Deloitte. UNIVERSITY OF TWENTE.

Deciding between mobile implementation platforms

A multi-criteria decision system for implementation platforms for mobile applications

Master thesis

Ernst Fluttert

June 2013, Enschede

Public

Author

Ernst Fluttert
e.h.fluttert@alumnus.utwente.nl
Student number: 0105066

Master Computer Science
Track Information System Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente
Enschede

Graduation committee

Pascal van Eck
Information Systems Chair
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente, Enschede
p.a.t.vaneck@utwente.nl

Christiaan Katsma
ERP Implementation and Organizational Change Chair
Faculty School of Management and Governance
University Twente, Enschede
c.p.katsma@utwente.nl

Albert te Boekhorst
Service line: Emerging Solutions
Deloitte Consulting, Amstelveen
ateboekhorst@deloitte.nl

Keywords

mobile; smartphone; mobile apps; native app; hybrid app; web app; app criteria; DSS; decision support tool; mobile implementation platform; mobile OS; native vs hybrid vs web apps;

Document info

Revision: 1.8 (Final)
Date: 15 July 2013
Public

Abstract

The current mobile market is segmented into several mobile operating systems: iOS by Apple, Android by Google, WindowsPhone by Microsoft, and BlackBerry by RIM. Each mobile operating system has its own software development kit (SDK), programming language, interface guidelines, and review process. To create a mobile app that works on all of these platforms, called cross-platform apps, there are roughly three options: native apps, web apps, and hybrid apps.
The first option would be to develop a native app for each OS that the organization wants to, which is basically developing one app, four times. The web app option is to develop a mobile-optimized website that can be accessed using a browser, independent of the platform. The last option, hybrid app, uses a wrapper that can extend a web app with native functionality and native distribution.

This exploratory research has several goals. The first is to determine criteria for mobile apps, and their relevance towards each implementation platform (native, web, or hybrid). The second goal, and deliverable of this research, is to develop a decision support system to determine the most favorable mobile implementation platform based on a set of criteria. The final goal is to validate this tool on correctness and practical appliance. In the conclusion this answers the main research question: “How to transform mobile application criteria into decisions for mobile implementation platforms using a decision support system?”.

The mobile app criteria were determined by interviewing experts, which was filtered by applying methods from qualitative data processing. Each interview consisted of questions about business drivers, technical requirements, and any other aspect that related to developing mobile apps. The result is a list containing 16 criteria, where the most relevant are: 1) supporting multiple mobile OS’s; 2) offline usage; 3) utilizing device sensors; 4) distribution via application store; 5) require native functionality; 6) type of content.

The foundation for the decision support tool is multiple-criteria decision making (MDCM). This technique evaluates multiple criteria to select the best alternative based on given input set. Two MDCM types were investigated: multi-criteria value function and analytical hierarchy process. The multi-criteria value function is a simple method where each criterion has a weight and the highest scoring alternative is the best. Analytical hierarchy process also includes priority of the criteria, which consumes more time but also gives a finer granularity than the multi-criteria value function.

The final prototype was based on a multi-criteria value function, as the tool should be used as an indicator. The prototype was tested by a case at a client and further reviews by experts. This test confirmed that the tool operates well in practice; but also needed some refinement in criteria definitions and weights. Further insight in the outcomes of the tool was done by a sensitivity analysis which calculates all possible combinations of criteria. This revealed that the current set of criteria strongly favor native implementations of apps.

Other results of this study are that the prototype is a very practical tool, but has to be updated frequently to keep up with the latest technology developments in the mobile ecosystem. The linearity of the model helps clients to understand how a criterion influences the alternatives. The tool gives a score to each platform and it could happen that these are in close range of each other, which also gives room for interpretation of the result.

Preface

Since my start at the university, I always had an interest in cutting edge developments of new IT related technology. Reading about something new and exciting just asks for hands-on experiments and experience. This ranges from software programs and utilities to concepts and programming paradigms.

When touch-based smartphones emerged, with concepts of marketplaces and apps, it intrigued me tremendously. Nowadays the smartphone has evolved to something much more than a phone with some apps. It is a personal device, on which you communicate with friends and update info on social media.

The point that interests me most is the mobile app. By being a Computer Science graduate it is second nature to make a program on the newly discovered platform and let it magically show the words “Hello world”. Yet with a bachelor in Business & IT, I also like to think about how new platforms can be used in new business propositions.

With this thesis I explored the transformation between the business side of apps and the technical details of the smartphone platform. It was an exciting journey that I could not have completed without the help of my graduation committee.
I want to thank Pascal and Christiaan for their continuous input, pointers and suggestions. With fast responses and critical notes they contribution was invaluable to my research.
Also I want to thank Albert, who has kept me on track and supervised me at Deloitte. When I was stuck he would let me explain it and give insight in creating order in the overwhelming informational chaos. He also was my main guide for the business aspects of this research.

I had the opportunity to do my research at Deloitte Consulting B.V. which gave me the freedom and support to pursue my research into the field of mobile applications. It was a chance to gain hands-on experience in the consultancy world.
A special mention for the interns at Deloitte, for all the useful insights, feedback and interesting chats!

Furthermore I want to thank my parents, friends and family who have supported me throughout the years or just through this thesis! From small conversations about ideas, to overarching concepts in IT, all of this has contributed to my graduation.

Last but not least my girlfriend Maisha, thank you for all your support, time and patience with me.

The future will bring many more apps and technology; let’s see if we can be part of that!

Ernst Fluttert

Contents

List of figures

List of Tables

Glossary

Term	Description
AHP	Analytical hierarchy process (AHP), a decision technique based on hierarchic ranking of priorities
API	Application Programming Interface (API), a specification that developers can use as an interface for their programs
App	See mobile application
Application store	The place where mobile apps can be downloaded for a smartphone. Each mobile OS has its own store, examples are: Play (Android); App Store (iOS); Marketplace (WP);
Chromeless browser	A browser without the user interface aspects like menus, buttons, progress bar and address bar
CSS	Cascading Style Sheet (CSS) is a language for describing the presentation of the HTML elements
Feature phone	A "dumb-phone", indicating a mobile phone that is intended for telephony only: calling, SMS and MMS
HTML	Hyper Text Markup Language (HTML) defines the basic building blocks of webpages
Hybrid app	Using web techniques in combinations with SDK-elements, which places the app between a native and a web app
IDE	Integrated Development Environment (IDE) is an application that provides facilities for software development
Implementation platform	In this thesis an implementation platform is based upon the usage of the presentation layer of the app. This is either native, web, or hybrid;
Mobile application	An application that runs on mobile devices, in this case smartphones and tablets
Native app	Mobile application developed with the SDK of the mobile platform in the default language of the operating system
SDK	Software Development Kit (SDK) is a set of development tools for the creation of applications for a specific platform
Smartphone	A mobile phone that is capable of running general-purpose applications. Usually it has a (large) touchscreen and has continuous connection to the internet.
WDM	Weighted decision matrix is multi-criteria decision technique were each criterion has a weight and the best alternative has the highest score
Web app	Online application that uses web techniques like HTML(5) / CSS and JavaScript for the presentation layer. The entire application runs in a browser. In this thesis it is usually in the form of a mobile-optimized website
WebView	See Chromeless browser
WP	An abbreviation for WindowsPhone, one of the major mobile OS's

1 INTRODUCTION

The mobile market is big and growing: In the first quarter of 2012 a total of 144 million smartphones have been sold. The total market of mobile devices, including feature phones, smartphones, tablets and others, sold a total of 419 million units in the same quarter (Gartner 2012). The sales of smartphones shows an annual increase, which was 44% more in the first quarter 2012 compared to the first quarter of 2011. In Europe mobile device penetration is 83%, this includes feature phones and smartphones. The current smartphone penetration in Europe is 34%, and expected to reach 67% in 2016 (Forrester 2011).

A key capability of a modern smartphone is to customize the experience with mobile apps. These can be searched and downloaded from an online distribution channel, which is by default the vendor's application store. Deloitte expects that the number of available apps from all application stores will exceed two million in 2012 (Deloitte 2012). The explosive consumption of apps makes it important for organizations to utilize this new and personalized mobile communication channel (Deloitte 2012). These mobile apps are very diverse in nature and range from games to weather forecasts and from social media to banking transactions. An example of the importance of apps is emphasized when the biggest social media site, Facebook, announced that they have started an advertisement program to promote mobile apps (Facebook 2012).

The current mobile market is segmented into several mobile operating systems (OS): iOS by Apple, Android by Google, WindowsPhone by Microsoft, and BlackBerry by RIM. Each mobile OS has its own software development kit (SDK), application programming interface (API), programming language, interface guidelines, application store, and review process. It requires a lot of effort and knowledge to develop an app for all of these platforms in their native implementations, basically developing the same app for each platform. Benefits for native apps are the distribution via the application stores, monetization options, and overall responsiveness.

Probably the most important feature on the smartphone is access to the internet; which in turn boosted the development of the browsers within each mobile OS. Most mobile browsers support web techniques like HTML5, CSS3 and JavaScript. The broad support for the open standards of the internet also led towards the web app: a mobile-optimized website. These apps are platform-independent as they only require a browser to render the app. Another advantage is that the presentation is fully customizable and unrestricted, unlike native apps. Web apps are very suitable for cross-platform mobile applications where multiple OS's or many different devices are involved.

A hybrid compromise are hybrid apps, the core of the app is a platform-independent web app, which is packaged in a wrapper that can extend this web app with native functionality. The gain is that the app can be distributed as a native app and have access to device sensors, without developing a native program for each mobile OS. An example of this approach is an app built with the open-source framework Apache Cordova, better known under the name PhoneGap[1].

[1] http://www.phonegap.com/

The hybrid app has only recently been emerging in the mobile ecosystem. A consequence is that there is almost no academic research in this area. This is also reflected in the lack of published academic literature about implementation techniques and technical comparisons between mobile platforms. A goal of this thesis is to deliver a contribution to the academic literature concerning mobile applications.
Not only academia, but also organizations would like to gain insight into the opportunities that mobile apps bring. A goal for Deloitte Consulting is to explore possibilities of mobile apps, where the results of the research can be used in practice by consultants and/or clients.

The main goal of this thesis is **to develop a decision support tool that recommends the best implementation platform (native, web, or hybrid) based upon a set of mobile app criteria**. A major component will be to ascertain the criteria for mobile apps. These are collected by conducting interviews with mobile experts; including the client, the business, and the developer perspective. Another part is to relate these criteria to the implementation platforms for mobile apps. A deliverable for this research is a prototype of the decision support tool. This prototype should be validated for correctness of the criteria, but also for practical usage.

Figure 1 visualizes how these main concepts are linked to each other.

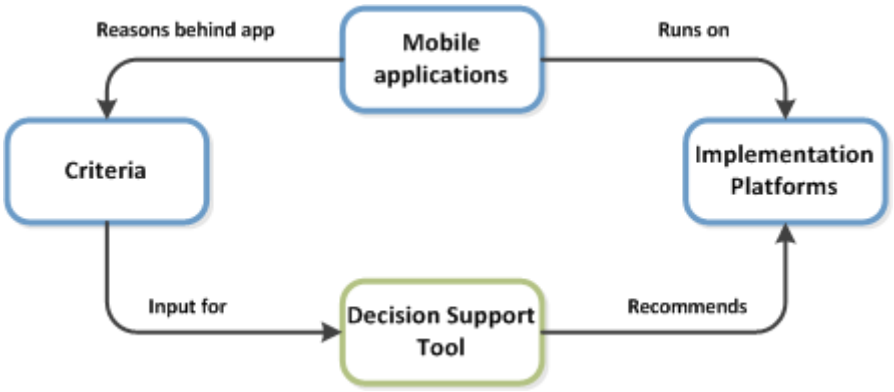


Figure 1: High level overview of research

1.1 App definition

According to the Oxford Dictionary[2] an app is "a self-contained program or piece of software designed to fulfill a particular purpose; an application, especially as downloaded by a user to a mobile device".

This definition touches various distinctive aspects for apps: it is a software component that can be downloaded (self-contained program); it is designed for a specific task/function/feature; it is specifically designed or optimized for usage on a mobile device; and the user (experience) should be a central or integrated component. The presentation, functionality, and features can differ greatly between apps.

