is a very good way to start evaluating a given solution for my KuDo project.

While evaluating the metrics I saw that there were some analysis aspects that were out of the scope of OpenBRR templates and it was necessary to introduce new metrics to complete the study, specially to provide a ”community oriented” point of view.

In the Overview section of this study I have explained the goals of the KuDo project, 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...

During this trade-off I have discovered that jQuery Mobile has the simpler code using a relatively small object model that allows you re-use already existing html code, which makes it faster and easier to learn. Sencha has a large object model, which provides more features out of the box (DOM manipulation, CRUD Operations, access to SOAP/REST web services etc...), but takes longer to learn.

There is no established architecture, jQuery Mobile is Layout based and this allows you the freedom to do whatever you want. It is true that it extensively uses Javascript and HTML but for me is a challenge to improve my knowledge in those fields. Also it easier to integrate with other frameworks, targets more devices than Sencha Touch (Sencha Touch is in a bit of a bind having tied themselves to WebKit) and it is not tied to a particular vendor.

Moreover it is licensed under a MIT weak copyleft license which is perfect for my KuDo project. Clearly Sencha Touch bussiness model is quite different from jQuery Mobile in the case of Sencha there is a clear objective of monetization of their products (pretty closed right now a commercial entity controls it) for that reason they take care of professional support, implementation of comercial tools for development with Secncha etc... but for me this is not relevant as I am an experienced developer and do not need that kind of support, what I am expecting from the selected solution is to have a strong Open Source community behind in which I can aalso get involved at some point.

As a final conclusion I would say that iif you intend to design business-friendly web apps then probably Sencha is the right option but if your goal is to create a market-friendly apps (like the ones within KuDo project) in a quick time, then you may choose to go with jQuery Mobile.

Chapter 6

Future Work

Chapter 7

Apendix A: jQuery Mobile ”Hello World” Prototype

Chapter 8

Apendix B: Sencha Touch ”Hello World” Prototype

Bibliography

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

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

[2] **The OpenBRR Corporate Community**

http://blogs.the451group.com/opensource/2006/04/26/the-openbrr-corporate-community/

[3] **jQuery Mobile**

http://jquerymobile.com/

[4] **Sencha Touch**

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

[5] **Ohloh**

http://www.ohloh.net/

[6] **Ohloh jQuery Mobile**

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

[7] **Ohloh jQuery**

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

[8] **Ohloh Sencha**

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

[9] **Ohloh Sencha Touch**

https://www.ohloh.net/p?ref=homepage&q=sencha+touch

[10] **FLOSSMetrics**

http://flossmetrics.org/

[11] **FLOSSmole**

http://flossmole.org/

[12] **FLOSShub**

http://flosshub.org/

[13] **FLOSSpapers**

http://flosshub.org/biblio

[14] **iPhone, Android, iPad, Tablet, Mobile LinkedIn group**

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

[15] **Mobile Software Development LinkedIn group**

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

[16] **Developers HTML5, Android, iOS, Blackberry LinkedIn group**

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

[17] **Wikipedia Web Based Application Framework comparison**

http://en.wikipedia.org/wiki/Multiple_phone_web_based_application_framework

[18] **Wikipedia Comparison of JavaScript frameworks**

http://en.wikipedia.org/wiki/Comparison_of_JavaScript_frameworks

[19] **jQuery Mobile Github repo**

https://github.com/jquery/jquery-mobile

[20] **jQuery Mobile resources**

http://jquerymobile.com/resources/

[21] **jQuery Foundation**

https://jquery.org/

[22] **jQuery Roadmap**

https://github.com/jquery/jquery-mobile/wiki/Roadmap

[23] **jQuery Issues**

https://github.com/jquery/jquery-mobile/issues

[24] **jQuery Milestones**

https://github.com/jquery/jquery-mobile/issues/milestones

[25] **jQuery API**

http://api.jquerymobile.com/

[26] **jQuery Themeroller**

http://jquerymobile.com/themeroller/index.php

[27] **jQuery Demos**

http://jquerymobile.com/demos

[28] **jQuery Codiqa**

http://www.codiqa.com/

[29] **jQuery CDN**

http://jquerymobile.com/download/

[30] **sencha Eclipse Plugin**

`http://www.sencha.com/blog/sencha-eclipse-plugin-2-0-new-and-noteworthy`

[31] **sencha Architect**

`http://www.sencha.com/products/architect`

[32] **sencha performance**

`http://www.sencha.com/blog/category/html5-developer-scorecard`

[33] **sencha support**

`http://www.sencha.com/support/`

[34] **sencha Learn**

`http://www.sencha.com/learn/touch/`

[35] **sencha Demos**

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

[36] **sencha Try**

`http://try.sencha.com/`

[37] **sencha API**

`http://docs.sencha.com/touch/2.2.1/#!/api`

[38] **sencha MVC**

`http://docs.sencha.com/touch/2.2.1/#!/guide/apps_intro`

[39] **sencha Download**

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

[40] **sencha License**

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

[41] **sencha Conferences**

http://senchacon.com/

[42] **sencha Apps**

http://www.sencha.com/apps/

[43] **sencha Devs**

http://www.senchadevs.com/

[44] **sencha vs jquery 1**

http://www.classicinformatics.com/JQuery-Mobile-vs-Sencha-Touch/

[45] **sencha vs jquery 1**

http://www.classicinformatics.com/JQuery-Mobile-vs-Sencha-Touch/

[46] **sencha vs jquery 2**

http://www.quora.com/Which-is-better-jQuery-Mobile-or-Sencha-Touch

[47] **sencha vs jquery 3**

http://altabel.wordpress.com/2012/12/14/mobile-web-app-frameworks-review-sencha-touch-vs-jque