

Quality Evaluation Criteria Based on Open Source Mobile HTML5 UI Framework for Development of Cross-Platform

Hyo-jung Sohn¹, Min-gyu Lee², Baek-min Seong³ and Jong-Bae Kim^{4*}

^{1,2,3,4*}Graduate School of Software, Soongsil University, Seoul 156-743, Korea

E-mail, ¹hyojung.sohn@gmail.com,

²marse101@naver.com, ³feeling127@naver.com, ^{4*}kjb123@ssu.ac.kr

Abstract

As diver mobile platforms appear, attention is paid to the development of cross-platform that can be applied to various platforms with one development rather than the development of native application that depends on each platform. Of many frameworks that enable the development of cross-platform, HTML5-based framework is being actively developed as open source project. The present study proposes the evaluation criteria by which the most suitable framework can be selected from many open source projects. In this study, 9 evaluation criteria[14] are proposed, derived from 'Mobile Software Quality Evaluation Model', which is based on existing ISO/IEC 25000[3], 'Evaluation Criteria for quantitatively OSS Selection' and 'Mobile Web App Framework Evaluation Standard'. And we were applied to 4 UI frameworks selected from Github Repository to verify the effectiveness of the criteria. This study expects that it can reduce effort required in selecting the most suitable framework and thus increase productivity through proper maintenance supported by the right selection.

Keywords: Cross-Platform development, Open Source Project, Mobile Application, HTML5, UI Framework, Quality Evaluation, Evaluation Criteria

1. Introduction

As smartphone has widely been used, the number of mobile service developing companies takes 93% of the total developing companies as of 2014. Therefore, the development of smartphone service [7] has become an essential part of homework to do. However, a variety of mobile platforms have come out so that it increases initial cost to develop application by platform. Moreover, the companies have to pay more for repair and maintenance which is getting shorter in interval as 35% of mobile service should be updated on week base [1]. Until now, various development tools for cross-platform have been produced and distributed by many solution companies. Of late, as HTML5, a next generation web standard, has grown and each platform applies HTML5 to mobile browser, the development of cross-platform using HTML is drawing attention in the industry. The development of cross-platform is based on existing web technologies such as HTML, CSS and JavaScript and one development can benefit many different platforms with same functions. As a result, it can reduce time and cost of development and realize a good idea quickly. Furthermore, it makes it possible that web and application can be developed at the same time by development tools such as Cordova or Phonegap and is much earlier to learn than one in native language. More than anything else, anyone who has experienced existing web development can develop application with ease. As a result, HTML5-based cross-platform development is thickening the segment of developers using it.

^{4*} Corresponding author. Tel. : +82-10-9027-3148.
Emailaddress: kjb123@ssu.ac.kr(Jong-Bae Kim).

A number of projects that support cross-platform, among many open source frameworks, are under development and a considerable number of open source projects have already developed and earned popularity along with commercial frameworks. Although there are many open source mobile HTML5 UI frameworks, each of them has its own characteristics and functions to seek. Therefore, it takes a great deal of time and effort to select a suitable framework. To ease this situation, this study is aimed to propose evaluation criteria that can be used in selecting cross-platform.

Based on the main and supplementary characteristics suggested in 'Mobile Software Quality Evaluation Model'[2] and 'Evaluation Criteria for quantitatively OSS Selection', among existing researches, the present study mapped the evaluation standard that reflects the characteristics of Mobile Web-App Framework and proposed selection evaluation criteria that reflects all the mobile software, open source and web-app framework.

2. Related Works

2.1 HTML5 Mobile UI Framework

Although it is supposed that the same web standard such as HTML5, CSS3 or JavaScript are used for mobile web page and hybrid application, it is imperative that hybrid application be additionally materialized in a precise and delicate manner so that it can provide the same experience as native application. To do so, elaborate JavaScript should be made for touch work and screen-changing animation and especially such functions as using local storage and application cache of HTML5 should be realized to work at a mobile terminal regardless of network condition. However, it can be troublesome for a developer to take care of such details of intended mobile optimization in terms of development time and stability [6].