[2] http://oxforddictionaries.com/definition/english/app?q=app

A study (Intelligence Group 2012) under the Dutch labor force in the second quarter of 2012 revealed that the following five mobile apps are the most popular: Facebook, WhatsApp, NU, Wordfeud and Twitter. Screenshots of these apps can be seen in Figure 2; from left to right it shows Facebook (iOS)[3], WhatsApp(Android)[4], Nu (iOS)[5], WordFeud (WindowsPhone)[6], and Twitter(Android)[7]. The mentioned apps have several versions so they are compatible with all major mobile operating systems.

Figure 2: Screenshots of the top5 mobile apps under the Dutch labor force

1.2 Cross-platform

Each mobile platform, like iOS, Android, and WindowsPhone are very different for application development. The used software development kit (SDK), program- and interface guidelines, and program languages are just a few aspects in which they differ. Most businesses want to facilitate their app to the whole mobile channel, but do not take these differences into account. The business desire is to have cross-platform applications, not taking into account that a native cross-platform mobile app consumes a lot of development time and resources. Basically the native app would be developed for each mobile platform from the ground up; thus developing the same app

Each mobile operating system has their user interface guidelines (Apple ; Google ; Microsoft) for the presentation of apps to differentiate themselves from competitors. This also makes it necessary for developers to provide a presentation of their app that corresponds to the set of rules for that mobile OS. These include the position of buttons, usage of icons, location of titles, location of menu, and button interaction. Google even showcases the differences between the iOS and Android on Pure Android[8] which is displayed in Figure 4.

To illustrate the differences between the look & feel in the four mobile OS's, the Twitter App is displayed in Figure 3, from left to right, in Android, iOS, WindowsPhone, and BlackBerry. Differences can be spotted in the title-bar, tabs, icon usage, text-styling, and screensize. Beyond the

[3] http://itunes.apple.com/us/app/facebook/id284882215
[4] https://play.google.com/store/apps/details?id=com.whatsapp
[5] http://itunes.apple.com/us/app/nu/id294726570
[6] http://www.windowsphone.com/en-US/apps/4c049346-c285-4bde-ba34-1fa5f3df4562
[7] https://play.google.com/store/apps/details?id=com.twitter.android
[8] http://developer.android.com/design/patterns/pure-android.html

presentation layer there are hardware differences that influence the input to the app like a physical or software “back-button”. Most BlackBerries include a hardware QWERTY keyboard. Another example can be found in APPENDIX A: EXAMPLES OF APPS on page 61.

Figure 3: The Twitter app on Android, iOS, WP7, and BlackBerry (from left to right)

Not only the presentation layer is different, the underlying OS requires knowledge and experience. The classic tutorial “Hello World” illustrates this further, where the application shows very different programming languages, IDE’s, and use of libraries and API’s. See APPENDIX B: NATIVE CODE on page 62 for the actual examples.

This leads to the observation that apps that are specifically developed for a mobile OS, will not automatically work and perform equal on other platforms. The point is that a developer that creates native apps for iOS, is not always capable do produce the same app for Android, WindowsPhone, or BlackBerry and vice-versa.

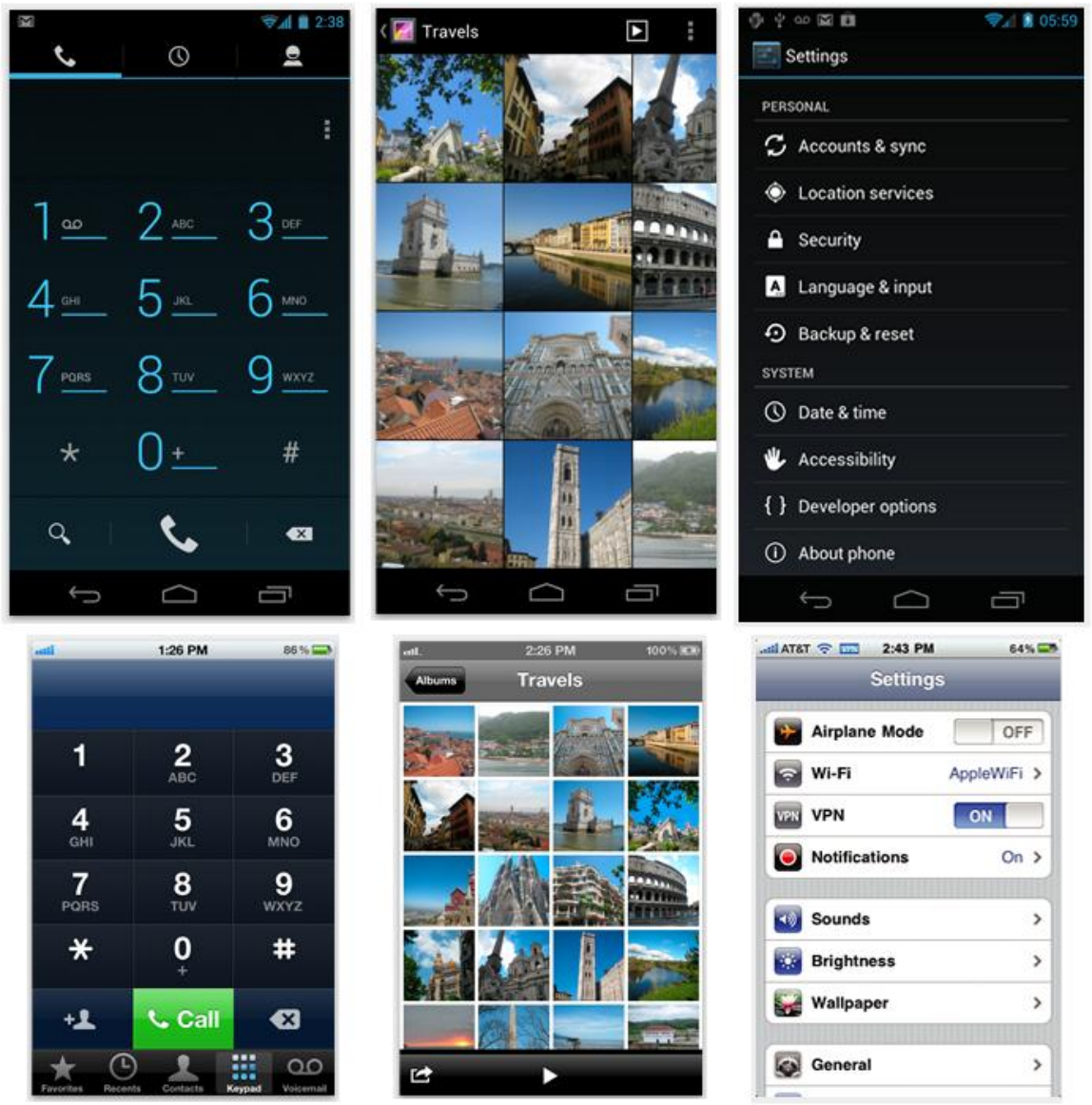


Figure 4: Different native screens, Android on the top row, iOS on the bottom row

1.3 Outline

The methods and methodologies used in this thesis are described in the research setup (chapter 2). The methods will be explained in more detail, and their appliance upon this research. In chapter 3 the various implementation platforms are further investigated and compared to each other. App criteria, like drivers and requirements, are collected and analyzed in chapter 4. These two chapters form the input for the decision tool in chapter 5, where the decision model and analysis can be found. In chapter 6 the prototype tool will be validated. The thesis ends with the conclusion in chapter 7 and discussion in chapter 8.

The outline of the thesis and how the chapters relate to each other are visualized in Figure 5.

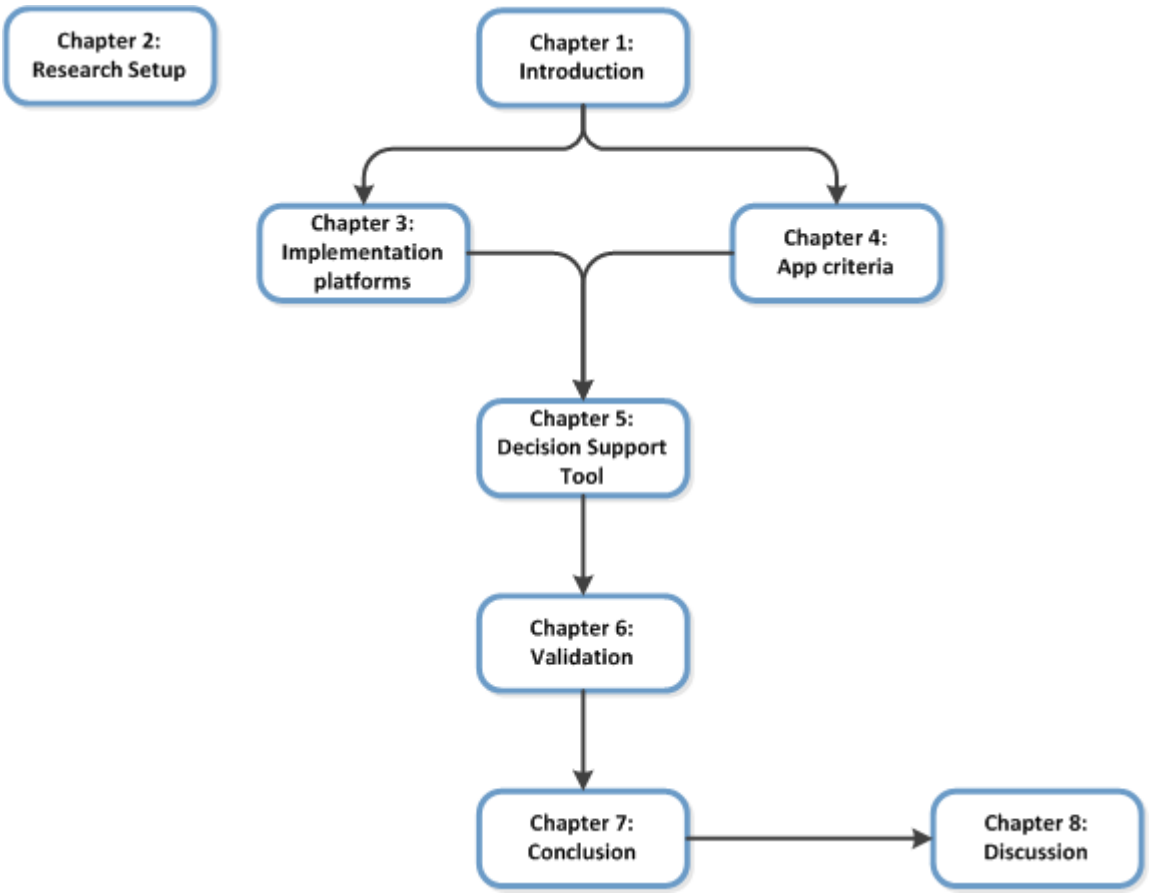


Figure 5: Thesis outline

2 RESEARCH SETUP

This chapter explains the overall goals and approach of the research. Then it describes the research questions and the methodologies to be utilized in answering the questions and reaching the overall goals.

2.1 Research goal

This explorative research informs about, and analyzes, the current mobile apps landscape and techniques where special interest is taken into the implementation platforms and frameworks. It determines several factors and criteria that drive organizations to build mobile apps. These criteria range from business oriented criteria, to more specific technical aspects of a mobile app.

Each criterion has a score on how well it is supported by each implementation platform (native, hybrid, and web). Based on these criteria, a decision support system determines the preferred implementation platform for a certain app. This decision support system provides more transparency to which implementation platform is preferable for their specific app; without having to be a mobile and/or technical expert.

The added value to the academic literature is to add information via the thesis for mobile implementation platforms; the current mobile ecosystem; and criteria for mobile apps.
The practical value comes in the form of the decision support system, which can be used to determine the preferred implementation for a mobile app based on the input of the client.

This results in the following research goals:

1. Explore the major criteria for mobile applications and implementation platforms
2. Develop a decision support system in which these criteria can be translated into platform preferences

2.2 Research approach

To create a validated decision support system, it is imperative that it is built on solid theory and methods. As a starting point two general research approaches have been used to setup the research: Designing Research (Verschuren and Doorewaard 2007), and the Engineering Cycle (Wieringa 1996). These methods describe ways to start analyzing the problem that the research tries to solve. It helps to construct valid research questions, conceptual models, and planning.

Designing Research has seven phases in which the research is clearly structured into a usable form (Verschuren and Doorewaard 2007). These phases explore the project framework, create a research model, defines key concepts (and research questions), and determines the strategy and planning. It helps into analyzing the research problem and translating them into a clear game plan and conceptual model.
A construct from Designing Research is to formulate the goal of the research in one sentence that has the form: the goal of this research is A through B; Where A is the external part that has a core description of the contribution that the research tries to deliver; and B is the internal part that describes how this contribution is realized(Verschuren and Doorewaard 2007). The resulting goal for

this research is: The development of a decision support system based on criteria and requirements of mobile applications to determine the most preferred implementation platform through literature study and conducting expert interviews that can be analyzed and processed into a data structure for the tool, which is then validated by follow up interviews and a validation experiment.

