



UNIVERSITY OF GOTHENBURG

# Evaluation of Cross-Platform Tools for Mobile Development

*Bachelor of Science Thesis in Software Engineering and Management*

ALIREZA PAZIRANDEH  
EVELINA VOROBYEVA

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
Göteborg, Sweden, June 2013

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

## **Evaluation of Cross-Platform Tools for Mobile Development**

Alireza Pazirandeh  
Evelina Vorobyeva

© Alireza Pazirandeh, June 2013.

© Evelina Vorobyeva, June 2013.

Examiner: Morgan Ericsson

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
SE-412 96 Göteborg  
Sweden  
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering  
Göteborg, Sweden June 2013

# Evaluation of cross-platform tools for mobile development

Alireza Pazirandeh  
Department of Computer Science  
and Engineering  
Gothenburg University  
E-mail: pazirandeh@student.gu.se

Evelina Vorobyeva  
Department of Computer Science  
and Engineering  
Gothenburg University  
E-mail: evelina@student.gu.se

**Abstract**—Cross-platform mobile development is a new area of software engineering that allows the companies to reduce development time and cost. The paper surveys major cross-platform mobile development tools available on the market, develops a framework for guided choice of such tools, and applies this framework to an actual decision problem from within an authentic industrial case. The paper’s main result is a framework for choosing cross-platform development tools, for a specific engineering context. The framework consists of four evaluation stages and a taxonomy with seven evaluation parameters: tool’s type, deployment mode, license, compatibility, mobile platforms, documentation, popularity. The case considered, is a large IT solution provider serving a range of manufacturers, design units, and sales organizations within the Automotive sector. Our conclusions are that, for this case, the most suitable tools were PhoneGap, RhoMobile, Appcelerator Titanium, Sencha Touch or Xamarin, with RhoMobile as the recommended choice. Availability of the framework improves the quality of decision making in IT solution providers, and thus the qualities and costs of their solutions.

**Keywords:** cross-platform tools, mobile development, grading framework, Android, iOS

## I. INTRODUCTION

Mobile technology has become an important part of people’s life. Today, people use different mobile applications everyday: apps for health, transportation, shopping and many other purposes. Smartphone is as popular as regular mobile phones were five years ago. The number of smartphone and tablet users is increasing, and so is the need for efficient mobile application technologies [1]. This tendency has forced companies to develop mobile applications in order to follow current market standards and satisfy customers.

However, there is a number of obstacles that companies need to address. One of these obstacles is development for an increasing number of mobile platforms, for example: Android, iOS, BlackBerry, Windows Phone, Windows Mobile and Symbian OS [2]. Building a different app for each platform is very expensive. [3]. This has been the main reason why cross-platform tools were created for mobile development. A few years ago these tools were not trusted enough to be used in the industry, but nowadays have improved and matured. Companies are more often using these tools to decrease the programming time and publish the app for all platforms simultaneously.

Over the years many different tools were created to address the difficulty of programming for different mobile platforms

and this has created a new dilemma: which tool is right for a given project? It is hard to understand from the tool’s description what kind of project it is most suitable for: games, media applications, etc. In addition, to standardize the products, create higher quality, have better security and more consistent applications most companies have their own guidelines and principles to be followed in their development process which some of these tools might not comply with.

**Research question.** This study addresses the following research question: ”How can a company choose the right cross-platform tools with the respect to a particular technical context?”. By technical context, local design guidelines, local development tools in use, and application domain are meant.

To answer this question, a single-case study was conducted at Volvo IT, a global company supplying ”reliable IT solutions to customers across multiple industries [4]”. It is important for Volvo IT to find out which cross-platform tools are currently available on the market and to choose a suitable tool with regards to Volvo Group Architectural Principles. This is why, the objective of this study was to develop a method, useful for practitioners in Volvo IT and other companies with development guidelines to follow.

Survey study was used as research strategy. The research process was divided into four steps: overview, selection, refinement and assessment. The overview step included finding suitable cross-platform tools for this research and collecting generic data about each of them. The selection step included ranking the tools with the designed ranking criteria and selecting five tools suitable according to the company’s needs . In the refinement step, specific data was collected for the five highest ranked tools in the selection step. The assessment step evaluated the five selected tools with the Volvo IT guidelines and principles. Finally, one of the tools was recommended to the company as the most suitable with the regards to their needs and requirements.

The rest of the paper presents the study’s theoretical framework (in section II), research method (in section III), and research results (in section IV). Discussion of study results, recommendations to the company, limitations and validity of the research are presented (in section V). Finally the conclusions are presented (in section VI).

## II. THEORETICAL FRAMEWORK

This section gives the terminology used in this study.

### A. Programming language

A programming language is an artificial language designed to communicate instructions between humans and machines (computers). There are many different types of programming languages and all of them are used in different projects for different kinds of tasks. The examples of programming languages used in mobile applications development are: Java, Objective-C, C++, HTML5, CSS3, and JavaScript [5].

### B. Integrated Development Environment (IDE)

Integrated Development Environment is an application which provides different tools and facilities for programmers. Usually, IDE consists of source code editor, debugger, and automation tools. In mobile development, the examples of such IDEs are XCode, Android Studio, Eclipse, and NetBeans [6].

### C. Software License

Software license is as a set of rules for usage and distribution of a software. Generally, software licenses are either proprietary or free and open source. The main difference between them is the permission to further distribute and copy the software, where a free license enables the end-user to copy and redistribute it further, while a proprietary license restricts the user from doing so [7].

### D. Mobile operating system (Mobile OS)

A mobile OS is an operating system (platform) which operates digital mobile devices e.g. smartphones and tablets. The most common mobile operating systems are the following:

- **Android.** Google Android platform is a Linux-based Mobile OS. It is a software package or software stack for mobile devices, containing an operating system, middleware, and main applications. Native Android applications are developed in Java, C++, and C programming languages with the help of Android Software Development Kit (SDK), which provides all the necessary tools and Application Programming Interfaces (APIs). [8]
- **iOS.** iOS is the operating system originally created by Apple Inc. for the following products: iPhone, iPad and iPod Touch. The iOS license restricts installation of the system to only original Apple Inc. products, so it is impossible to legally install iOS on other devices. Objective-C and C++ programming languages and XCode IDE are used for developing native mobile applications for the iOS platform.
- **Windows Phone.** Windows Phone is a mobile OS created and released by Microsoft. The native mobile applications for this platform are mainly developed with C# and VB.Net programming languages, with the use of Windows Phone SDK, which contains all the necessary tools needed to develop a native app for Windows Phone OS.
- **Windows Mobile.** Windows Mobile is a predecessor of Windows Phone and is incompatible with Windows Phone software or devices. The native applications are developed in Visual Studio IDE with VB.NET and C#.