In the meantime, mobile UI framework is already equipped with such optimized functions for a mobile application, so a developer needs only to utilize ready-made functions to compose screen up to the level of native application. It indicates that the use of mobile UI framework can be a productive and efficient choice for the development of hybrid application.

2.2. Mobile Web App Framework Evaluation Standard

Cross-platform development is based on web development technology. Considering this characteristic, this study aims to take advantage of 'Mobile Web-App Framework Evaluation Standard [5]' proposed by 'Heitkotter (2013)'.

The present study proposed each standard from developer's and user's perspective. For the standard from developer's viewpoint, it brought forth 7 standards: does initial cost occur to introduce framework?(License and Costs); is it framework that can be managed and used continuously?(Long-term feasibility); is it well documented and supported for a developer?(Documentation and Support); is it framework whose concept is already familiar to a developer?(Learning Success); does configuration such as development environment help a developer minimize the effort of development?(Development Effort); is it possible to extend framework?(Extensibility); and are source codes well modularized?(Maintainability). On the other hand, this study proposed 4 standards from user's perspective: is UI composition optimized for a mobile application? (User Interface Elements); does framework give the same experience that native application does? (Native Look and Feel); does it provide the same loading time as native application? (Load Time); and is its response time short and of high performance?(Runtime Performance). They are summarized in Table 1.

Table 1. Evaluating Standard of the Developer's and User's Perspectives for Mobile Web-app Framework

Developer's Perspective	User's Perspective
D1. License and Costs	U1. User Interface Elements
D2. Long-term feasibility	U2. Native Look and feel
D3. Documentation and Support	U3. Load Time
D4. Learning Success	U4. Runtime Performance
D5. Development Effort	
D6. Extensibility	
D7. Maintainability	

2.3 Mobile Software Quality Evaluation Model

'Suh' proposed a quality evaluation model of mobile software [2] based on ISO/IEC 25000 in consideration of 7 characteristics of mobile application (mobility, small device, reduction of access time to device, expansion of user segment, all-over access, security and privacy, environmental heterogeneity). With it, he suggested 6 characteristics and 13 sub-characteristics. 6 characteristics include performance efficiency, compatibility, usability, reliability, security and portability.

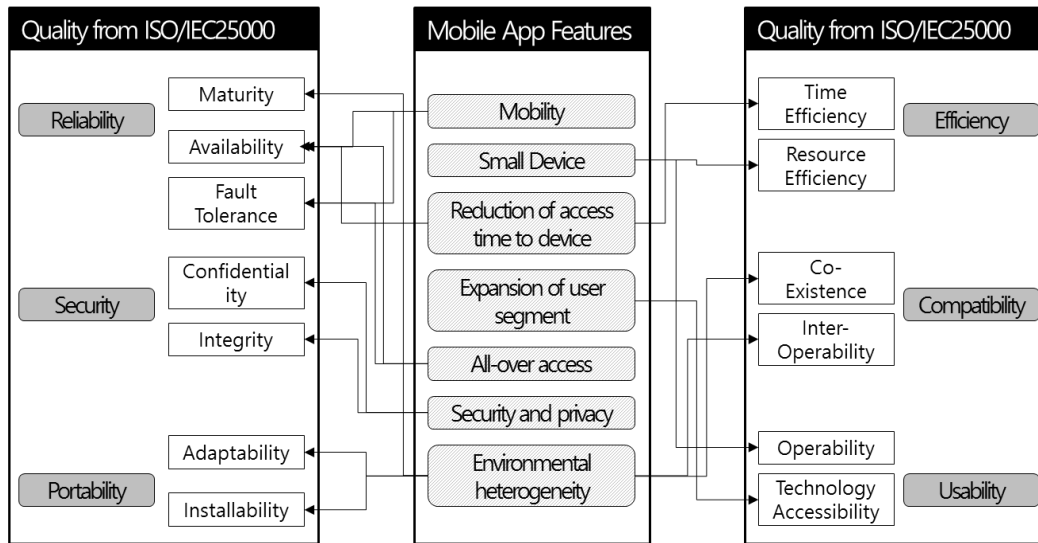


Figure 1. Quality Evaluation Model for Mobile Application

2.4. Evaluation Criteria for Selecting Open Source Software

In his study, H. J. Lee [4] suggested initial-stage evaluation criteria necessary to screen candidate groups in selecting open source software(OSS) [8] and detail-stage evaluation criteria that enables quantitative evaluation of the candidate groups by the characteristic of project.