The other starting point is applying the Engineering Cycle (Wieringa 1996), which is a five-phase cycle that can be applied to products and services. The phases are: problem analysis, solution design, design validation, design implementation, and implementation evaluation; see also Figure 7. It is a structured way to analyze the problem that leads to developing, validating and implementing the solution. A main concept in the theory is to state the engineering argument, which describes how the solution and domain interact to reach the goal (Wieringa 2009). For this thesis the argument would be: By increasing the knowledge of mobile applications and implementation platforms with a decision support system, it will help to transform mobile app criteria into deciding for the right platform.
The Engineering Cycle shows the lifecycle of the decision support system that this research delivers. Another reason to use this method is that this research follows the same pattern as the engineering cycle; this also shows a clear relation between the research questions and the phase that the tool is in. This makes conceptualizing and reflecting the steps towards the final prototype tool an easier task.

2.2.1 Conceptual models

Figure 6 visualizes the conceptual model, as described in Research Design (Verschuren and Doorewaard 2007). The vertical arrows are the confrontation between two research object, resulting in a horizontal arrow that is the product or result from this confrontation.
There are five distinct consecutive stages in the model, shown in Figure 6. It starts with reading mobile platform documentation, conducting preliminary interviews, and reading literature about mobile apps. This results in an initial list of criteria and implementation platforms, which is then confirmed and updated with experts via interviews. From this a final list of implementation platforms and a final list of app criteria are formulated. Together with literature from Decision Support Systems (DSS) form the foundation for the decision tool. With a validation session and expert opinions the tool is validated and leads to the final conclusion of the thesis.

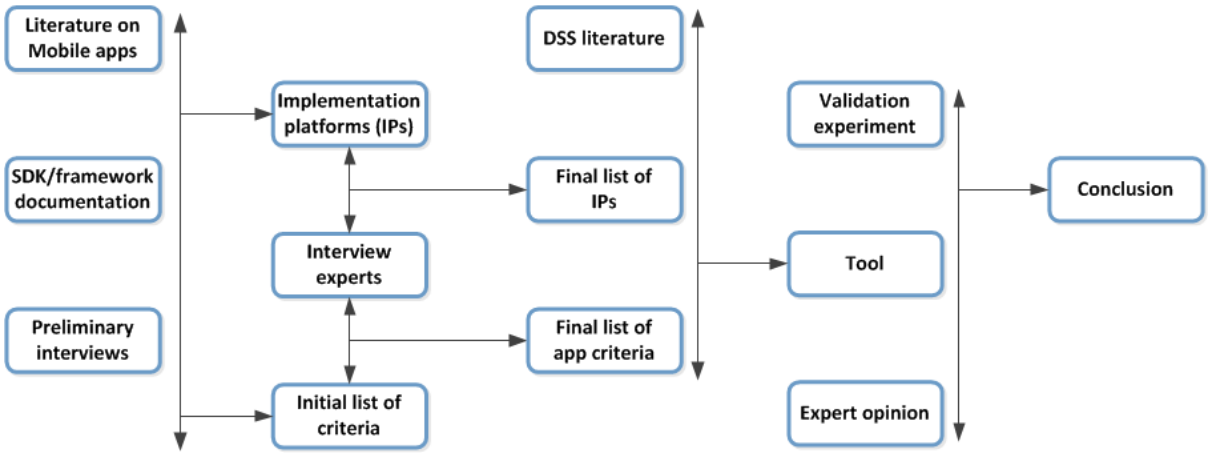


Figure 6: Conceptual model of research setup, based on (Verschuren and Doorewaard 2007)

As mentioned earlier, the Engineering Cycle is used to guide the development of the decision tool. This engineering cycle is visualized in Figure 7, based on the work of Wieringa (Wieringa 1996; Wieringa 2009). The problem analysis phase determines what the (real) problem is by identifying the stakeholders and investigating the problem, which results in requirements for the decision tool. The output of the solution design phase is the artifact itself, in this case a prototype of the decision support tool. In the third phase the tool is validated, by checking if the requirements from the problem analysis phase are met. The design implementation phase is concerned with implementing the proposed solution in the entire organization. The implementation evaluation phase is to check if the design implementation phase meets the requirements that were set earlier. If not the cycle repeats itself, as mostly it will be a continuous process (Wieringa 2009).

In this thesis the first three phases will be followed, which is reflected in the structure of this document and formulating the research questions.
Due to the limited period in which this research is carried out, it will not be possible to go through the design implementation phase and the implementation evaluation phase. These last two phases are only relevant when the tool would be incorporated into the business processes and be evaluated. As the goal is to deliver a prototype it is also not necessary to go through these phases.

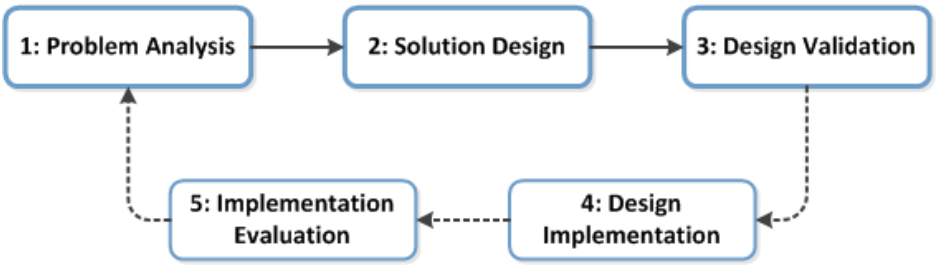


Figure 7: The engineering cycle, adapted from (Wieringa 2009)

2.3 Research questions

Based on the research goals and conceptual models, the main research question has been formulated as:

> *How to transform mobile application criteria into decisions for mobile implementation platforms using a decision support system?*

To further decompose the main research question, six sub questions have been formulated:
Q1: What are the implementation platforms in the current mobile ecosystem?
Q2: What are key criteria for mobile applications according to mobile experts?
Q3: How do the mobile criteria relate to the mobile implementation platforms?
Q4: What decision support system methods are suited for linking mobile app criteria with the implementation platforms?
Q5: Which decision support system can be transformed into a prototype decision tool?
Q6: What is an appropriate way to validate the design?

The sub questions can be closely related to the different stages in the engineering cycle. An overview can be seen in Table 1.

Phase	Questions	Description
1: Problem Analysis	Q1, Q2	Investigate into the current mobile ecosystem, to determine the primary criteria and key information
2: Solution Design	Q3, Q4	Transforming the criteria into information for the decision support tool
3: Design Validation	Q5, Q6	Validation of the decision support tool
4: Design Implementation		Out of scope
5: Implementation Evaluation		Out of scope

Table 1: Mapping of subquestions to the engineering cycle

2.4 Methodology

This research employs multiple methods to collect and transform data into usable information. The four methods used are:

- **Literature study**: Scientific articles, journals and books form the backbone of academic theory;
- **Documentation**: Some articles, specifications and blog-items are not considered literature, but still contribute in providing insight and detailed information about some of the subjects;
- **Interviews**: To retrieve information from experts, interviews are used to extract, add and validate key concepts in the decision support models;
- **Validation experiment**: A validation experiment is conducted to see the how prototype behaves in a real-life setting. The experiment is mostly based on a mix of literature from interviews and case studies (Yin 2009).

A visual representation of where the methodologies are applied in the thesis subjects can be found in Figure 8. Literature study is not mentioned as it is ubiquitously applied, alongside the other methods.

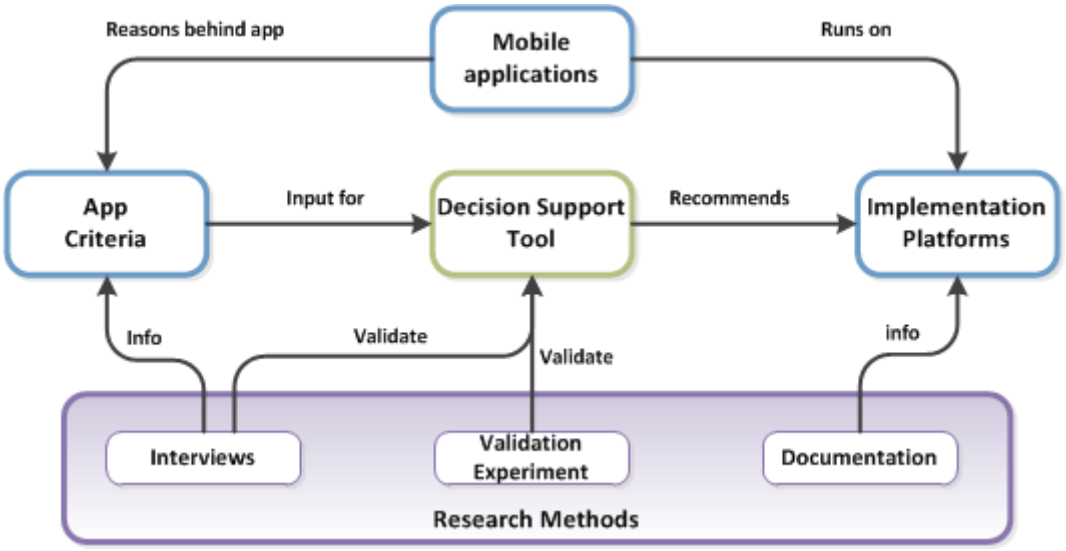


Figure 8: Mapping research methodologies to parts of thesis

2.4.1 Literature

The academic literature is a solid base for all other research methods, but also for the app criteria and implementation platforms. To structure the approach a systematic literature review process is employed (Kitchenham 2004; Budgen and Brereton 2006; Kitchenham, Pearl Brereton et al. 2009)

The main difference between an ad-hoc literature search and a systematic approach is that the systematic approach has a review protocol, a documented search strategy, and a review process with specifications for the information that is to be obtained (Budgen and Brereton 2006; Kitchenham, Pearl Brereton et al. 2009).

The review protocol describes what research question is being addressed and which methods are employed in the review process (Budgen and Brereton 2006). It also describes the necessity for a systematic literature review (Kitchenham 2004). In this thesis the research questions for the protocol can be the same as the ones provided in chapter 0. Especially Q1 and Q4 will primarily be based on literature, supported by market analysis reports and other sources.

The search strategy is to start with a general term formulated in the question, read a few relevant papers and refine the search terms with more specific terms or add other terms with the "AND" and "OR" keywords. The results of the search terms were cross-referenced between different scientific search engines: Scopus[9], Web of Science[10] and Google Scholar[11].
The review process consists of steps to determine how relevant the articles are, and how to synthesize data from it (Kitchenham 2004). As the current apps are emerging since around 2007 with the introduction of the iPhone, the related articles should be very recent: from 2008 or newer. Indicators that are used for articles are: the author(s) has published more than one paper on the particular subject; the article comes from peer-reviewed or high quality sources; number of citations; and references from the article.

Table 2 shows the used keywords for certain areas that reflect in some of the research questions. It also includes examples of the found literature from these queries.

Area of search	Question related	Search query	Examples of found literature sources
Literature approach	-	"performing systematic literature review"	(Kitchenham 2004); (Budgen and Brereton 2006); (Kitchenham, Pearl Brereton et al. 2009);
Mobile ecosystem	Q1	"mobile AND HTML5" "mobile AND application AND development"	(Padley 2011); (Wasserman 2010);
Key app criteria	Q2	"mobile AND native AND web" "mobile AND application AND criteria"	(Charland and Leroux 2011); (Işıklar and Büyüközkan 2007)
Decision support systems	Q4, Q5	"multi-criteria decision making" "multi-criteria decision making" AND "decision matrix" "multi-criteria decision making" AND "AHP"	(Saaty 1980); (Dyer and Sarin 1979); (Zanakis, Solomon et al. 1998); (Arnott and Pervan 2005);
Validation by experiment	Q6	"validation AND software engineering" "validation AND case study"	(Zelkowitz and Wallace 1998); (Hevner, March et al. 2004); (Yin 2009)

Table 2: Keywords for literature searches

[9] http://www.scopus.com/
[10] http://webofknowledge.com/WOS
[11] http://scholar.google.com

2.4.2 Documentation and non-academic sources

Due to the limited amount of available academic literature which covers mobile apps, also non-academic resources have been accessed. Key examples are the developer portals[12][13][14][15] for each mobile operating system which hosts the official documentation and useful resources. These source are used with regard to native apps.