- **BlackBerry OS.** Blackberry OS is an operating system which was created by Blackberry Inc. for its' Blackberry smartphone devices. The native mobile applications are created in Java using QNX Momentics IDE.
- **Symbian OS.** Symbian OS is a mobile operating system originally designed for Nokia smartphones and currently supported by Accenture. The applications are developed in C++ with the help of Qt framework.

### E. Cross-platform mobile development tool

A cross-platform mobile development tool is a type of software (e.g. library, framework, programming language, set of tools, etc) which enables the user to develop an application once and compile it to several mobile operating systems [9]. Using such tools have several benefits:

- *Development time reduction.* The fact that the code is written once, but compiles to several mobile operating systems reduces development time.
- *Ease of development.* Each mobile platform requires you to use a specific programming language for development. Cross-platform tools, most often, requires one programming language in the development for all the platforms. The tools usually require easy-to-learn web development programming languages: HTML5, CSS3, JavaScript.
- *Ease of maintenance.* It is cheaper and easier to maintain one code base as opposed to maintaining different codes for different mobile platforms [10].

Even though cross-platform tools have many advantages, there are some known drawbacks of using them as well. For example:

- *Performance.* Poor performance in comparison to native applications.
- *Access to native API.* Not all the tools have access to all the mobile devices' functionality, e.g. access to camera, accelerometer, storage.
- *Reliability.* Your project will be dependent on the tool and there is no guarantee that the tool will continue to exist.
- *Up to date.* Mobile platforms are changing at a fast pace and it takes cross-platform tools time to support the new features in the new mobile platform [10].

### F. Architectural principles and guidelines

Different companies are creating the architectural principles and guidelines which the developers and software architects have to follow. Usually, it is done in order to improve the security and make the project easier to maintain [11].

## III. METHOD

Survey study was chosen to be the research strategy for this thesis. Survey study [12] is defined as follows: "...surveys share with their physical counterparts three crucial characteristics:

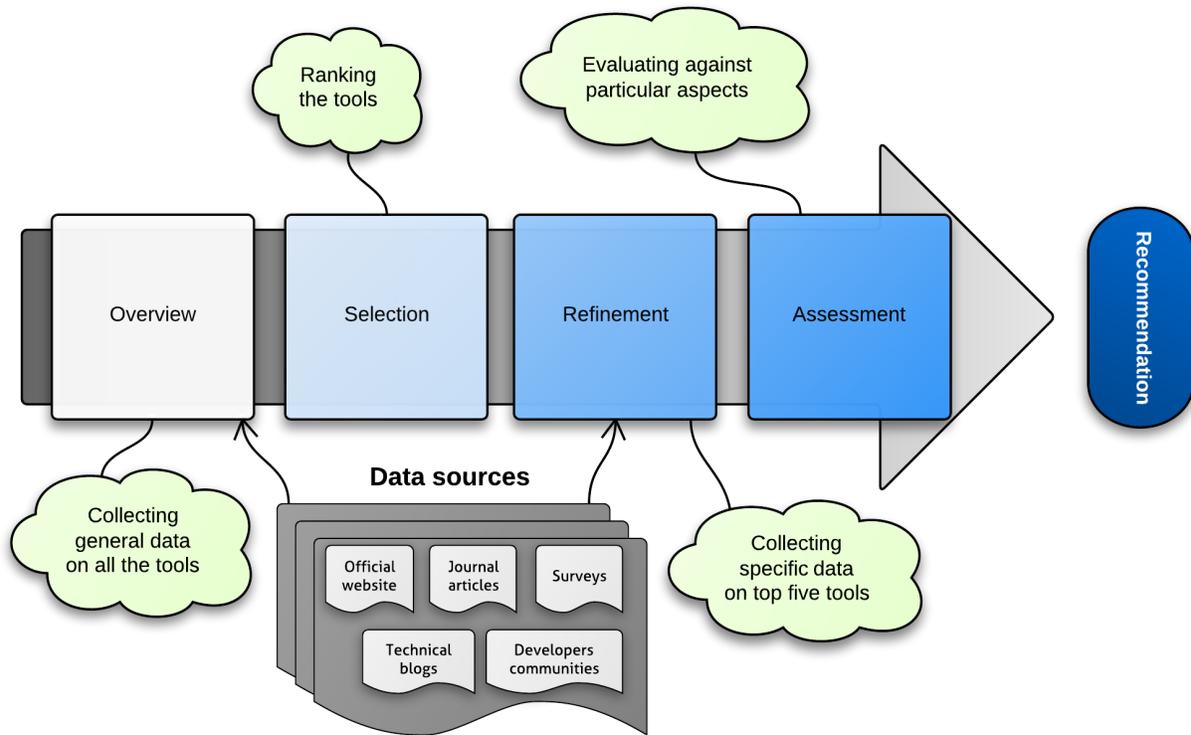


Fig. 1: Research process

- *Wide and inclusive coverage.* A survey, in principle, should take a panoramic view and 'take it all in'.
- *At a specific point in time.* Surveys provide a snapshot of how things are at a specific point in time.
- *Empirical research.* It involves the idea of getting out of the chair, going out of the office and purposefully seeking the necessary information out there [12]."

This survey study was mainly a "survey of documents", which means that survey strategy was applied to documents and not to people [12]. In this case, documents consisted of documentation and API from the official websites of the cross-platform tools, previous research papers, articles, and forums' entries.

The research process consisted of four steps (see Figure 1.):

- 1) **Overview.** Creating a list of the most popular tools for this research.
- 2) **Selection.** Choosing five tools that suited the company's needs.
- 3) **Refinement.** Expanding knowledge on the five selected tools.
- 4) **Assessment.** Evaluating the five selected tools against the company's specific guidelines and principles.

The following sections provide a detailed explanation for each of the research steps.

#### A. Overview

During this first step, general information about cross-platform tools was collected and analyzed. First, "Google"

web search engine was used with "Cross-platform mobile development tool" as keyword. Tools were chosen and put in a spreadsheet from the first five pages of the search results. These tools were cross checked with the tools that were mentioned in research articles from previous years [9], [13]. In addition to this, information about existing tools was gathered from different developer communities (e.g. "Stack Overflow").

Initially, there were 24 tools in the list, but some of the tools were removed at this stage due to being outdated (the official website did not exist or had not been updated for years).