Initial-stage evaluation criteria include 4 (community support, legitimacy, documentation, compatibility) and detail-stage evaluation criteria include 10 (applicability, analyzability, changeability, easy installation, testability, operability, functionality, non-functional compliance, performance and security).

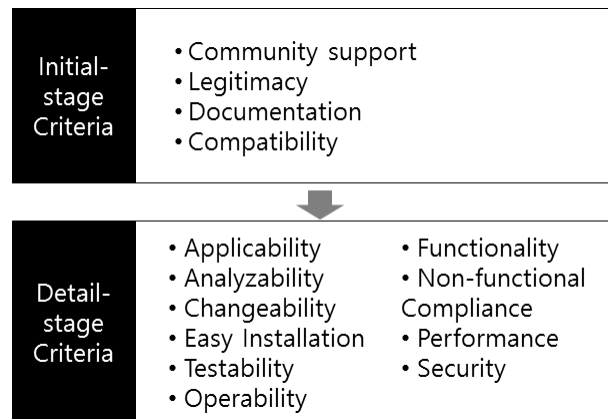


Figure 2. Evaluation Criteria for Quantitatively OSS Selection

3. Derivation of Evaluation Criteria for Selecting Open Source Mobile HTML5 UI Framework

In this chapter, evaluation criteria for selecting open source mobile HTML5 UI framework is derived by mapping Heitkotter (2013)' 'Mobile Web Application Framework Evaluation Standards' on the main and sub-characteristics of Suh's 'Mobile Software Quality Evaluation Model' and H. J. Lee's 'Evaluation Criteria for quantitatively OSS Selection'.

Since it turned out that the characteristics of 'Mobile Software Quality Evaluation Model' were mostly mapped with the evaluation standards from user's perspective on 'Mobile Web App Framework Evaluation Standard', 4 evaluation standards (UI Elements, Native Look and Feel, Load Time and Runtime Performance) were derived. And initial-stage evaluation criteria of 'Evaluation Criteria for quantitatively OSS Selection' were mapped with the evaluation standards from developer's perspective and thus 4 standards (License and Costs, Long-Term Feasibility, Documentation and Support, Learning Success) were drawn. Last, this study chose an addition index (Multi-Platform Support) in consideration of cross-platform framework and the number of mobile platform. Eventually, a total of 9 evaluation criteria [14] were attained. For security, which are commonly proposed from 'Mobile Software Quality Evaluation Model' and 'Evaluation Criteria for quantitatively OSS Selection', this study excluded it because we focus on UI framework. These standards chosen for selecting open source mobile HTML5 UI framework are summarized in Figure 3.