Using the Deloitte network, this research could benefit from the TMT predictions (Deloitte 2012); but also use Gartner and Forrester for analysis of the current markets (Forrester 2011) and (Gartner 2012). These are used in the introduction and parts regarding the general mobile market and mobile operating system market shares, like chapter 3.2.

Other supporting sources like blogs can be used for gathering opinions on subjects, but these should be relatively established and trusted sites. These mainly concern hybrid apps, chapter 3.4.

2.4.3 Interviews

Interviews are used to validate, delete, or add criteria to the model for the decision tool. To extract this information a semi-structured interview approach is used where there is a protocol used to lead the interview. Each interview consisted of five open questions for the experts to gain more knowledge about the aspects that are important to mobile apps.
The conducted interviews were qualitative in nature, in the form of a conversation. A semi-structured interview has the advantage to set out the big picture, while leaving room for (social) interaction: deviation on topics and/or further questioning on specific topics. A structured interview (or questionnaire) is most likely too limited for our goal to gain insight, where we need the flexibility to ask further, go into details, or deviate from the set of questions. An unstructured interview could work, but is not preferred as we would like to loosely compare answers on several topics which require general questions (Saunders, Lewis et al. 2009; Yin 2009).

The complete protocol and questions can be found in APPENDIX C: INTERVIEW PROTOCOL (page 61).
There were a total of nine persons that participated in the interviews; the date and their relation to mobile apps are listed in Table 3. It also states when the interview took place and which relation the interviewee has with mobile apps.
Two experts (#8 and #9) were consulted in a follow up interview to give an opinion on the tool, thereby judging and reviewing the tool for validation purposes.

	Interview date	Relation to mobile apps
Interview 1	14 June 2012	General Application development
Interview 2	15 June 2012	Consulting on mobile app development
Interview 3	18 June 2012	Develops in-house application
Interview 4	20 June 2012	Develops in-house application
Interview 5	21 June 2012	App development in specific sector

[12] http://developer.apple.com
[13] http://developer.android.com
[14] http://developer.windowsphone.com
[15] http://developer.blackberry.com

Interview 6	27 June 2012	Manager of mobile website
Interview 7	03 July 2012	General Application development
Interview 8	17 July 2012	General Application development
Interview 9	17 July 2012	General Application development

Table 3: Interview participants list

Preparing the semi-structured interviews, additional papers were read specializing in interviewing within the computer science realm. Hove and Anda summarized experiences of 280 semi-structured interviews that the authors conducted in the field of software engineering. They describe several activities: scheduling; collecting of background information; preparing interview guides; discussions/meetings; summary writing; transcribing. They also include practical advice, based on recommendation found in literature. The paper describes six types of questions and how these can be applied in different interview techniques (Hove and Anda 2005).
Myers and Newman use the metaphor of a drama play (dramaturgical model) to conceptualize the aspects within an interview: stage, actor, audience, script, entry, exit and performance. This makes many of the intangible aspects more understandable. It describes the assets in detail, gives recommendations, and contains eight guidelines for better qualitative interviews. This literature was also specialized in IS research (Myers and Newman 2007).

All interviews were recorded, transcribed and summarized. The summaries can be found in APPENDIX D: SUMMARIES OF INTERVIEWS (page 66). These will be further analyzed in chapter 4.

Using qualitative data processing (Miles and Huberman 1994), the transcripts and summaries from the interviews can be transformed into usable information. For this research it was used to extract relevant criteria from the interviews.
Miles and Huberman suggest that the analysis consists of three procedures. The first is data reduction, where the mass of the qualitative data is reduced and organized in for example summaries where all irrelevant data is discarded. At this stage it is encouraged to start with coding the data. The second procedure is to display the data from the first stage in charts, tables, and other graphical formats; this is essential to draw conclusions. The last procedure is drawing conclusions and validation; the analysis should develop initial conclusions that can be validated by reference of the existing field data or with further data collection.

An important part is coding the summaries of interviews; this is the part where structuring and clustering of the criteria are formed. Based on Saunder, Lewis et al (Saunders, Lewis et al. 2009), the steps in this process are:

1. Transcribe the interview;
2. Mark keywords in each interview, to form categories (open coding);
3. Recognize relationship between categories (axial coding);
4. Defining the criterion and relative weight, which is the integration of categories into theories (selective coding).

The results of the qualitative data processing, analysis of the interviews, and graphical representation of the coding process are further described in chapter 4.2 on page 31.

2.4.4 Validation experiment

A validation experiment was conducted at a client of Deloitte to test the theoretical models that the research delivered. The experiment consists of several components: a short qualitative interview, model testing, and reflection on the results of the session. This is based on literature from semi-structured interviews as well as case study research (Eisenhardt 1989; Hevner, March et al. 2004; Yin 2009; Thomas 2011). The goal of this experiment was to get hands on experience, determine the practical usage of the prototypes, and to decide which DSS method (simple additive weighted value function; or analytical hierarchy process) should be used in refining the tool.

The qualitative interview is the first stage of the interactive session, where open questions are asked about the app that the customer wants to realize. The interview format gives room to let the client describe the app criteria in their own words, and prevents bias from the interviewer. The open questions are formulated from a business perspective. It is expected that some questions could suggest the usage of one or more app criteria. After the interview the list of criteria is summarized and checked with the client.
The testing component of the experiment is by entering the criteria in the prototypes. These prototypes will return the preferred platform instantly.
The evaluation of the results, in conjunction with the preference and feedback from the client provided valuable input for further improvement on the final prototype.

The results of the experiment are found in chapter 6.1 on page 45. The complete protocol for the experiment can be found in APPENDIX H: VALIDATION EXPERIMENT PROTOCOL on page 105.

2.4.5 Validation review

In addition to the practical experiment, the tool was also reviewed by three mobile experts for further validation of the tool. Two of these reviews were follow-ups with interviewee #8 and interviewee #9 from the first series. The last interviewee was a mobile app expert, which did not have previous knowledge of this research.

Two of sessions are based on semi-structured interviews where the focus lies on validating the criteria and corresponding norms. Another goal is to validate the tool itself and asses the usefulness. For the other sessions, a review package was prepared which was reviewed without the presence of the author. The results of these validation sessions are found in chapter 6.2 on page 48.The complete protocol for these interviews can be found in APPENDIX J: VALIDATION SESSION PROTOCOL on page 115.

2.5 Research sponsor

The Dutch member firm of Deloitte Touche Tohmatsu Limited (from now on referred to as Deloitte) sponsored this thesis project. Deloitte is one of the four world biggest accountancy firms. Besides auditing services, Deloitte also provides consulting services under the label Deloitte Consulting. This research was executed at the service line Emerging Solutions, which is a part of the Technology branch within Deloitte Consulting.

2.6 Research summary

A structured approach is used to reach the goal of further investigating the relation between mobile app criteria and implementation platforms.

The main research question is:

> *How to transform mobile application criteria into decisions for mobile implementation platforms using a decision support system?*

This is further divided into the following sub questions:

Q1: What are the implementation platforms in the current mobile ecosystem?

Q2: What are key criteria for mobile applications according to mobile experts?

Q3: How do the mobile criteria relate to the mobile implementation platforms?

Q4: What decision support system methods are suited for linking mobile app criteria with the implementation platforms?

Q5: Which decision support system can be transformed into a prototype decision tool?

Q6: What is an appropriate way to validate the design?

Table 4 shows in the chapters in which the sub question are covered, including the used methods.

Question	In chapter(s)	Used method(s)
Q1	3, 4	Literature
Q2	4	Interviews with experts
Q3	5	Theory building (based Q1 and Q2)
Q4	5	Literature
Q5	5	Literature
Q6	6	Validation Experiment, Interview

Table 4: Mapping research questions to methods

3 IMPLEMENTATION PLATFORMS

This research makes the distinction between three implementation platforms: native, web, and hybrid. This is based on the presentation layer of an app; which can use native elements, but also rendering through a browser using web techniques. This choice is related that web techniques are available on each mobile platform and has great potential for cross-platform techniques.
This chapter will explore the three implementation platforms, which extends the information from chapter 1.2. In chapter 4 the criteria for apps for each platform will be investigated.

3.1 Implementation platform overview

The first implementation platform is composed of native apps. These apps run only on the OS they were designed for, like specifically for Android, or specifically for iOS. The second implementation platform consists of web apps: mobile-optimized websites that can be accessed by any (mobile) browser. The last implementation platform consists of apps that are a mix of the first two groups, the so called hybrid apps. Third party applications like Phonegap; Appcelerator; Rhodes; and Appspresso; make it possible to wrap web apps into native browsers and/or code, bridging the gap between web and native apps. This is all based on the philosophy of "code once, deploy-to-many".

Figure 9 supplies an overview of the implementation techniques; this figure is deduced upon knowledge from mobile OS's, but also inspired upon various articles (Google ; W3 Cubes 2011; GeoChalkboard 2012) which describe anatomy of the apps.

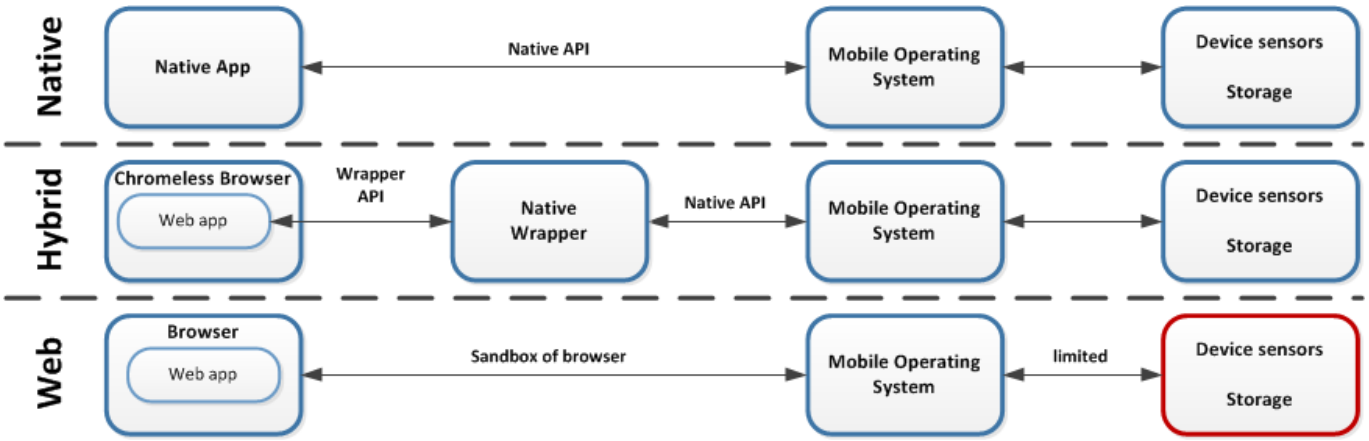


Figure 9: The three approaches: Native, Hybrid, and Web apps

The image visualizes that native apps communicate directly with the operating system, using the native application programming interface (API).

The hybrid approach uses a chromeless browser for rendering, but the app can still access the native API by using a native wrapper which facilitates programmable bindings for native functions. A chromeless browser (or WebView) is the native browser without any interface like buttons or progress bars; its main purpose is to render HTML that is given through a native wrapper.

The web app utilizes the browser for rendering HTML on the mobile device, hereby sacrificing all native functionality as the browser runs the app in a sandbox. A sandbox is an environment with limited permission and functionality.

An obvious difference is the methods used to communicate with the mobile operating system (OS). A native implementation of an app provides all features of the application programming interface (API) of the OS. The hybrid approach is limited by the functionality specified in the native wrapper; which mostly acts as an intermediary or as a shell without business logic. The web app is limited by the privileges and functionality of the browser.

Explicitly stated on Figure 9 is communication with the device sensors like GPS, camera, compass, accelerometer, barometer, proximity sensor, etc. The rules that apply on for the mobile OS can be directly translated towards these sensors. A native implementation can take full advantage of all aspects of the sensors, which is a result of programming in the native language of the OS. The hybrid approach is limited by the functionality of the native wrapper, which can be programmed to support various sensors. A web app is fully dependent on the browser and thus its limited access to these.