All the data gathered for these tools were stored on a spreadsheets. A representation of these data can be found in the Results section. The gathered data were categorized in a way described below (see Figure 3):

**General information.** The name of a company or organization that made the tool, information about the documentation quality and the tool's type were collected. According to Hartmann et al. [9] cross-platform tool can be of one of the following types:

- *Platform.* "A set of frameworks, tools and services that not only allow the user to build a complete mobile application but also to configure, package and distribute it to app stores or the cloud. Platforms normally include some sort of IDE to ease app construction, comprehensive documentation, support and automation tools."
- *Product.* "Offers a specific functionality or service ready to be used and integrated into a mobile solution."

A product is built using a combination of libraries, frameworks and/or platforms but these are not normally visible to the end user.

- *Framework.* "A set of libraries, software components and architecture guidelines that provides the developer with a comprehensive toolkit to build a complete mobile application, from top to bottom. Commonly called full-stack development frameworks; normally more complex to use than a single library."
- *Library.* Library is a "small, self-contained toolkit that offers very specific functionality to the user. Normally used in conjunction with other libraries and tools to make up the full mobile app. Examples include UI widgets and 3D graphics libraries [9]."
- *Cross-compiling programming language.* Cross-compiling programming language is an easy-to-use language with which you can create an application and compile it to several platforms.

**Deployment mode.** According to Korf and Oksman [14] there are mainly 3 deployment formats of applications that can be created with cross-platform mobile development tools (see Figure 2). They are:

- *Native.* Native applications are "specific to a given mobile platform (iOS or Android) using the development tools and language that the respective platform supports (e.g., XCode and Objective-C with iOS, Eclipse and Java with Android) [14]."
- *Hybrid.* Hybrid applications "make it possible to embed HTML5 apps inside a thin native container, combining the best (and worst) elements of native and HTML5 apps [14]."
- *Web (HTML5).* Web mobile application "use standard web technologies typically HTML5, JavaScript and CSS. This write-once-run-anywhere approach to mobile development creates cross-platform mobile applications that work on multiple devices [14]."

Data stating the tool's ability to deploy in each of these modes was gathered in this sheet.

**Programming language.** The programming languages that are used for coding with each tool were included in this sheet.

**Coverage.** Coverage sheet contained different mobile OS support. The data showed if the tool could create apps not only for iOS and Android but for other mobile operating systems as well.

**OS compatibility.** In this sheet the ability to use the tool on different desktop OS (Mac OS, Windows, and Linux) was recorded. Some tools gave the possibility to code on a web IDE and also provided a cloud system for mobile deployment. These data were also added to this sheet as extra columns.

**Popularity.** This data was collected from a survey and Stack Overflow developers community [13], in order to understand where on the market the tool currently is. User satisfaction rate was found in the survey done by VisionMobile, an ecosystems analyst firm [16]. Number of Stack Overflow tags and search result on a tool were also collected in this sheet.

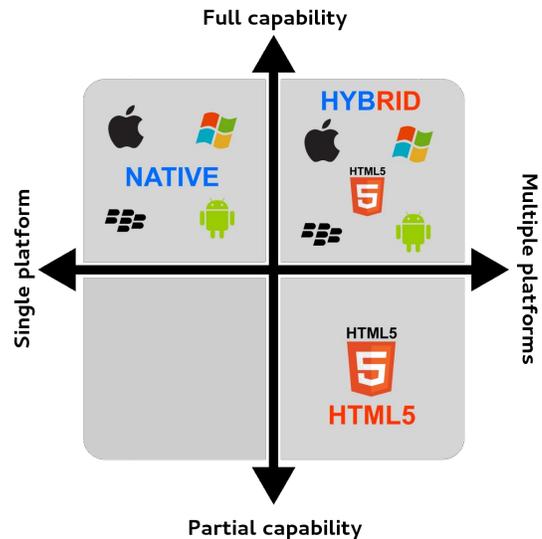


Fig. 2: [15] Deployment formats

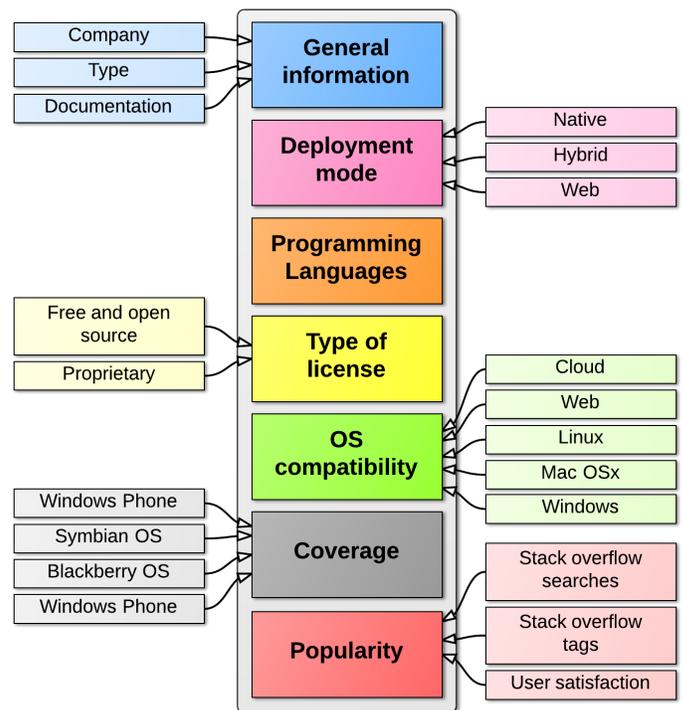


Fig. 3: Taxonomy tree

**Type of licence.** Information about the licence (Proprietary or Free and Open Source) was stored on this sheet.

### B. Selection

On this step a grading system was introduced. All tools were given a score of 0 to 5 on predefined important parameters (see Figure 4). Then each parameter was given a weight. The total points of each tool were calculated as sum of each score multiplied by its weight:

$$\sum (parameter_{score} \times parameter_{weight})$$

These parameters and their weights were chosen with the help of Volvo IT as their preference was essential for the result of this paper.

The parameters, how their points were calculated and their weights were as follows:

- **Popularity.** Weight: 4  
Number of stack overflow tags and search results were brought to a range of 0 to 5 where 5 was equal to the maximum number between the tools. The survey result was in the range of 0 to 5. The final popularity score was calculated with the following formula:

$$\frac{(tags_{norm} \times 2) + (searches_{norm} \times 1) + (satisfaction \times 3)}{6}$$

- **Coverage.** Weight: 1  
Windows Phone and BlackBerry each had 2 scores, and Windows Mobile and Symbian each had 1 with the total score of 5. Android and iOS are excluded from the list, as they are supported by all the graded tools.
- **Documentation quality.** Weight: 2  
Zero was given to non-existing documentation and five to excellent documentation.
- **OS compatibility.** Weight: 2  
Each of Windows, Linux, Mac OS, Web IDE, and cloud had 1 score.