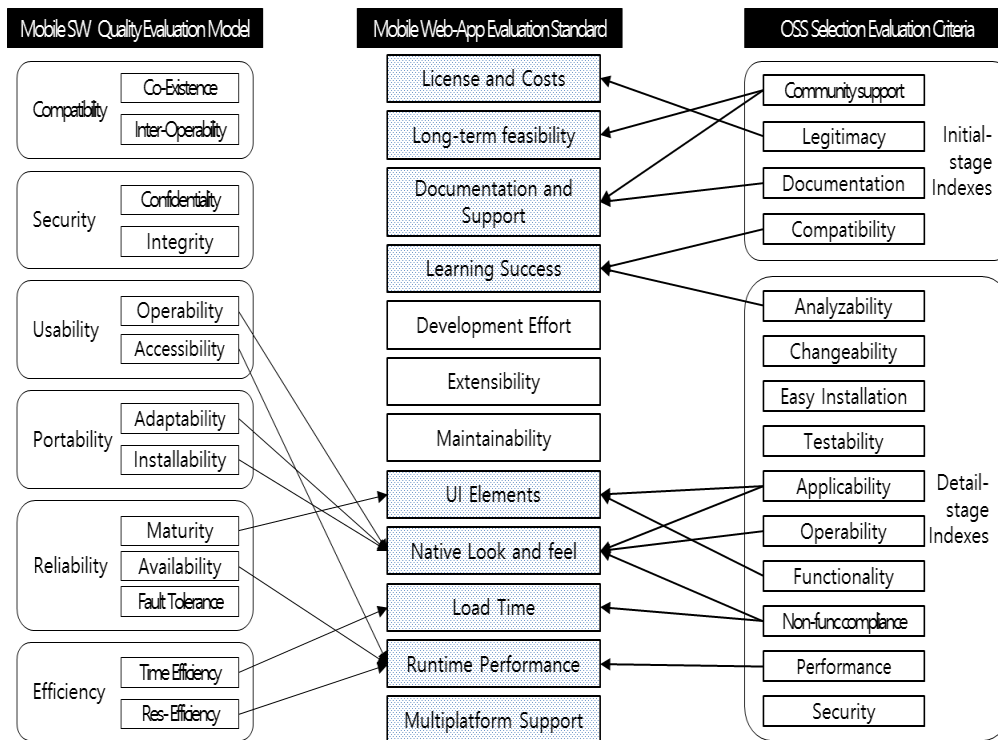


Figure 3. A Map of Open Source Mobile HTML5 UI Framework Derived Evaluation Criteria

3.1. UI Unity

It is a standard to evaluate whether realized UI gives uniformed feel regardless of mobile platform and whether it is optimally expressed to each terminal or not. Although the form of UI element, which is expressed by platform browser, varies slightly, UI should be realized in a way to give unified feel so that its effect can be maximized in cross-platform development and the framework should provide screen composition without cracking or curtaining regardless of fragmented size of terminal screen.

3.2. User Experience

It is to evaluate smooth screen changing and multi-touch felt when a user uses native application. If an application developed on cross-platform framework gives a user less convenient and natural experience than more native application, he will turn away from the application. Therefore, user experience [9] is a very important part of evaluation standards.

3.3. Load Time

It is to evaluate the time and taken to the first run and response speed of event in comparison with those of native application. Cross-platform framework is based on web technology. Therefore, cross-platform framework should provide the same run speed and response speed, at least, as those of native application in event of network disconnection so that it can satisfy a user.

3.4. Performance

It is to evaluate if it functions properly whenever and wherever a user wants. To achieve satisfactory performance, technical support is necessary to tackle error in application instantly and help a user use a terminal efficiently.

3.5. License

It is to evaluate the types of licenses adapted to framework. Open source framework that adopts open license will be preferred to that using closed license that involves strict regulations.

3.6. Community Activation

Support for bug fixing and updating relies on the activated level of community. Therefore, it is to evaluate the degree of community activation. Backed by the activated community, a developer can expect that he can continue to a concerned framework even when there is change in development environment by updating mobile platform. Therefore, this standard should be considered.

3.7. Documentation

It is to evaluate the level of documentation and the presence of support team. They can save a developer's effort, which lead to cost saving. Therefore, when framework provides the same functions, one with higher level of documentation should be chosen.

3.8. Learning Curve

It is to evaluate whether framework complies with existing framework development standards or whether it is well modularized. Framework in a good compliance with the standards is easily understood and better used by a developer.

3.9. Multi-Platform

It is to evaluate the types of mobile platform that a concerned framework supports. When it supports more platforms and more recent version of platform, it can reduce a developer's effort and cost. In addition, it can increase productivity.