Also important for apps feature is the possibility to access and store data on the phone; for example contacts, user-sessions, preferences, achievements, etc. Also the same principle applies here as with the sensors. A side note is that web-apps and hybrid apps are inclined to store most data online on servers and/or the cloud; but storage on the phone is essential when no connection with the internet is available.

Although not stated on the diagram, but one of the typical properties of a mobile app is the distribution through an application store. Native and hybrid apps can use the application store of the mobile OS's vendor. This also increases the options for monetization, by the applying the store purchase and transaction model. Web apps can only be accessed by a browser, they don't require distribution.

Figure 10 shows the advantages of each approach, which will be elaborated in chapters 3.2, 3.3 and 3.4.

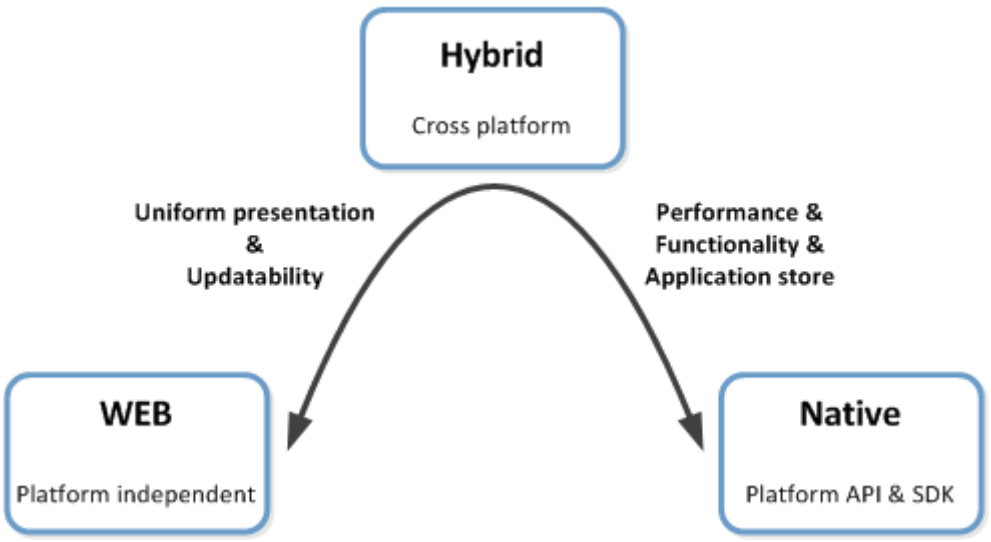


Figure 10: Positioning of implementation platforms

3.2 Native platform

The current mobile ecosystem is dominated by four big players: Apple (iOS), Google (Android), Microsoft (Windows Phone) and RIM (Blackberry); and each OS has its own development language, SDK, API's, and application store. Apps that are developed for a specific platform with the associated SDK and API are called native apps. A number of these characteristics are compared in Table 5.

Platform	Android	iOS	Blackberry	WindowsPhone
Vendor	Google	Apple	RIM	Microsoft
Language	Java (ADK)	Objective C, C++	Java (J2ME)	C#, .NET
App Extension	.apk	.app	.cod	.xap
IDE	Eclipse	XCode	Eclipse	Visual Studio
Application Store	Play	iTunes	App World	Marketpace
Source	Open Source	Proprietary	Proprietary	Proprietary

Table 5: Overview of smartphone and tablet operating systems

The big advantage of native apps is that they communicate directly with the API of the OS. This ensures that the app-developer is in total control of all aspects of the smartphone. As they communicate with the OS directly, the overhead is minimized and relate to high performance.

The main disadvantage of developing for a native platform is that an app cannot be transferred to another platform, so they are not cross-platform. In other words, an iOS application will not run on an Android-phone. Native development requires developers that know the programming language and guidelines for that platform; usually they are specialized in one native platform.

Figure 11 displays the smartphone OS market share based on smartphone sales in the second quarter of 2011 and 2012 (IDC 2012). The total market in the second quarter in 2012 was 154 million smarthphones, which was 46 million more than in 2011(IDC 2012). It also shows how rapid the landscape can change in a relative short period.

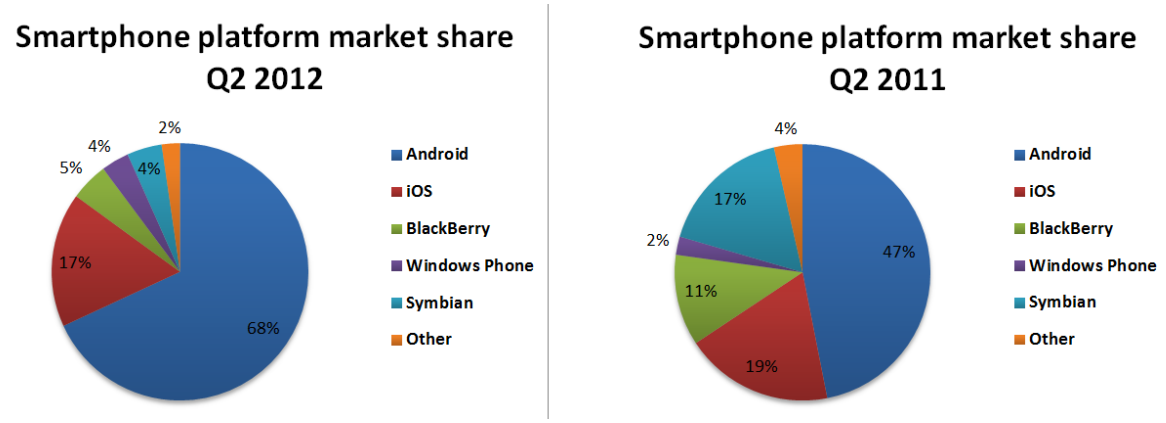


Figure 11: Smartphone platform market share, based on (IDC 2012)

In this year Android has increased their market share from 47% to 65%, and is the absolute leading platform. Although Android has the biggest market share, it is fragmented by the many versions that are currently in use. The majority is still on version 2.3 “Gingerbread” with 45%, followed by 4.0 “Ice Cream Sandwich” with 29%. The newest version “Jelly Bean” is only used by 13.6% (Google 2013). The implication for apps is that most Android apps have to be compatible with the relative old “Gingerbread” versions.

iOS from Apple, has a steady market share of around 18%. The iPhone is a high-end smartphone that has almost no fragmentation. Adoption to the newest version is relative high. The current iOS version 6 is adopted by 85%, based on various internet sources[16] but no official statistics are known.

BlackBerry from RIM saw their market share drop from 11% to 5%, and is still declining. These smartphones have their roots in the business segment, but competitors are catching up quickly. The WindowsPhone platform by Microsoft has recently overhauled and rebranded their entire mobile OS from WindowsMobile to WindowsPhone; which places them in an underdog position. It is unclear at the moment if this radical change will also result in a bigger market share.

A potential fifth player is Nokia’s Symbian, but Nokia will cease to support Symbian and focus on WindowsPhone. This means no, or very few, new smartphones for the Symbian platform, which will eliminate the Symbian platform from the market (Nokia 2011).

[16] http://chitika.com/ios-version-distribution

3.3 Web platform

Web apps are mobile-optimized websites, built with web techniques like HTML5, CSS, and JavaScript. The websites are viewed through the (native) browser, which implicates the smartphone needs to be online and connected to the internet for accessing the web app.

A major advantage is that this approach is platform-independent; it works on any mobile device that has an internet connection and a browser. Any web-developer can create a web app without specific knowledge about the mobile OS's, as it is basically a website suited for a small display. Responsive and/or fluid designs are often used to display the app universally on different screensizes. Webservices can be used to communicate data between the app and the server.

The main disadvantage of the approach is that the app is limited by the browsers sandbox, so it has limited capabilities to interact with the phone itself. This usually means (very) restricted access to anything non-web. For storage a web app can use the LocalStorage, as defined in the HTML5 specification. There are also proposals for GPS and camera support directly from the browser.

Another disadvantage is that a web app cannot be distributed through a vendor's application store, nor will it be available like a native app on the phone. This also implies that updates to the app are independent of the application store and do not require any update action from the end-user.

3.4 Hybrid platform

The principle of a hybrid app lies in the fact that it combines features from both a web app and a native app. It combines the advantages of a native app, by having a native wrapper. This wrapper is a native program that encapsulates the web app and offers bindings for native functionality. This can be a framework, where the bindings are generalized to fit the usage on multiple platforms. The wrapper is an actual native app, which can be distributed through the vendor's application store.

For the presentation and logic, a web-app runs in a chromeless browser which is controlled by the native wrapper. A chromeless browser is the native browser, without any user interface elements like the menu and the typical generic browser buttons "back" and "home". This increases the uniform appearance on different platform, using HTML, CSS and JavaScript. The downside is that it cannot use the native user interface elements; although some web frameworks mimic it closely.

The hybrid approach combines the advantages of a web app for cross-platform and uniform presentation, while it adds the advantages of the native app by using distribution through the application store and access to device sensors and storage.

4 MOBILE APP CRITERIA

This chapter explores which mobile app criteria can be used for determining a preference between the native, hybrid and web approach. These criteria are collected using semi-structured interviews with mobile experts. First the data collection process is explained, followed by the results from the interview which delivers a list of criteria for mobile apps. It will then present the transformation towards the final list of criteria.

4.1 Data collection

The semi-structured interview had all the same schema; five open questions for the experts to learn which criteria were important for mobile apps.

1. What is your experience with mobile applications and implementation platforms?
2. What are (common) business drivers behind the mobile apps?
3. What are (common) implementation requirements for applications?
4. When is an implementation platform preferred above another platform?
5. How do company competencies impact the choice for a mobile platform?

The first question is important to establish a baseline, to determine the experience with one or more mobile OS's; but also to determine the experience with the different implementation platforms (native, web, and hybrid). The second question is to determine the drivers of apps; this is why customers want a mobile app. The third question is how they envision the app, based on requirements. The fourth question collects the opinion of the interviewee, to come up with situations where a particular implementation technique would be chosen above the other. The last question is about the constraints that might be coming from the customer, which might not always be directly visible.

The complete interview protocol can be found in APPENDIX C: INTERVIEW PROTOCOL on page 62. Each interview was transcribed, summarized and sent back for approval by the interviewees. Processing and extracting data is based on qualitative data processing as described by Miles and Huberman (Miles and Huberman 1994) , also described in chapter 2.4.3. The processing of the interview has the following steps, according to Saunder, Lewis et al:

1. Transcribe the interview;
2. Mark keywords in each interview, to form categories (open coding);
3. Recognize relationship between categories (axial coding);
4. Defining the criterion and relative weight, this is the integration of categories into theories (selective coding).

The complete summaries of interviews can be found in APPENDIX D: SUMMARIES OF INTERVIEWS starting on page 66.

4.2 Interview results

An initial set of criteria from preliminary interviews, combined with literature research, was supplemented with the criteria that came up during the interviews. These criteria were distilled using open and axial coding of each interview, which can found in APPENDIX E: CODING INTERVIEWS starting on page 92.

From the interviews the criteria-set was hierarchical ordered, as some criteria overlapped. A total of thirteen aggregated categories for mobile apps are listed below. The detailed hierarchical ordered list can be found in Table 6 which marks the criteria that came up or were discussed with the experts. .