Five tools with the highest points were chosen to continue with in the following steps.

**C. Refinement**

On this step, more information about the five chosen tools was collected subjectively. This was done to provide deeper knowledge and detailed specification about benefits and shortcomings of each of them for the company.

**D. Assessment**

Volvo Group IT Governance has established ten architectural principles based on the most important quality attributes for them. When designing and developing a new application, the programmer should always take into considerations each of the principles. In this study, the principles were used to evaluate the most suitable cross-platform mobile development tools for Volvo IT. The description of each principle used is as follows:

- 1) **Conformity to standards.**  
Drive usage of open and industry standards at Volvo.
- 2) **Autonomous and loose coupling, between components and applications.**  
Flexible subsystem and granular component setup, avoiding monoliths.
- 3) **Simplicity in solutions and work methods.**  
Clean solutions from technical, application and user perspective.
- 4) **Strive for usage of existing Volvo services.**  
Whenever possible avoid application specific infrastructure and instead use already existing services at Volvo.

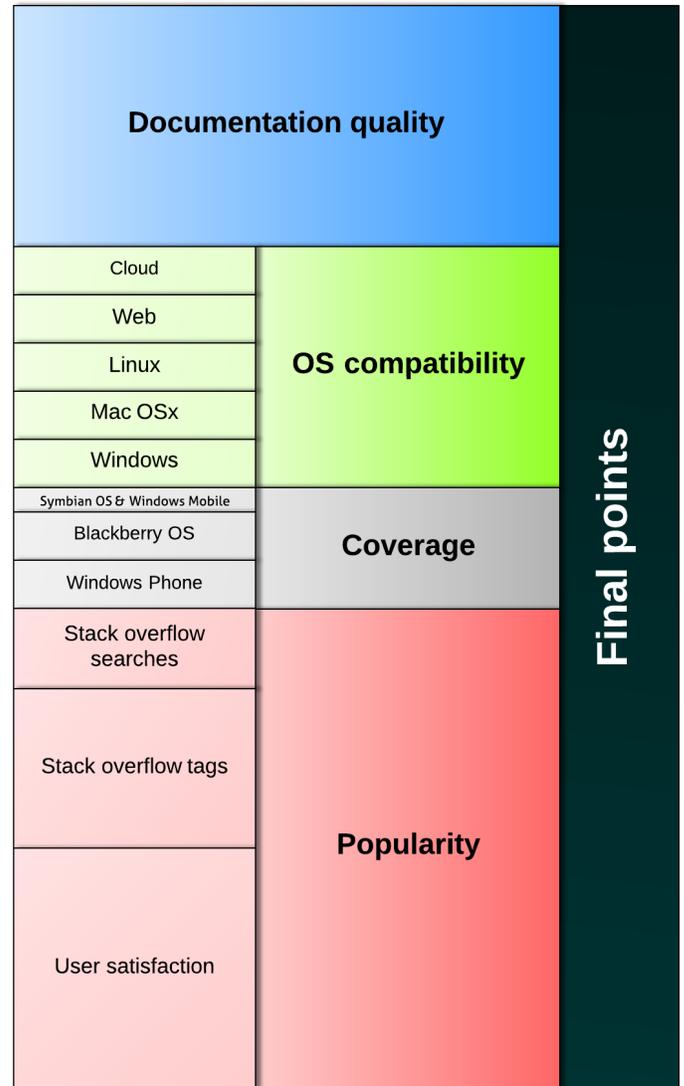


Fig. 4: Selection parameters

- 5) **Robust solutions.**  
Strive for robust solutions securing uptime.
- 6) **Performance focus from the start.**  
Strive for good performance in solutions from the start.
- 7) **Secure solutions.**  
Strive for secure solutions from the start.
- 8) **Good integration solutions.**  
Follow Volvo Group integration policies and guidelines.
- 9) **Usage of Agile work methods and design principles.**  
Use Agile system development and implementation principles.
- 10) **Maintainable solutions.**  
Deliver maintainable solutions to Maintenance.

## IV. RESULTS

This section presents the results of the research and is divided into several subsections with the results of each of the steps in Method section, overview, selection, refinement, and assessment.

### A. Overview and Selection

The results of these steps are presented in form of tables, including all the collected data and ranking results for each of the tools. All the tables can be found on page 7. The tables present tools', general information including the tool type and documentation quality ranking (see Table I), deployment mode (see Table II), popularity (see Table III), OS compatibility (see Table IV), and mobile OS coverage (see Table V). Missing rows in the tables indicate that relevant data was not found on the tool's official website. Collected data was analyzed and the final ranking table was created, where the top-five most suitable cross-platform tools can be found (see Table VI).

### B. Refinement

According to this study's result, the top five cross-platform tools currently on the market are: PhoneGap, Titanium, Rho-Mobile, Sencha Touch, and Xamarin. This subsection gives a detailed description for each of the mentioned tools, in order to widen the company's knowledge about each of them.

1) *PhoneGap*: is an open-source cross-platform mobile development tool released by Adobe Inc. under Apache 2.0 license in 2008. PhoneGap is one of the most popular tools on the market with large developers community. Applications are developed using HTML5, CSS, and JavaScript. It follows the hybrid model by providing native app wrappers for mobile platforms with an embedded browser which renders the UI and makes the interaction between the mobile device and web application possible. The PhoneGap JavaScript API provides the access to the native functionality (e.g. camera and contacts). The best suited projects for this tool are existing web applications, needed to be converted into a mobile application.

a) *PhoneGap Strengths*: PhoneGaps primary strength is that it is very simple. Also any native platform that supports a web view can become a PhoneGap platform, meaning that native extensions can be developed fast. Another strength is the existence of a JavaScript API which provides native functions. HTML, CSS, and JavaScript code can be wrapped in a native app and published. Another strong side of this tool is that the native wrapper source code is provided so it can be changed according to the project's requirements. Finally, PhoneGap has a strong developers community.

b) *PhoneGap Weaknesses*: The user interface quality is the biggest issue of PhoneGap due to the quality of the platform's rendering engine. Web views on different platforms have some limitations, and this is why the quality of PhoneGap UI is significantly lower than applications with a native UI. In addition to this PhoneGap cannot be extended with native user interface. The last thing to mention is that due to the iOS restrictions, the iOS compass can not be accessed with PhoneGap, which makes development of some projects impossible.