4. Evaluation

The evaluation criteria proposed in this study were applied to some HTML5 UI Frameworks selected from the actual open source repository and evaluated for effectiveness. To select the subject framework to the evaluation, key word searching ('mobile' and 'HTML5') was conducted in Github.com and the results were arranged in 'Most Stars' according to preference. Of the candidates, only UI frameworks were chosen and finally 4 candidate projects were determined. Figure 4 shows the screen of the searching process.

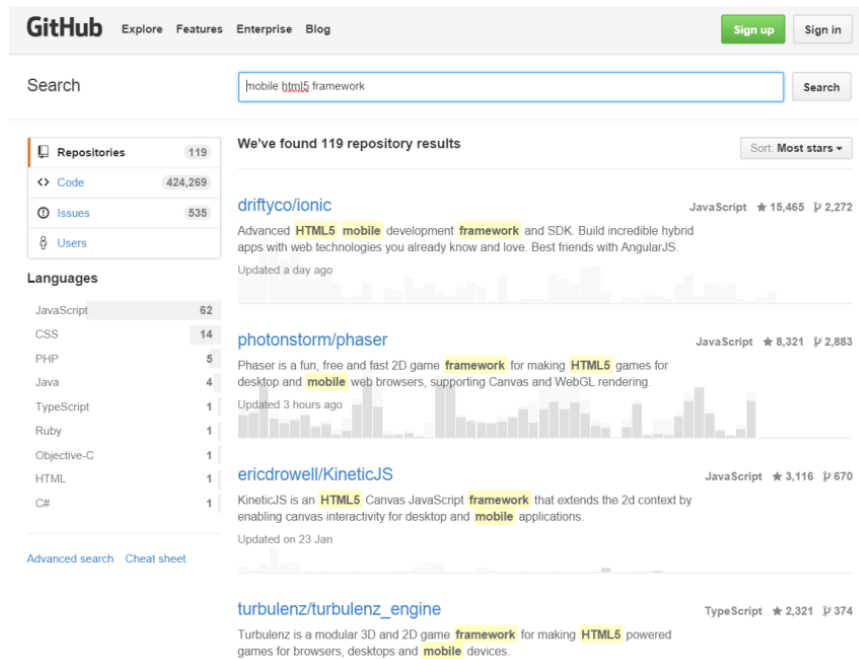


Figure 4. A Result of searching HTML5 UI Framework on Github.com

Each of 9 evaluation criteria was scored for 4 different frameworks on a 5-point scale from ‘1’ for ‘very poor’ to ‘5’ for ‘very good’. For data to evaluate, each project page and the official homepages in Github.com and external developer communities, if necessary, were referred to.

Table 2. Assessment Summary

Criterion	Ionic	Intel's App Framework	OnsenUI	The-M-Project
UI Unity	5	5	5	5
User Experience	5	2	5	1
Load Time	3	5	4	3
Performance	5	5	3	5
License	5	5	5	5
CommunityActivation	5	3	4	2
Documentation	5	3	2	4
Learning Curve	4	5	5	3
Multi-Platform	2	4	3	5

4.1. Ionic Framework

Ionic [10] explains itself that it is a framework ultimately designed to make a hybrid application a same UI as native application in quick and easy way. Therefore, the realized *UI elements* have excellent unity (5) regardless of platform as well as a similar level of *User Experience* to that of native application (5).

Since Ionic was developed to have 380kb capacity in development, it is regarded as taking some *LoadTime* (3). As for *Performance*, it supports iOS 6+ and Android 4.1 so it can be said that it has somewhat limited performance. But it supports more various OS versions than other candidates (5).

In addition, it adopts MIT *License* so it is easy for many users to have access and use It (5). Furthermore, the activities in github.com for a month show (March 12, 2015-April 12, 2015) which performed 2 releases and 4 merges. And, 24 authors have pushed 36

commits to master and 63 commits to all branches. On master, 57 files have changed and there have been 3,064 additions and 992 deletions. Therefore, it shows the most active *Community Activation* among the candidates. It actively provides forum and support activities that is active not only in github.com but also external communities such as StackOverflow.com so that it is expected to easily get help for repair and maintenance and new version updating(5). It also provides a variety of *Documentation* formats, which is highly evaluated. On its current official homepage, video, tutorial, formulas and guide document are presented and several demo applications can be confirmed (5). Looking from the perspective of *Learning Curve*, the concerned framework consists of JavaScript (75.5%), HTML (13.4%) and CSS (10.8%) so web developers can have an easy access to the framework. It yet provides 'Create Builder' beta version, so following development is expected to be much easier (4).