1) **User Interface**: Criteria that have requirements on the user interface part of an app. The level of responsiveness that is required for app types. The influence of screensizes and how sophisticated a user interface must be
2) **Offline access**: The importance of online/offline access to apps
3) **Device capabilities**: The sensors that apps need (GPS, camera, etc) and other wishes for the device
4) **Visibility**: The visibility in the application store, and the impact of this feature
5) **Multiplatform**: Running apps on multiple native platforms
6) **App development**: Aspects of development of apps; from the skills required for developers to open source. From used IDE's to outsourcing and update cycles
7) **Business Model**: The business reasons behind an app, like the proposition, budget, and goal of apps
8) **Targetgroup**: The intended platform and/or ecosystem, aspects like high-end market or budget and BYOD(bring your own device)
9) **Security**: The security aspects for app, like encryption of data and mobile device management.
10) **Platform features**: General features for the native/web/hybrid platform.
11) **Future proof**: How will the app hold up in the near future
12) **Preference**: Preference of company, executives, and company policy
13) **Personalization**: The importance of personalized content for a user (user based content)

TOPICS / INTERVIEWEE	#1	#2	#3	#4	#5	#6	#7	#8	#9
1. User Interface									
a. Responsiveness			x			x	x	x	x
b. Animations and sophistication:	x		x	x	x	x	x		
c. Screensize	x	x	x						x
2. Offline access:					x		x	x	x
3. Device capabilities:									
a. Sensors	x	x	x	x	x	x	x	x	x
b. Special needs		x					x		
4. Visibility	x		x		x	x	x	x	x
5. Multiplatform		x					x	x	x
a. Portability			x		x	x			
6. App development									
a. Skillset required by developer	x		x	x	x			x	x
b. Tooling / IDE	x								
c. In-house vs outsourcing		x	x	x		x	x	x	
d. Open source	x		x	x	x		x		
e. Updatability		x	x			x	x		
7. Business Model									
a. Business case		x	x	x			x	x	x
b. Budget	x	x			x	x	x	x	
c. Support & SLA	x	x		x			x		
d. Purpose / goal of app		x	x						
8. Target group									
a. Heterogeneity		x	x	x		x			
b. BYOD (bring your own device)					x				
c. Hi-end or low-end market?	x			x					
9. Security									
a. Mobile Device Management		x					x		
b. Encryption of data		x	x	x			x		
10. Platform features									
a. Ease of back-office integration		x				x			
b. Legal issues (patents)									
c. Licenses / Fees / TOS			x						
d. Distribution				x			x		
11. Future proof		x	x	x	x			x	x
12. Preference	x	x	x	x	x	x	x	x	X
13. Personalization	x	x	x	x	x	x	x	x	x

Table 6: Criteria discussed during interviews

4.3 Criteria relevance

The multi-perspective, from general-purpose mobile app development to consulting and in-house development, gave an insight to the many facets of mobile apps. The scope of the research has a focus on the choices made before the actual app development; this also sets boundaries for the criteria-set. Not all criteria mentioned by the experts are relevant for the choice of implementation platforms.

The criteria that were relevant to implementation platform choice are: user interface, responsiveness, offline access, device capabilities, sensors, multiplatform, portability, updatability, heterogeneity of devices, mobile device management, encryption, future proof, preference, and personalization.
Most of these can be directly linked to an advantage or disadvantage of an implementation platform as described in chapter 3; like platform independence, access to API's, user interface, update cycles, and performance.

The criteria that might be relevant are: sophistication, screensize, developer skillset, bring your own device, high-end, integration with back-office, and restrictions of content.
These criteria are relevant only in certain scenarios. For example integration with the back-office is only relevant if this back-office cannot provide a webservice.

The criteria that had no influence on the implementation platform choice were: special needs; tooling; outsourcing; business case; support; purpose; remote wipe, and legal issues.
Special needs, like ruggedized devices, is a hardware criteria and thus not applicable to the software choice. Some criteria were not related to the implementation platform choice, but by the operational and strategic vision of the commissioning organization; like business case; support; and purpose. Some criteria are only relevant after the choice is made, or during development: tooling and outsourcing. Legal issues over patents are considered to be between the vendors like Apple, Google, and Microsoft. Remote wipe overlaps with the Mobile Device Management (MDM).

Although budget is one of the most determining criteria, it is also tightly coupled with many of the other criteria. Another reason to exclude this criterion is that it will always get the highest priority and make decisions based on a single criterion instead of an objective view on the model.

The complete criteria and their relevance for the implementation platform choice can be found in Table 7.

Criteria	Description	Relevant for implementation?
User interface	The user interface of an app	YES
Responsiveness	How fast the interface reacts to input from the user	YES
Sophistication	The UI sophistications, scrolling and animations	Sometimes
Screensize	The optimum or minimal screensize	Sometimes
Offline access	The need for offline access (no internet)	YES
Device capabilities	Properties of the device	YES
Sensors	Internal sensors like: GPS and camara	YES
Special needs	Like ruggedized and certifications	NO
Multiplatform	The amount of platforms that the app runs on	YES
Portability	Re-use of code between platforms	YES
App development	Development aspects that belong to apps	
Developer	The skillset and availability of developers	Sometimes
Tooling	The used IDE and other tooling	NO
Outsource	Choice between inhouse or outsourcing	NO
Updatability	The ease of control on updating the app	YES
Business model	The business reasons behind an app	
Business case	The proposition, value model, ROI	NO
Budget	The available budget to cover the investment	YES
Support	The support or SLA arrangements	NO
Purpose	How does the app contribute to the user?	NO
Targetgroup	Who is the targetgroup of the app?	
Heterogeneity	Heterogeneity of devices in the targetgroup	YES
BYOD	Bring your own device? (internal app)	Sometimes
High-end	High-end or low-end market?	Sometimes
Security	Security aspects of the app	
MDM	Mobile Device Management	YES
Remote wipe	Option to remote wipe (is an option of MDM)	NO
Encryption	Secure and encrypted data? Datastorage	YES
Platform features	Features of the platform	
Integration	Integration with other systems like back-office	Sometimes
Legal issues	Issues over patents	NO
Licenses	Licenses, in-app-purchase shares	YES
Content	Restrictions on the content	Sometimes
Future proof	How will the app hold up in the near future	YES
Preference	Preference of company, executives, company policy, managers.	YES
Personalization	Is the content different for each user?	Yes

Table 7: Overview of the hierarchical ordered criteria

4.4 Final list of mobile app criteria

Based on the relevant criteria, described in section 4.3, a final transformation is made towards relevant quantifiable criteria that are suitable for the decision support tool. Each criterion should have a norm to which it can be tested. As an addition the inverse norm is also included to indicate the other end of the spectrum for the criterion. The final list of criteria, norms and inverse norms are shown in Table 8.

The final list of mobile app criteria is:

- **Connectivity** determines the level of required internet connectivity for the intended app. It is related to the criteria “Offline access”
- **Device sensors** determines if the app requires any sensors in the app
- **Market** determines if the app is required to be distributed through the vendor’s application store. This is related to the criteria “Platform features”
- **Platform amount** determines if the app should be cross-platform. This is linked with the criteria “Multiplatform”
- **Mechanisms** determines if certain native programming is required for the app, which cannot be built with web techniques. It is based on “Platform features” and “Device capabilities”
- **Content** determines what kind of interactivity is required for the content in the app. This is based on the criteria “Sophistication” and “Platform features”
- **Updatability** determines the lifecycle of the app, focusing on the amount of planned updates. This is based on “Updatability” and “Portability”
- **Personalization** determines the personalization factor of the app; this is based on the uniformity of content for the end-user
- **Responsiveness** determines the acceptable respond time of the user-interface and content loading
- **Native UI** determines if the app should have the native look and feel, or a cross-platform uniform user interface. This is based on “Sophistication” and “BYOD”
- **Screensize** determines the variety of screensizes that the app should support
- **Heterogeneity of devices** determines the amount of different devices the app should support
- **User expectation** determines if the end-users has a certain expectation from the app. This linked with “Preference” and “High-end”
- **Encrypted data** determines the required level of encryption of data for the app
- **Market fee** determines the monotization model and fees of the app
- **Regulations** determines if the app has content that can be restricted by vendors

The first six criteria (C1 – C6) should have a higher relevance then last ten (C7 – C16); they have the strongest relation with the choice for implementation platforms. Criteria C7 – C16 are used to further fine-tune the choice.

#	Criteria	Norm	Inverse Norm
C1	**Connectivity**	App is used without internet connectivity (offline access)	Device is always connected to internet
C2	**Device sensors**	App uses 1 or more device sensors	App uses no sensors
C3	**Market**	App must be in an application store	No explicit need to be in an application store
C4	**Platform amount**	App is to be deployed on more than 1 mobile OS	App only runs on a specific mobile OS
C5	**Mechanisms**	App requires native functionality to function correctly. It cannot be built with web technology (like advanced encryption / total control of sensors / etc)	App can run with web technology (HTML/CSS/JavaScript)
C6	**Content**	App shows only text and images	App has interactive presentation of content.
C7	**Updatability**	There are multiple versions planned on the roadmap of the app	App is final (bug fixing only)
C8	**Personalization**	App show personalized content for used (based on user and/or location)	App shows uniform content
C9	**Responsiveness**	App has to respond immediately on input	Response of more than 1 second is acceptable.
C10	**Native UI**	App requires native elements	App has simple interface (no requirements)
C11	**Screensize**	App has to support many screensizes (phone + tablet + desktop)	App supports limited amount of screensizes
C12	**Heterogeneity devices**	App supports many devices	App is made for limited amount of devices
C13	**User expectation**	User has high expectations, both visually and performance wise	User has normal expectations.
C14	**Encrypted data**	Storing encrypted data on device	Storing NO sensitive data on device
C15	**Market fee**	App will use in-app transactions	App does not use transactions
C16	**Regulations**	Impact of restrictions (Terms of Service) from OS vendors	App has explicit and/or age restricted content

Table 8: Overview final list of criteria, including the norm and inverse norm.

4.5 Criteria favorability for implementation platforms

With the mobile app criteria established, the favorability can added to each criterion. This indicates when a criterion norm is matched, whether this is favorable for this implementation platform (native / hybrid / web). Table 9 contains the criteria from chapter 4.4 and how favorable they are when the norm is matched. This is mainly based on matches with a (dis)advantage of the implementation platforms. For a recap please check Figure 10 on page 26.

Scale and legend for the table:

-1	Impossible, create a negative impact for this platform for the specified criteria
0	Not favorable or not supported / not possible
1	Neutral favorable or limited (functional) support
2	Very favorable or fully supported

	Criteria	Weight native	Weight Hybrid	Weight Web
C1	**Connectivity**	2	1	-1
C2	**Device Sensors**	2	1	-1
C3	**Market**	2	2	0
C4	**Platform Amount**	0	1	2
C5	**Mechanisms**	2	1	-1
C6	**Content**	0	1	2
C7	**Updatability**	0	1	2
C8	**Personalization**	2	1	1
C9	**Responsiveness**	2	1	0
C10	**Native UI**	2	0	0
C11	**Screensize**	0	2	2
C12	**Heterogeneity devices**	0	2	2
C13	**User expectations**	2	1	0
C14	**Encrypted data**	2	1	1
C15	**Market fee**	0	0	2
C16	**Regulations**	0	0	2

Table 9: Weights per criterion per implementation platform

A special note for criteria C1, C2, and C5: These list a negative impact for web apps (weight of -1), as these explicitly require support from native API's which is cannot be implemented in web apps.

For the criterion C1 Connectivity the app has to be used without internet connection (offline modus) which is very favorable for a native app. A hybrid app can be used offline but has a web app component that usually has to work online for data exchange. For a web app it is impossible, as these require a connection to the internet and favor to be connected the whole time.

For the criterion C2 Device Sensors the app uses 1 or more sensors. A native app has full access to sensors using the native API. A hybrid app has also access, albeit limited by the wrapper, to the

sensors. The web app is fully dependent on the sandbox of the browser; it cannot access sensors by itself.

For the criterion C3 Market the app must be downloadable in the application store. This is fully supported for native and hybrid apps. Yet web apps cannot be downloaded in the application store.

For the criterion C4 Platform Amount the app must be available on multiple mobile OS's. The time required to build a native solution for each native platform is very high, thus unfavorable. A web app is engineered to be platform-independent and thus very favorable for this criterion. A hybrid app has a small native component, but overall favorable for cross-platform apps.

For the criterion C5 Mechanisms the app requires certain native mechanisms that cannot be provided by web techniques. An example could be push notifications, GPU-instructions or heavy encryption on stored data. A native app has no problem with this. The wrapper in the hybrid app can be adjusted to support this native mechanism, but could take some time and has extra overhead. For a web app this is impossible.

For the criterion C6 Content show only text and images. This would imply a simple app that does not require a fully-fledged native app when a web app could already suffice. Layout, positioning and ease of development with text and images favor web apps (HTML) more than native apps.

For the criterion C7 Updatability the app has an ongoing development planned with multiple iterations of the app. A native app has to be updated through the distribution store of the vendor; it also requires a manual update on the smartphone. A hybrid app only needs to update through the application store if the wrapper (and/or web app depending on the app architecture), but can load content through the internet. A web app can be updated any time as they are not dependent on the application store.

For the criterion C8 Personalization the app has to show personalized content based on features on the smartphone like the GPS or user. When sensors or features of a phone are used, then a native app has some advantage. But nowadays even web-apps can use GPS or position a user based on IP or WIFI-access point.

For the criterion C9 Responsiveness the app has to respond immediately, for example when playing a game. When responsiveness is a key factor than a native implementation provides the most fluid performance. A web app relies on content downloaded from the internet, which has some lag. A hybrid app can put data and functions in the wrapper to get better performance than a web app, but will not be as quick as a native app.