**Name:** PhoneGap.

**Cross-platform tool type:** Framework.

**Deployment mode:** Hybrid mobile applications.

**Mobile Operating Systems:** Android, iOS, Windows Phone, BlackBerry, Symbian.

**Programming Languages:** HTML5, JavaScript, CSS.

**License** Free and open-source (Apache 2.0)

2) *RhoMobile*: applications are developed mostly in Ruby language using a Model View Controller (MVC) architecture, separating the logic (Ruby) from the UI design (HTML). The apps generated are compiled into Ruby 1.9 bytecode, which is then interpreted by the Ruby Virtual Machine for the target platform. The access to the mobile device's native API is provided by Rhodes API. JQTouch is supported for creating touchscreen UIs. RhoMobile is targeting enterprise application development and provides several tools for that. Some of the tools are as follows:

- **Rhodes** is an open source Ruby-based framework for cross-platform mobile development.
- **RhoSync**, working together with RhoConnect, updates the device with any changes in the database.
- **RhoConnect** provides the connection to the enterprise databases.
- **RhoHub** is a cloud based development environment, with Git support, providing the possibility to compile an application for all the different platforms on the cloud, enabling development of iOS applications on a non-Mac OS.
- **RhoGallery** is a part of RhoHub, providing an enterprise "app store" for the company.
- **RhoStudio** is an IDE, providing templates, simulators and debugging tools which makes developing applications easier.

a) *RhoMobile Strengths*: MVC architecture support is the main benefit of RhoMobile.

b) *RhoMobile Weaknesses*: Need to know Ruby well to do anything, which is not as popular as other programming languages like JavaScript, Java or PHP. The tool does not generate source code, and only generates native packages [17].

**Name:** RhoMobile.

**Cross-platform tool type:** Platform.

**Deployment mode:** Hybrid and native mobile applications.

**Mobile Operating Systems:** Android, iOS, BlackBerry, Windows Phone, Windows Mobile

**Programming Languages:** Ruby for controllers, HTML, JavaScript, CSS.

**License** Rhodes is free and open-source (MIT License), but other components need to be purchased. The applications created with the tool can be licensed with any license, however if the created app is using some features (e.g. Barcode extension), the Motorola Solutions Software License has to be applied to it [18].

Tools	Company	Type	Documentation
Adobe AIR	Adobe Systems	Platform	3.70
Titanium	Appcelerator	Platform	4.50
Corona SDK	Anscas Mobile	Product	3.90
FeedHenry	FeedHenry	Platform	3.10
Haxe	Haxe Foundation	Programming Language	3.80
Icenium	Telerik	Product	2.90
KonyOne	Kony	Framework	1.00
Marmalade Juice	Ideaworks3D	Platform	3.60
Monaca	Asial	Platform	3.90
Moscrif	Moscrif	Product	4.10
MoSync	MoSync	Platform	3.30
Mulberry	-	Framework	0.00
NeoMAD	Neomades	Platform	1.30
PhoneGap	Adobe Systems	Framework	4.70
RhoMobile	Motorola Solutions	Platform	4.80
Sencha Touch	Sencha	Framework	4.60
Umbrella SDK	Zimsoft	Platform	2.10
Unity	Unity Technologies	Game engine	4.50
VSNomad	Red Gate Software	Library	2.00
Xamarin	Xamarin	Platform	3.60

TABLE I: General information

Tools	Windows	Mac OSx	Linux	Cloud	Web
Adobe AIR	✓	✓	✗	✗	✗
Titanium	✓	✓	✓	✗	✗
Corona SDK	✓	✓	✗	✗	✗
FeedHenry	✓	✓	✓	✓	✓
Haxe	✓	✓	✓	✗	✗
Icenium	✓	✗	✗	✓	✓
KonyOne					
Marmalade Juice	✓	✓	✗	✗	✗
Monaca	✗	✗	✗	✓	✓
Moscrif	✓	✓	✗	✗	✗
MoSync	✓	✓	✗	✗	✗
Mulberry					
NeoMAD	✓	✓	✗	✗	✗
PhoneGap	✓	✓	✓	✓	✗
RhoMobile	✓	✓	✓	✓	✗
Sencha Touch					
Umbrella SDK	✗	✗	✗	✓	✓
Unity	✓	✓	✗	✗	✗
VSNomad	✓	✗	✗	✗	✗
Xamarin	✗	✓	✗	✗	✗

TABLE IV: OS compatibility

Tools	Native apps	Hybrid apps	Web apps
Adobe AIR	✓	✓	✗
Titanium	✓	✗	✗
Corona SDK	✓	✗	✗
FeedHenry	✓	✓	✓
Haxe	✓	✗	✗
Icenium	✗	✓	✗
KonyOne	✓	✓	✗
Marmalade Juice	✓	✓	✗
Monaca	✓	✓	✗
Moscrif			
MoSync	✓	✗	✗
Mulberry			
NeoMAD	✓	✗	✗
PhoneGap	✗	✓	✗
RhoMobile	✓	✗	✗
Sencha Touch	✗	✓	✓
Umbrella SDK	✓	✗	✗
Unity	✓	✗	✗
VSNomad	✗	✓	✗
Xamarin	✓	✗	✗

TABLE II: Deployment mode

Tools	Windows Phone	Black-berry	Windows Mobile	Symbian
Adobe AIR	✗	✓	✗	✗
Titanium	✓	✓	✗	✗
Corona SDK	✗	✗	✗	✗
FeedHenry	✓	✓	✗	✗
Haxe	✗	✗	✓	✗
Icenium	✗	✗	✗	✗
KonyOne	✓	✓	✗	✓
Marmalade Juice	✗	✓	✗	✗
Monaca	✓	✗	✗	✗
Moscrif	✗	✗	✗	✗
MoSync	✓	✗	✗	✗
Mulberry	✗	✗	✗	✗
NeoMAD	✓	✓	✗	✓
PhoneGap	✓	✓	✗	✓
RhoMobile	✓	✓	✗	✗
Sencha Touch	✓	✓	✗	✗
Umbrella SDK	✗	✗	✗	✗
Unity	✗	✗	✗	✗
VSNomad	✗	✗	✗	✗
Xamarin	✗	✗	✗	✗