As of *Multi-Platform*, it supports only iOS and Android. But it declared that it would support more diverse mobile platforms. It had low score on this evaluation item (2).

4.2. Intel's App Framework

Intel's App framework [11] is an open source project that Intel Co. manages. It realizes the same UI for all its platforms. Thanks to it, it has a high level of *UI Unity* for various platforms (5). But it has lower *User Experience* than native application. Its user experience was evaluated is more like that of HTML5 (2).

Intel's App framework provides the function of setting desired style and colors with Style Builder and CSS Generator, respectively, and downloading them to apply only concerned library. It takes the maximum capacity of about 200kb and *Load Time* is expected to be very fast (5). It supports IE 10+, Blackberry 10+ and Firefox OS as well as iOS 6+ and Android 4+, so it has the least limitation among the candidates to development and operation, allowing us to expect its high *Performance* (5).

Since Intel's App framework also adopts MIT *License*, it is easy for many people to access and use it (5). The review of its *Community Activities* in Github.com for a month shows (March 12, 2015-April 12, 2015) that it performed no release and merges six times. And, 5 authors have pushed 28 commits to master and 29 commits to all branches. On master, 4 files have changed and there have been 195 additions and 111 deletions. Therefore, this framework was similarly evaluated to other candidates (3). Although its *Documentation* can be seen relatively poor because it provides guide documentations and API references (3), it supports development with the tools such as CSS generator, App Strater and Style Builder, beside documentation. Furthermore, this framework is composed of JavaScript (35.4%), HTML (9.0%) and CSS (55.6%), so that a web developer can be familiar with them during development. Particularly, it helps a user confirm the result of a test through App Stater, which is helpful in developing(5).

Intel's App framework supports a number of *Multi-Platforms* including iOS, Android, Microsoft Windows 8, and new RIM Blackberry and so on. So it received a high score on multi-platform performance evaluation item(4).

4.3. OnsenUI

OnsenUI [12] is based on AngularJS like Ionic and supports a number of *UI Unity* regardless of platforms (5). It shows a similar level of *User Experience* to that of a native application (5).

OnsenUI helps download the source codes by composing templates and colors by theme on its official page. Although different by theme, the source code of chosen theme takes about 270kb of the total capacity of the framework, so page *Load Time* is relatively fast (4). In addition, it supports iOS 7+ and Android 4.0.2+ version so that it has relatively high limitation compared with other candidates. As a result, it received a low score on *Performance* (3).

Because OnsenUI adopts Apache *License* (version 2.0), it seems that it is less burdensome for an individual and a company developer to choose and use it (5). Looking at its monthly *Community Activity* (March 12, 2015-April 12, 2015), on Github.com repository, it let us know that it has one release and 35 merges. And, 9 authors have pushed 167 commits to master and 171 commits to all branches. On master, 401 files have changed and there have been 46,264 additions and 28,400 deletions. Compared with other candidate frameworks, its activities are greater and highly evaluated (4). As for *Documentation*, it provides guide documentation and API documentation, but a user has to directly download framework, which causes inconvenience. As a result, it receives a low score on the evaluation item (2). Looking from the perspective of *Learning Curve*, the concerned framework consists of JavaScript (56.7%), HTML (5.4%), CSS (37.8%) and others (0.1%), so that a web developer can easily design it. Besides, since it provides Monaca IDE, which is HTML5 IDE, it is easy to work with it and easier to designate a theme and a style and download them (5).

Besides OnsenUI supports not only iOS, Android but also desktop browser. Therefore, it was evaluated to have less number of *Multi-Platforms* to support (3).