For the criterion C10 Native UI the app requires the usage of native elements in the user interface to conform to the vendor's look and feel. Only the native app provides such elements, the hybrid and web app can only mimic the native elements.

For the criterion C11 Screensize the app has to support multiple screensizes, like phone-, tablet- and/or desktop-screensizes. HTML is ideally suited to provide a responsive design based on

screensize. The native app has to use native elements which require much more time to create the necessary views. A hybrid app uses the same rendering engine as a web app, thus both are very favorable for this.

For the criterion C12 Heterogeneity devices the app has to support many devices, for example budget to high-end Android smartphones, where not only screensize can vary but also the used hardware. A hybrid- and web app are ideal for these situations as the browser will run in virtually any environment. A native app can require more performance then the hardware allows, also it can require specific knowledge to link the performance to specific hardware profiles.

For the criterion C13 User expectation the app has to live up to high user expectations both visually and performance wise. This favors native implementation, especially for the performance. A web app and hybrid app cannot match a fluently complex interactive user interface combined with performance.

For the criterion C14 Encrypted data the app has to store encrypted data on the device. The native app can store and encrypt data on the smartphone using the API. A hybrid- and web app have to rely on localstorage with encryption through JavaScript. A hybrid can extend the wrapper for saving secure encrypted data

For the criterion C15 Market fee the app uses in-app transactions. For a native and hybrid app this results in mandatory use of the vendor's application store, which usually includes a fee of around 30% of the total amount. This is not favorable compared to a web app where such restrictions are not applicable

For the criterion C16 Regulations indicates the impact that the Terms of Service (ToS) of the various vendors' can have on the app. This is not applicable to a web app, and thus favorable. The native and hybrid app have to comply with the terms and conditions that are set by the vendor, which can restrict certain aspects of the app like strict guidelines for interfaces.

5 DECISION SUPPORT SYSTEMS

A decision support system (DSS) is a computer program providing both problem-solving and communications capabilities for semi-structured problems(McLeod and Schell 2001). This research will employ DSS methods to develop a theory to link the implementation platforms (chapter 3) with the final list of criteria (chapter 4.4) and their favorability between criteria and platform (chapter 4.5). Based on DSS theory prototypes will be developed.

DSS very broad field in information systems includes many different techniques that support the user to reach decisions. These techniques include OLAP, data warehousing, data mining, model building, expert systems, neural networks and intelligent agents(Alter 2004). Although the term DSS was coined in the 1970s, it is also includes umbrella terms like business intelligence (BI), decision support applications (Alter 2004), personal decision support systems, executive information systems and group support systems (Arnott and Pervan 2005).

Two research questions are related the topic of DSS: “What decision support systems are suitable for the linking criteria with the implementation platforms?” and “Which decision systems can be transformed into a prototype decision tool?”. A DSS can provide a solid method to transform the collected data into a tool that can be used in practice.

5.1 Decision support system theory

In four decades there is still no consensus over the exact definition of a DSS, although they should have the following characteristics: It is aimed at the less well structured, under-specified problem that managers typically face. It attempts to combine the use of models and/or analytical techniques with traditional data access and retrieval functions. The DSS specifically focuses on the feature that makes them easy to use by non-computer people in an interactive mode. And the last is the emphasis on flexibility and adaptability to accommodate changes in the environ and decision-making approach of the user (Er 1988).

The functional components within an DSS should contain at least the following three: management of the dialog between user and program; management of the data; and the management of the models (Ginzberg and Stohr 1982).
These definitions of characteristics and functional components make DSS a very broad field. A general classification for the different types of DSS consists of data-driven systems; model-driven systems; knowledge-based systems; communication-based systems; and document-driven systems (Alter 2004).

This thesis seeks the most optimal alternative implementation platform based on ranking multiple criteria. This type of decision support system (DSS) is classified as a “preference determination system” (Finlay 1994) or “multi-criteria decision making”. These kind of DSS fall into the knowledge-based systems. A requirement for a preference ranking decision support system is that there is a clear set of well-defined alternatives (Finlay 1994).

There are two types of models commonly used in decision analysis: consequence models and value models (Keeney and von Winterfeldt 2009). A consequence model incorporates facts, judgments,

and uncertainties inherent with decision problems, while value models incorporate the values or value tradeoffs and risk tolerances to evaluate consequences. In the decision analysis field, the value model are sometimes referred as utility functions, value functions or preference functions (Keeney and von Winterfeldt 2009).

A quick scan in literature suggests two potential candidates that are associated with a preference function (Belton 1986; Finlay 1994; Zanakis, Solomon et al. 1998): a (simple) multi-attribute value function, and analytical hierarchy process (AHP).

5.2 Multi-attribute value function

A widely used method is the simple multi-attribute value function (MAV), which is often based on simple additive weighted value function (SAW)(Belton 1986).

The general formula is: $V_i = \sum_j w_j x_{ij}$

Where V_i is the total value of alternative i.
w_j is the weight of criterion j to reflect its relative importance compared to other criteria
x_{ij} is the score of alternative i on criterion j

For the prototype the alternatives will consists of the native, hybrid, and web platform. The criteria (j) are C1 Connectivity, C2 Device sensors, C3, Marktet … C16 regulations. The score of each criteria (x_{ij}) is reflected in the favorability towards a platform.
The relative weight of the criterion can to be determined by the user, although the suggestion is to make C1 until C6 at least twice as important as C7 - C16. The latter are used for further fine-tuning in the choice between implementation platforms.

A simplified example, given the following values in a decision matrix with 4 criteria

Criteria (j)	Weight (w_j)	Weight Normalized	Native	Hybrid	Web
C1: Connectivity	3	0,2	2	1	-1
C2: Device sensors	2	0,13	2	1	-1
C3: Application Store	4	0,27	2	2	0
C4: Multiplatform	6	0,4	0	0	2

The score for **native**: 0,2 * 2 + 0,13 * 2 + 0,27 * 2 + 0 * 0,4 = 1,2
The score for **hybrid**: 0,2 * 1 + 0,13 * 1 + 0,27 * 2 + 0 * 0,4 = 0,87
The score for **web**: 0,2 * -1 + 0,13 * -1 + 0,27 * 0 + 2 * 0,4 = 0,47

The preferred approach would be native, followed by the hybrid approach. The web approach has the disadvantage that C1 Connectivity (offline access) and usage of C2 Device Sensors were requirements, which brought the total score down.

The multi-attribute value function has simple calculations, which rely on the correct values and weights from the input.

5.3 Analytical hierarchy process

Analytical hierarchy process (AHP) is an multi-criteria decision making approach where the factors are arranged in a hierarchic structure(Saaty 1980; Saaty 1990). It is in essence also based on a simple additive function, just like a multi-attribute value function (Belton 1986), but without the explicit definition of the scores.

The AHP model uses three steps:

1. Structure the problem as a hierarchy, see also Figure 12;
2. The elicitation of pairwise comparison judgments;
3. Establish the composite or global priorities of each of the options (Saaty 1990).

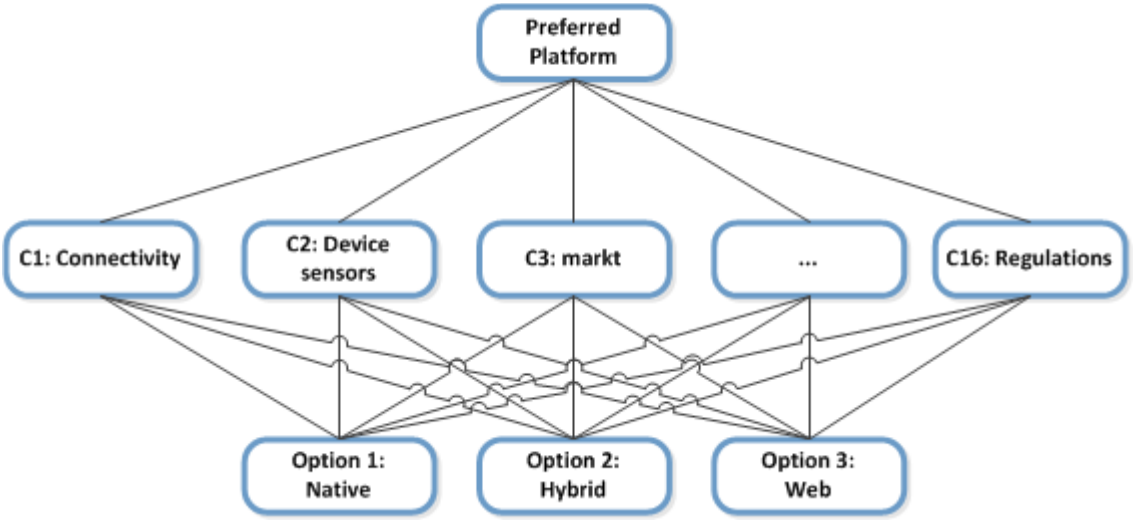


Figure 12: AHP structure for implementation platform based on app criteria

The method relies on making pairwise comparison between the criteria (C1 – C16). Each criteria is compared with another indicating which is more important on a scale from 1 (equal important) to 9 (extreme importance). This scale only used odd numbers: 1, 3, 5, 7, and 9.(Saaty 1990). It is still possible to give each criterion a weight, in our case to stress the importance of criteria C1 until C6.

Each result of a comparison between criteria (scale 1 - 9) is put in a square matrix, which calculates the priority vector (eigenvector). The last step is to calculate for each driver the local priorities, which is again done by calculating the priority vector but then on the options per driver.

The main problem with this approach is the necessary pair-wise comparisons. If all 16 criteria are used, then a 120 compares between the criteria needs to take place. This is not very feasible.

The amount of pairwise comparison is exponential: $(n^2 – n) / 2$

# Criteria	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
# Pair-wise questions	0	1	3	6	10	15	21	28	36	45	55	66	78	91	105	120

5.4 Prototype

Based on the previously described decision support system types, two prototypes were developed to transform criteria into a decision. The screenshots of the prototypes in Excel can be found in APPENDIX F: PROTOTYPE SAW (page 103) and in APPENDIX G: PROTOTYPE AHP (page 104).

5.4.1 Multi-attribute value function

Based on the final list of criteria and indication described in section 4.4 a weighted decision matrix was created, based on a modified multi-attribute value function. Each of criteria is already defined, including the weights. In the prototype the same weights are applied for each criterion, which spans from -1 to 2. The prototype does add two columns:

- **Enabled**: This column determines if the criterion should be added to the total. If the client conforms to the norm, then put this on 1. If the client conforms to the inverse norm, then put this on -1. When there is no mention of this criterion, just leave it on 0.
- **Criteria weight**: The relative weight of the criterion, compared to other criteria. Default value is 1, except for C1-C6 these are considered major criteria and have the weight of 2 or 3.

Both columns can be freely adjusted by the user.

To calculate the total score per platform the sum of all criteria, and each criterion as the score of: Criterion Enabled * Criterion Weight * Criterion Platform Score.

$Platform\ score = \sum(C_n e * C_n w * C_n p)$ which is roughly the same as $V_i = \sum_j w_j x_{ij}$

The Excel sheet then checks which platform has the maximum score, as that one is the most preferable. An example can be found in Figure 13.

5.4.2 AHP

The prototype of AHP requires more work, as first the criteria should be prioritized. This is done by making use of freely AHP template to determines the eigenvector, criteria weights and handles the pairwise comparison (Goepel 2012).

This is then subsequently introduced into a similar prototype like the multi-attribute value function. Only the scale has been adjusted to 1 to 9, as this is the standard for AHP.

The platform with the highest score will be most favorable.