TABLE V: Mobile OS coverage

Tools	Stack Overflow tags	Tags 0-5	Stack Overflow searches	Searches 0-5	Satisfaction
Adobe AIR	0	0.00	3,069	0.47	3.70
Titanium	3,184	1.14	6,995	1.07	3.70
Corona SDK	1,369	0.49	2,526	0.38	3.80
FeedHenry	0	0.00	3	0.00	
Haxe	325	0.12	1,162	0.18	
Icenium	0	0.00	39	0.01	
KonyOne	0	0.00	27	0.00	
Marmalade Juice	121	0.04	377	0.06	3.90
Monaca	0	0.00	1	0.00	
Moscrif	0	0.00	2	0.00	
MoSync	32	0.01	218	0.03	4.00
Mulberry	0	0.00	30	0.00	
NeoMAD	0	0.00	0	0.00	
PhoneGap	13,975	5.00	32,826	5.00	3.70
RhoMobile	201	0.07	687	0.10	3.70
Sencha Touch	5,541	1.98	4,392	0.67	3.70
Umbrella SDK	0	0.00	12	0.00	
Unity	1,594	0.57	8,396	1.28	3.70
VSNomad	0	0.00	1	0.00	
Xamarin	7,179	2.57	18,228	2.78	3.90

TABLE III: Popularity

Tools	Popularity	Coverage	Documentation	SDK OS	Total
PhoneGap	4.35	4.50	4.70	4.00	<b>39.30</b>
RhoMobile	1.89	4.00	4.80	4.00	<b>29.17</b>
Titanium	2.41	4.00	4.50	3.00	<b>28.63</b>
Sencha Touch	2.62	4.00	4.60	0.00	<b>23.69</b>
Xamarin	3.27	0.00	3.60	1.50	<b>23.28</b>
Unity	2.25	0.00	4.50	2.00	<b>22.01</b>
Adobe AIR	1.93	2.00	3.70	2.00	<b>21.11</b>
Marmalade Juice	1.97	2.00	3.60	2.00	<b>21.10</b>
MoSync	2.01	2.00	3.30	2.00	<b>20.64</b>
Corona SDK	2.13	0.00	3.90	2.00	<b>20.31</b>
FeedHenry	0.00	4.00	3.10	5.00	<b>20.20</b>
Haxe	0.14	0.50	3.80	3.00	<b>14.65</b>
Monaca	0.00	2.00	3.90	2.00	<b>13.80</b>
Moscrif	0.00	0.00	4.10	2.50	<b>13.20</b>
Icenium	0.00	0.00	2.90	3.00	<b>11.81</b>
NeoMAD	0.00	4.50	1.30	2.00	<b>11.10</b>
Umbrella SDK	0.00	0.00	2.10	2.00	<b>8.20</b>
VSNomad	0.00	0.00	2.00	2.00	<b>8.00</b>
KonyOne	0.00	4.50	1.00	0.00	<b>6.51</b>
Mulberry	0.00	0.00	0.00	0.00	<b>0.01</b>

TABLE VI: Final result

3) *Titanium*: was released in 2006, but support for Android and iOS was added in 2009. Titanium applications are written using HTML, JavaScript, and CSS, with support for PHP, Ruby, and Python. Titanium APIs allows the developer to access native features (e.g. camera), native UI modules, and some optional modules. Titanium compiles native libraries and JavaScript code into bytecode. The compiler then builds the package for the target platform (e.g. iOS). The company claims that 70-80% of the developed code is reusable across other platforms [19]. The output application contains mostly native code, and rendering is executed natively. Titanium is mostly suitable for informative and interactive applications, rather than games. Appcelerator Studio is an add-on Eclipse-based IDE, which contains an editing tool with Git integration, inline debugging and cloud connection.

a) *Titanium Strengths*: The goal of Titanium is to provide a higher level API for native mobile development across platforms, and reduce the gap between the native devices and cross-platform mobile development tools. Appcelerator (the company) claims to be covering 90% of the native functionality and provides a platform giving the possibility to develop a native API that is not provided by their API [20]. Moreover it is possible to use native UI extensions in Titanium, which makes it possible to achieve more native look and feel on the application. The animations and behavior match what a native apps provides, as the native UI controller is used and UI is not rendered with Flash or OpenGL. Another advantage is the ease of use of Titanium: all the developer needs to know is mostly some JavaScript, HTML and CSS.

b) *Titanium Weaknesses*: Titanium is only available for a few mobile operating systems, due to the complications with development of Titanium API for all platform. The Mobile web browser is lacking quality, so the Titanium UI Widgets do not look and feel as native. Extending Titanium with native plug-ins is not as easy, the official documentation is lacking information, and in most cases the developer has to know the Titanium architecture in order to create a well functioning extension [20].

**Name:** Titanium.

**Cross-platform tool type:** Platform.

**Deployment mode:** Native, hybrid and web (HTML5) mobile applications.

**Mobile Operating Systems:** Android, iOS, BlackBerry, Windows Phone.

**Programming Languages:** HTML5, JavaScript, CSS.

**License:** Free and open-source (Apache 2.0), and enterprise licenses for extended features.

4) *Sencha Touch*: lets web applications have a consistent look and feel across both the iPhone and Android. It has created a blend of widgets that dont look like any specific operating system. Sencha Touch has a good community support. There is the possibility to develop different applications compatible with different sizes of the devices (phones vs tablets). In the latest release, the MVC framework support was added, which is valuable when developing enterprise applications, as the code will be more structured.

a) *Sencha Touch Strengths*: Sencha Touch has excellent expanded touch events support - great for touch or canvas

oriented applications. Also, it has a very solid documentation and a large community.

b) *Sencha Touch Weaknesses*: It relies on a heavyweight framework, which is quite inconvenient. Another drawback is complex learning: sometimes it is not worth it to go deep into JavaScript. For example, if the application consists of only one activity frame with not a complicated logic and GUI. In addition, the UI development is quite hard, as the elements are added with JavaScript and not HTML.