4.3. The-M-Project

The-M-Project [13] provides a variety of the same *UI Elements Unity* on diverse mobile platforms (5). But it is a framework designed on the basis of jQuery Mobile, so it is more dependent upon *User Experience* provided in jQuery Mobile rather than platform, which results in less nature than a native application (1).

In developing a framework using The-M-Project, it requires about the large capacity of about 400kb, so it can be a relatively heavy page, which requires a long *Load Time* (3). As indicated in the *Performance* criteria, it supports iOS 6+ and Android 4.0+, so it is considered somewhat limited but it received a high score because it can support OS version for the candidates(5).

It adopts MIT *License*, which facilitates access by many people (5). Unlike the other candidates, the-M-Project does not use the forum, community, release functions which are provided in Github.com. Therefore, it is not easy to simply compare with the other candidates but it does not actively upload posts in the forum run by Google Group, running only 350 themes. As a result, it was evaluated that the-M-Project lacks in *Community Activity* (2). As for its *Documentation*, it has well-organized guide documents as well as tutorial and API documents, also providing sample applications. However, of the contents, it is not deep (4). Its *Learning Curve* consists of JavaScript (35.3%) and CSS (64.7%). Therefore, it does not require additional effort for a web developer but it adopts jQuery Mobile in development method and it has a very broad range of API to apply, which takes some time and effort to get used to it (3).

Of the candidates, the-M-Project supports the most number of mobile platforms including iOS, Android, Window Mobile, Fire Fox and Blackberry platform, which gave it a high score on the section of *Multi-Platform* (5).

To apply the proposed evaluation criteria to an actual project, weight should vary by the size and functions of an application to develop. In addition, the evaluation results can be different by the level of skillfulness and the size of development team. However, the present study has its significance that it proposes the standard for selecting the most suitable project of open source projects that support UI in developing. This evaluation selected the most popular mobile HTML5 frameworks out of the results of keyword searching and a relative evaluation was made on them by each evaluation criterion, focusing on the embodiment of UI and results.

The results show that all of 4 frameworks have a high level of UI Unity in order to realize entire native applications but they have different user experiences: Intel's App framework and the-M-Project received low scores due to its realization method of HTML5 and jQuery Mobile, respectively. Judging the capacity of framework from

loading time, it turned out that Intel's App framework is the lightest framework while Ionic is the heaviest framework. As for the section of functional performance (run at anytime and anywhere), OnsenUI received the lowest score due to its sole support of OS version. All of 4 frameworks also received the highest score on license openness.

Ionic was evaluated to have most active for Community Activation while The-M-Project received the lowest score on this section of evaluation because it does not have a community in due Github.com. As for Documentation, Ionic was given the highest score thanks to its providing various kinds of documents while OnsenUI had the lowest score on the section of evaluation due to its late start of business. All of them were evaluated to give convenience to a web developer with easy access, but both Intel's App framework and OnsenUI that support with IDE (developmental tool) received relatively higher scores. Last, all the four frameworks turned out that they basically support iOS and Android platform as multi-platform. Particularly, The-M-Project received a high score on this evaluation item because it supports additional platforms including Window Mobile, Blackberry and Fire fox. This chapter can be summarized in Table 2 above.

5. Conclusions

The present study derived a total of 9 evaluation criteria by which a developer can chooses a framework that reflects characteristics of mobile platform most and utilizes the merits of open source software best among a number of open source frameworks when a developer develops a mobile application using open source mobile HTML5 UI framework.

Those evaluation criteria were derived from the mapping of existing studies: 'Mobile Software Quality Evaluation Model', 'Evaluation Criteria for quantitatively OSS Selection' and 'Mobile Web App Framework Evaluation Standard' and the derived criteria consists of 9 valuation criteria that include UI Unity, User Experience, Load Time, Performance, License, Community Activation, Documentation, Learning Curve and Multi-Platform.

This study has its significance in that it proposed evaluation criteria to evaluate frameworks by focusing on the realization of UI. To apply the criteria to an actual case, however, the propose evaluation criteria system needs additional work to reflect the factors in designing stage such as size and purpose of project and skillfulness of a developer and to establish objective evaluation criteria rather than relative evaluation. In a following study, an attempt will be made to practically apply the evaluation model proposed in this study to some of frameworks of open source project sites and to verify the effectiveness.

These evaluation criteria are expected to save time and cost taken in selecting framework and as a result a well-chosen framework enhances smooth maintenance and repair, which leads eventually to increased productivity.

References

- [1] "Mobile Trend in The First Quarter", (2014), <http://www.mobizen.pe.kr>.
- [2] J. H. Suh, J. H. Choi, J. B. Kim and J. W. Park, "Design of Quality Evaluation Model for Mobile Application", Journal of the Korea Institute of Information and Communication Engineering, vol. 18, no. 10, (2014) October.
- [3] "ISO/IEC 25000. Software and System Engineering: Software Product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE", FDIS. ISO, (2005).
- [4] H. J. Lee, H. J. Kim, D. Y. Choi and I. W. "A Study on Evaluation Criteria for quantitatively OSS Selection", Journal of the Korea Academia-Industrial cooperation Society, vol. 13, no. 4, (2014) April.
- [5] H. Heitkotter, T. A. Majchrzak, B. Ruland and T. Weber, "Evaluating Frameworks for Creating Mobile Web Apps. In 9th International Conference on Web Information Systems and Technologies, (2013) May 7-9, Aachen, Germany.
- [6] W. S. Chae, W. S. Park, C. W. Choi, W. Ahn, S. Y., Roh, B. S. Lee and J. W., "HTML5 and Mobile Framework Trends for Mobile Web Applications", ETRI, (2012).

- [7] S. Al-Fedaghi, "Developing Web Applications", International Journal of Software Engineering and Its Applications (IJSEIA), vol. 5, no. 2, (2011) April.
- [8] S. Munish and K. Kuljit, "A Review of Open Source Software Development Life Cycle Models", International Journal of Software Engineering and Its Applications (IJSEIA), vol. 8, no. 4, (2014) March.
- [9] M. L. Devita, H. Dadan nd N. H. Achmad, "Analysis of User Experience Quality on Responsive Web Design from its Informative Perspective", International Journal of Software Engineering and Its Applications(IJSEIA), vol. 8, no. 5, (2014) May.
- [10] "Ionic", <http://www.ionicframework.com/>, <https://github.com/driftyco/ionic>.
- [11] "Intel's App Framework", <http://app-framework-software.intel.com/>.
- [12] "OnsenUI", <http://onsen.io/>.
- [13] "The-M-Project", <http://www.the-m-project.org>.
- [14] H. J. Sohn, M. G. Lee, B. M. Seong and J. B. Kim, "A Study on Quality Evaluation Indexes to Select Open Source Mobile HTML5 UI Framework", Proceeding International Workshop Art, Culture, Game, Graphics, Broadcasting and Digital Contents 2014, (2015) April 15-18, Jeju, Korea.

Authors



Hyo-Jung Sohn, she received her bachelor's degree of Business Administration in Soongsil University, Seoul (2006). And she is studying her master's degree of software engineering in Graduated Soongsil University, Seoul. Her current research interests include pen source development and management information system.



Min-Gyu Lee, he received her bachelor's degree of Information and Telecommunication in Dongguk University (2014). And he is studying his master's degree of software engineering in Graduated Soongsil University, Seoul. His current research interests include Open source software and Security.



Baek-Min Seong, he received his bachelor's degree of Computer Science in Soongsil University, Seoul (2014). And he is studying his master's degree of software engineering in Graduated Soongsil University, Seoul. His current research interests include database.



Jong-Bae Kim, he received his bachelor's degree of Business Administration in University of Seoul, Seoul (1995) and master's degree (2002), doctor's degree of Computer Science in Soongsil University, Seoul (2006). Now he is a professor in the Graduate School of Software, Soongsil University, Seoul, Korea. His research interests focus on Software Engineering, and Open Source Software.