**Name:** Sencha Touch.

**Cross-platform tool type:** Framework, library.

**Deployment mode:** Hybrid and web (HTML5) mobile applications.

**Mobile Operating Systems:** Android, iOS, BlackBerry, Widows Phone

**Programming Languages:** JavaScript.

**License:** Free and open-source (GNU GPL v.3) and commercial licenses.

5) *Xamarin*: allows developers fluent in C# and .NET to write applications for iOS and Android platforms. While Xamarin does not provide direct Windows Phone support, C# and .NET application code can also run on Windows Phone. Xamarin is based on the free and open-source Mono .NET framework, that allows Microsoft .NET applications to run on other platforms, and it offers two mobile development products, Xamarin.iOS (formerly MonoTouch) and Xamarin.Android (formerly MonoDroid). It consists of class libraries, a virtual machine, and a C# compiler. Applications are created either in Visual Studio or Xamarin's own IDE "Xamarin Studio". Applications developed with Xamarin are made in a way, that all the logic is written in C# and .NET, and this code is shared between all the platforms. In order to ensure the native look and feel the UI layer is different for every platform and has to be developed with the native API.

a) *Xamarin Strengths*: The main advantage in using Xamarin is the possibility to create iOS applications with Visual Studio on Windows. Another advantage is the Component Store which is a tool that is integrated in Xamarin Studio and Visual studio. It provides additional components and plug-ins for applications, that can be added to the project just with couple of clicks.

b) *Xamarin Weaknesses*: Unfortunately Xamarin.iOS for Visual Studio currently has no user interface editor. Developers wanting to have access to the UI editor have to use the one available in XCode or write the whole UI layer in the code.

Even though the iOS applications can be created in Visual Studio on Windows, "Xamarin still requires Apple's iOS SDK tools to complete the code generation. For example, the XCode simulator is required for code debugging, and the application's interface must be designed using XCode's layout and storyboard editor, Interface Builder (IB). Finally, to distribute apps in the App Store you must use XCode's code-signing mechanism [21]."

In addition, Xamarin Studio is a great IDE, but supports development of both iOS and Android applications only in Mac OS environment. On Windows, Xamarin Studio gives the developer possibility to develop applications only for Android.

**Name:** Xamarin.

**Cross-platform tool type:** Product.

**Deployment mode:** Native mobile applications.

**Mobile Operating Systems:** Android, iOS, Windows Phone.

**Programming Languages:** C#, .Net Libraries.

**License** Commercial/Proprietary license. The tool uses Mono virtual machine, which is licensed under LGPL v2., so if the application developed with Xamarin is intended for commercial use, there is a possibility to purchase a commercial license and avoid any complications.

### C. Assessment

On this step, the tools named in the previous step were evaluated against VGAP. Although VGAP consist of ten principles, not all of them were relevant for this research. The company decided that suitable architectural principles for this context were the following five:

- *Conformity to standards.*  
This principle is about usage of open and industry standards at the company. As PhoneGap, Titanium, Sencha Touch and RhoMobile are open-source, but Xamarin is not, Xamarin is lower than the other tools in this principle.
- *Autonomous and loose coupling, between components and applications.*  
This principle is mostly dependent on the application's implementation. However, the tool being used can become an obstacle in implementing a software which meets this principle. After studying the top five tools, it was deducted that none of the tools go against this principle. Some of the tool by supporting MVC framework help with the implementation. Therefore, the tools were ranked as follows: RhoMobile, Sencha Touch (because they support MVC), PhoneGap, Titanium, and Xamarin.
- *Robust solutions & performance focus from the start*  
These two principles were grouped together because in this context they are closely related. For these principles, tools that can deploy native applications rather than hybrid and web were ranked higher. This was due to the fact that applications with native UI generally have higher performance. Moreover tools which are more mature on the market and have better support were considered as more robust. After taking these findings into consideration, the top five tools would be ranked as follows: Titanium, RhoMobile, Xamarin, PhoneGap, and Sencha Touch.
- *Maintainable solutions.*  
Maintainability of the app depends on the way it was implemented. However, there are some factors in which a tool can influence maintainability. For example, some tools require different UI implementation for each platform, which makes it harder to maintain the code. Taking this into consideration, the tools were ranked in the following way: RhoMobile, PhoneGap, Titanium, Sencha Touch and Xamarin.

The results of the research showed that for Volvo IT, the top five cross-platform development tools currently on the market are: PhoneGap, Appcelerator, RhoMobile, Sencha Touch, and Xamarin. However, it was discovered that some of these tools are more suitable for specific types of projects. For example, some PhoneGap reviewers mention that "Applications that are content heavy, but not really graphically heavy and do not rely heavily on using the phones features are a good fit for PhoneGap [22]." This means that if the company wants to develop a highly graphical and dynamic game, PhoneGap might not be a good choice. This is why, it is always important to take into consideration that even though, these tools are in the top five list, each of them has some benefits and drawbacks, which should be considered when starting a new project.

In addition, the results of the research were a little different from the results of the similar survey conducted earlier this year by [16], which means that cross-platform mobile development is a dynamic and fast developing sphere of software engineering, and such researches have to be conducted constantly, in order to provide up-to-date results and observe the changes and tendencies.

Moreover it is important to note that in the developed framework, "Refinement" and "Assessment" steps do not include any quantitative analysis or guidelines on how to conduct them, but mostly rely on subjective qualitative analysis based on the data collected from developer communities and user's blogs. From the point of view of positivism, the only truth is in scientific knowledge, where only the first two steps of the framework, "Overview" and "Selection", fit this idea. However, from the pragmatic point of view, practice is a true knowledge, and the objective of this research was to create a method, which will be useful for practitioners. These types of methods exhibit a trade-off between rigor and execution cost. However for this study, on a quickly evolving market with new offers every month, it was not considered worthwhile to spend a significant amount of time on "proving" what option is the best.

### A. Recommendations

Based on the research results and authors' deduction, RhoMobile Suite is recommended to Volvo IT to use as a cross-platform mobile development tool with the regards to VGAP. The reasons are that RhoMobile Suite:

- is suitable for enterprise applications
- supports MVC framework
- creates applications for all the major mobile OSs
- consists of several products, supporting the development of enterprise applications in all the different ways
- creates native applications that support full device features (camera, compass etc.) and can be easily published on all the app stores

### B. Limitations

There are some limitations and shortcomings to take into consideration, and they are listed further:

- **Sample size.** Twenty of the major cross-platform mobile development tools were evaluated in this research. Due to the fact that the new mobile development tools are being released every day, there are more than a hundred tools on the market and it is almost impossible to include all of them in this research.
- **Tools testing.** This research did not include any tools testing or prototype development. And its findings are based on different documentation types which means that the study results could be different if some important data would be collected during the tools testing process.
- **Grading system.** The grading system was created for this study, and the tools were graded subjectively. However, the authors believe that this system worked well in this research, and can be customized in every other case, according to preferences and requirements.

### C. Validity of the framework

The feedback that was given by Volvo IT stated that the results of this research were different from the results of similar research conducted two years ago. This means that this study brought fresh knowledge with several new cross-platform tools which did not exist then. They also mentioned that copies of this research will be distributed amongst mobile developers and architects within their company to provide them with the latest information on this topic.

However the creditability of the developed framework can be questioned as the findings in "Refinement" and "Assessment" steps rely on the honesty of the tool providers and the accuracy of answers in developer communities and blogs. The proposed method can be used without "Refinement" and "Assessment" steps, which will improve the creditability of the framework.

## VI. CONCLUSION

Cross-platform mobile development is a recent and unexplored area of software engineering. The possibility to develop for mobile platforms at once is a benefit appreciated by most of the mobile application developers. However, due to the variety of cross-platform tools for mobile development, choosing the right tool is not an easy task. Moreover not all companies can use cross-platform tools due to the necessity of following their own guidelines and principles for developing an application.

This research evaluated major cross-platform mobile development tools, taking Volvo Group Architectural Principles (VGAP) into consideration. The main result of the research was a framework for choosing suitable software tools, for a specific engineering context. During the research, general data was collected in order to provide a full picture of what cross-platform mobile development tools are available on the market. More specific data on five highest ranked tools (PhoneGap, RhoMobile, Titanium, Sencha Touch, and Xamarin) was gathered and the five tools were then carefully evaluated against VGAP.

The current state of the cross-platform mobile development tools market is dynamic, which means that the researches similar to this have to be conducted several times a year. The

situation changes every day and there is no guaranty that a similar research conducted six months in the future will have the same results.

The future work for this research, other than updating the existing data and adding possible new tools, may be improving the collected data by creating prototypes using each tool. The researcher will be able to experience the strengths and shortcomings of each tool while testing them, and therefore provide the research with more accurate and reliable data. Moreover the developed framework can be improved by creating a systematic research guideline for "Refinement" and "Assessment" steps. Finally, triangulation can be added to the method in order to increase the validity of "Recommendations".

## ACKNOWLEDGMENT

We would like to express our sincere gratitude to our supervisor, Dr. Lars Pareto, for the continuous support of our study and for his patience, motivation, enthusiasm, and immense knowledge. We would also like to thank our supervisor from Volvo IT, Micael Andersson, for his advice, assistance, and for providing us with an exciting research topic. Finally, we wish to thank our parents and families for their support and encouragement throughout this study.

## REFERENCES

- [1] Y. e. a. Natchetoi, "Service oriented architecture for mobile applications," in *1st international workshop on Software architectures and mobility/International Conference on Software Engineering*, Leipzig, Germany, 2008, pp. 27–32.
- [2] C. Teng and R. Helps, "Mobile application development: Essential new directions for it," in *Information Technology: New Generations (ITNG)*, IEEE, Ed., Las Vegas, NV, April 2010, pp. 471–475.
- [3] A. Charland and B. Leroux, "Mobile application development: Web vs. native." *Queue - Data*, vol. 9, no. 4, p. 20, 2011.
- [4] "Volvo it is a global company, and a part of the volvo group." [Online]. Available: [http://www.volvoit.com/volvoit/global/en-gb/about\\_us/Pages/aboutus.aspx](http://www.volvoit.com/volvoit/global/en-gb/about_us/Pages/aboutus.aspx)
- [5] J. C. Mitchell, *Concepts in Programming Languages*. United States of America: Cambridge University Press, 2003.
- [6] S. K. Satav, S. K. Satpathy, and K. J. Satao, "Design and implementation of an integrated development environment for multiple languages using c," *International Journal of Advances in Science and Technology*, vol. 2, no. 3, p. 77, March 2011.
- [7] "Software licensing." [Online]. Available: <http://its.uncg.edu/Software/Licensing/>
- [8] X. Shu, Z. Du, and R. Chen, "Research on mobile location service design based on android," in *Wireless Communications, Networking and Mobile Computing*, 2009, pp. 1–4.
- [9] G. Hartmann, G. Stead, and A. DeGani, "Cross-platform mobile development," March 2011. [Online]. Available: [http://www.mole-project.net/images/documents/deliverables/WP4\\_crossplatform\\_mobile\\_development\\_March2011.pdf](http://www.mole-project.net/images/documents/deliverables/WP4_crossplatform_mobile_development_March2011.pdf)
- [10] J. Parker, "Pros and cons of cross-platform mobile app development tools," March 2013. [Online]. Available: <http://www.techrepublic.com/blog/app-builder/pros-and-cons-of-cross-platform-mobile-app-development-tools/2379>
- [11] O. Vogel, I. Arnold, A. Chughtai, and T. Kehrer, *Software Architecture A Comprehensive Framework and Guide for Practitioners*. Springer Berlin Heidelberg, 2011.
- [12] M. Denscombe, *The Good Research Guide for small-scale social research projects*, 4th ed. Open University Press, 2010.
- [13] "Stack overflow." [Online]. Available: <http://stackoverflow.com/>

- [14] M. Korf and E. Oksman, "Native, html5, or hybrid: Understanding your mobile application development options." [Online]. Available: [http://wiki.developerforce.com/page/Native,\\_HTML5,\\_or\\_Hybrid:\\_Understanding\\_Your\\_Mobile\\_Application\\_Development\\_Options](http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options)
- [15] "Html 5 vs native vs hybrid: Is the jury finally in?" March 2013. [Online]. Available: <http://www.cabotsolutions.com/2013/03/html-5-vs-native-vs-hybrid-is-the-jury-finally-in/>
- [16] "Developer economics 2013. developer tools: The foundations of the app economy." 2013. [Online]. Available: <http://www.developereconomics.com>
- [17] J. Rowberg, "Comparison: App inventor, droiddraw, rhomobile, phonegap, appcelerator, webview, and aml," 2010. [Online]. Available: <http://www.amlcode.com/2010/07/16/comparison-appinventor-rhomobile-phonegap-appcelerator-webview-and-aml/>
- [18] "Rhoelements licensing." [Online]. Available: <http://docs.rhomobile.com/rhoelements/licensing>
- [19] "Case study: Imgzine." [Online]. Available: [http://www.appcelerator.com.s3.amazonaws.com/pdf/casestudy\\_imgZine.pdf](http://www.appcelerator.com.s3.amazonaws.com/pdf/casestudy_imgZine.pdf)
- [20] K. Whinnery, "Comparing titanium and phonegap," May 2012. [Online]. Available: <http://developer.appcelerator.com/blog/2012/05/comparing-titanium-and-phonegap.html>
- [21] "Review: Xamarin 2.0 works mobile development magic," April 2013. [Online]. Available: <http://www.infoworld.com/d/application-development/review-xamarin-20-works-mobile-development-magic-217236>
- [22] C. Ramsey, "Phonegap vs. native builds a 12 month review," 2013. [Online]. Available: <http://chelramsey.com/phonegap-vs-native-month-five-of-a-12-month-review/>