			Fill these columns		
		Criteria	**Enabled**	**Criteria weight**	Norm
Major criteria	**C1**	**Connectivity**	1	2	App is used without internet connectivity (offline access)
	C2	**Device Sensors**	1	2	App uses 1 or more device sensors
	C3	**Application store**	0	2	App must be in a market/store
	C4	**Platform Amount**	1	4	App is to be deployed on more than 1 mobile OS
	C5	**Mechanisms**	1	2	App requires native functionality to function correctly
	C6	**Content**	1	2	App shows only text and images
Criteria	**C7**	**Updatability**	1	1	There are multiple versions planned for the app
	C8	**Personalization**	0	1	App show personalized content for used (based on user and/or location)
	C9	**Responsiveness**	0	1	App has to respond immediately on input
	C10	**Native UI**	1	1	App requires native elements
	C11	**Screensize**	1	1	App has to support many screensizes (phone + tablet + desktop)
	C12	**Heterogeneity devices**	0	1	App supports many devices (like for BYOD)
	C13	**User expectations**	0	1	User has high expectations, both visually and performance wise
	C14	**Encrypted data**	0	1	Storing encrypted data on device
	C15	**Market fee**	0	1	App will use in-app transactions
	C16	**Regulations**	0	1	Impact of restrictions (Terms of Service) from OS vendors
Outcome			Native	Hybrid	Web
		Platform Score	14	15	10
		Preferred platform:	**Hybrid**		
Legend & Explanation		Enabled:	Did the customer mention this criteria? 0 = No, 1 = Yes		
		Criteria weight:	How strong is the preference for this criteria. Scale is from 1 till 3 (higher = stronger)		
		Major criteria:	These criteria have a high impact on the outcome. Preference is set to 2 or 3		
		Normal criteria:	These criteria have a normal impact on the outcome. Preference is set to 1		
		Platform weights:	The scale for weight are from -1 to 2.		
			-1	Not possible, create a negative impact for this platform for the specified criteria	
			0	Not favorable or no support	
			1	Neutral favorable or limited (functional) support	
			2	Very favorable or fully supported	

Figure 13: Prototype example multi-attribute value function

6 VALIDATION

The validation consists of three parts: validation experiment with a client; validation interviews with experts; and a tool analysis. The first is to validate the usability of the tool in practice. The second part is to validate the criteria and the model. The last one is to give insight in the answers of the tool.

6.1 Validation experiment

The experiment is based on several components: a short qualitative interview, model testing, and reflection on the results of the session. This is based on literature from semi-structured interviews as well as case study research (Eisenhardt 1989; Hevner, March et al. 2004; Yin 2009; Thomas 2011). A complete protocol can be found at APPENDIX H: VALIDATION EXPERIMENT PROTOCOL on page 105. The summary of the experiment can be found in APPENDIX I: VALIDATION EXPERIMENT SUMMARY on page 110.

6.1.1 Goal

The goal of the interactive session is to validate two prototype models. The research provided several mobile app criteria based on literature and expert interviews. The impact of these criteria has been weighted in relation to the mobile implementation platforms, and molded into decision support tools. The experiment is to validate the two prototype tools on the following three points:

- **Practical applicability**: The extent to which the models are applicable in practice
- **Correctness of the models**: Does the sessions deliver new insights to improve the models? But also to which extent the results match the expected results
- **Preference of a model**: A personal preference by the participant for a model

6.1.2 Setup

The experiment is an interactive session with several components. It is qualitative in nature, which creates flexibility to improvise during the session and deviate from the pre-set script.

The four components are:

1. **Introduction**: An introduction of myself, the thesis, and goal of the session. Also the time for a disclaimer on the results
2. **Interview**: A short interview that is based on open questions asked from a business perspective. It retrieves the criteria on a qualitative manner. Each question is formulated in a way that it could answer if 1 or more criteria are used for the mobile app that the client wants to realize. After the interview it is necessary to discuss the found criteria, to confirm understanding of the requirements on the app.
3. **Model testing**: The found criteria are entered in the Excel prototypes. The prototypes are based on a simple additive weighting (SAW) and Analytical hierarchy process (AHP).
4. **Reflection**: Afterwards the results are discussed, but also the suggestions for improvement. The last question is which model has the preference of the client.

6.1.3 Interview

The interview is based on the same constructs as the semi-structured interviews that were conducted with the mobile experts, see also APPENDIX C: INTERVIEW PROTOCOL on page 62 and APPENDIX D: SUMMARIES OF INTERVIEWS on page 66. The reason for the qualitative approach is that the participant can express him/herself in his/her own words and terms. The interviewer needs

to be alert if these match any of the criteria, and adjust or further inquire when it is not clear what the participant means. The interview will have a maximum duration of 40 minutes.

There are four leading questions:

1. What is your experience with mobile applications and devices?
2. What is the purpose of the mobile application?
3. What are the (business) drivers behind the mobile app?
4. Are there (technical) requirements for the mobile app?

The first question is to get a feeling for the understanding of mobile apps that the participant has. The last three are mainly to determine what kind of app the customer has in mind. Each question has multiple follow-up question, each with indicators of which criteria to expect. After finishing the interview, the interviewer will provide the participant with a list of found criteria. Together they go over the list, and confirm by each criterion the participation in the models. By doing so the participant gains insight into the criteria that are sought; and the interview can confirm if he/she understood the requirements of the app.

6.1.4 Model testing and reflection

There are two models that are tested in the experiment. Both prototypes of the tools are built in Excel, which provided immediate feedback as the criteria are entered into the Excel-sheets. The first model is based on a simple additive weighting (SAW) approach, while the second is based on Analytical hierarchy process (AHP). See also APPENDIX F: PROTOTYPE SAW on page 92 and APPENDIX G: PROTOTYPE AHP on page 104.

The reflection is the opportunity to have a critical look at how the goals are reached. The goals are defined to which extend the models proved to be applicable in a practical environment, and where improvements may lie. The results of the models are discussed and further clarified if needed.

A final question is which model has the preference of the participant, and why.

6.1.5 Results from experiment

This experiment has been conducted at a client of Deloitte, at an organization that is involved in entrepreneurship. The organization has decided to take advantage of the mobile communication channel and is planning to create an app. The app should have functionality similar to that of the current website. For example: looking up businesses, links to LinkedIn, finding nearest location of offices, etc. In the future new features and functionality should be added incrementally.

The interview was relatively short, where the participant was mainly describing the mobile app from her perspective. During the interview we noted which criteria were mentioned, so after the interview these could be double checked.

The list of criteria can be found in Table 10. Note that the names of the criteria have been updated to the current ones, but still contains two obsolete criteria: “Cost” and “User coverage”. This experiment tested both prototypes.

The result of the first prototype, based on the simple additive weighting, was: web (21), hybrid (17), and native (6). This clearly favors the web approach, which can be explained by the fact that the client only listed non-native smartphone features.
The result of the second prototype, based on Analytical hierarchy process, was: web (42%), hybrid (34%), and native (25%). This also favored web as best platform.

In the reflection of the experiment, it became clear the first model was simple in usage. The client could see what impact a certain criterion had on the total score. The second prototype (AHP-based) was more complicated and knew more limitations (maximum of 8 criteria); the in-depth categorization of criteria is not essential or necessary in this phase of application.
The experiment proved that the tool can be used in practice.

		Criteria	Answer Interview	Applied inWDM	Applied in AHP?	AHP weight
Major criteria	C1	Connectivity	No	No	No	-
	C2	Device sensors	No	No	No	-
	C3	Application store	Yes	Yes	Yes	8%
	C4	Platform amount	Yes	Yes	Yes	17%
	C5	Mechanisms	No	No	No	-
	C6	Content	Yes	Yes	Yes	22%
		Criteria	**Answer Interview**	**Applied in WDM?**	**Applied in AHP?**	**AHP weight**
Normal criteria	C7	Cost[17]	Yes	Yes	Yes	6%
	C8	Updatability	Yes	Yes	Yes	12%
	C9	Personalization	No	No	No	-
	C10	Responsiveness	No	No	No	-
	C11	Native UI	-	No	No	-
	C12	Screensize	Yes	Yes	No	-
	C13	Heterogeneity devices	Yes	Yes	No	-
	C14	User expectations	-	No	No	-
	C15	User coverage[18]	Yes	Yes	Yes	21%
	C16	Encrypted data	Yes	Yes	Yes	14%
	C17	Market fee	No	No	No	-
	C18	Regulations	No	No	No	-

Table 10: Results from validation experiment

[17] Costs was later removed from the list of criteria
[18] User coverage was also removed, as this overlapped to much with other criteria.

6.2 Validation sessions

The criteria and the tool were also presented to mobile experts to validate the criteria and relations. For the complete protocol see APPENDIX J: VALIDATION SESSION PROTOCOL on page 115. For the complete summaries see APPENDIX K: VALIDATION SESSION SUMMARIES on page 122.

6.2.1 Setup

The session has four parts:

1. **Introduction**: Introduction of the research, starting with a general overview of the researched topics. It also explains the chosen implementation platforms: native, web, and hybrid.
2. **Criteria**: The process of selecting criteria for mobile apps is discussed, where a total list is presented with all criteria. As not all criteria are necessary affecting the implementation platform, only those that are relevant are included to a final list. Then the weights per platform per criteria are introduced, which is the base of the tool.
3. **Tool**: When all criteria are clear, a test case will be done in the tool. This will confirm or disprove the validity of the tool. Furthermore it gives insight in the workings and practical application of the tool.
4. **Evaluation**: The last part of the review is the reflection of the criteria and tool. Any remarks and comments are reviewed. This is also where the practical use of the tool is assessed.

6.2.2 Results

Remarks and comments of the first expert were:

- **Connectivity (C1)**: Defined too strong, there are multiple ways (full online, full offline, and hybrid) that an app can use the connection for online and offline activities. The question (or norm) could better the rephrased to: can the app be used without internet?
- **Content (C6) and Personalization (C8)**: Very important criteria which shows how the app will interact with the user.
- **Responsiveness (C9)**: Is a non-criterion. An app should always be as fast as possible, whenever the app should retrieve something from a server then a user expects to wait for it. Perhaps this is a combination of the criteria “connectivity” (C1) and “User expectations”(C13)
- **Native UI (C10)**: Include the question if phone-buttons are used (like the menu-button from android).
- **Screensize (C11) and heterogeneity of devices (C12):** A good thing to keep these separated. Screensize support goes beyond the smartphone platform, while heterogeneity of devices is about the diverse smartphones.
- **Encrypted data (C14)**: Storing sensitive data on the device itself is as strong as the used implementation option (and thus up to the developer). But native app can make use of the standard implementation by the vendor, or choose something more advanced. If sensitive data is stored on servers and downloaded through a secure connection than this is also very safe. Perhaps adjust the weights to 1 : 0 : 1 (Native : Hybrid : Web).

When reflecting upon the criteria and the tool, he definitely sees the practical use. The first expert states that all criteria that really matter for an app-implementation-choice are listed. This list can be used when asking a client/customer what kind of app they have in mind. He is also positive about the weighted decision matrix tool, where the linear connection between criteria and outcome can be easily observed. This gives more insight in the dynamics of the criteria influence.

A remark is that the extraction of the criteria from a client depends heavily on the skill of the interviewer (usually a consultant or advisor). Another remark is that the criteria should be updated periodically, because smartphones and new mobile OS versions come out on a regular base. A final comment is that some of the criteria are at best an estimated guess, which is also a fairly good reason that the tool should be adaptable and simple.

Remarks and comments of the second expert were:

- **Market (C3)**: The market is more than just visibility, it is also about findability. Being in the market is not enough if you cannot be found by your users.
- **Platform amount (C4)**: Perhaps rename this criterion to “market reach” as this can be seen from a user target perspective; instead of relying on mobile OS alone.
- **Content (C6)**: The formulation could be improved, towards passive versus (inter)active app. Or to one-way communication versus two-way communication for content. This depends on the service and organization that offers the app.
- **Updatability (C7)**: Include a roadmap and/or lifecycle in the description and/or norm. This is much clearer than “versions”.
- **Personalization (C8)**: A very important criterion, as the smartphone is a personal device. His opinion is that this should be a focus from the start for any app.
- **Responsiveness (C9)**: Also a very relevant aspect of any app, yet difficult to formulate a norm. It really depends on the context of the app and the user.
- **Native UI (C10)**: Users are used to User Interface (UI) patterns that belong to each mobile OS. Examples are look and feel of buttons; usage of (physical) buttons; screen position of elements (like menus, back-button and tab-bars).
- **Screensize (C11)**: Next to screensizes for smartphones and tablets, it could also be another factor that is considered: screen-rotation. So landscape or portrait modus should be included when considering screensizes.
- **Heterogeneity of devices (C12)**: Make explicit that this is about the amount of phones (hardware differences on a platform). An example for this is the amount of Android phones; they all run Android but vary in CPU, RAM, storage space, GPU etc.
- **User expectations (C13)**: Expectations differ per market segment. There is a difference between the consumers and the enterprise market. Consumers are more focused on design and branding, while the enterprise market is concentrating on functionality and reliability.

Expert 2 misses a criterion that covers development, which in the initial criteria list covered the following topics: skill set required by developer; tooling and IDE; in-house versus outsourcing; open source; and updatability. None of these have an impact on the choice for the implementation platform, except updatability which is included as a criterion (C8). The skill set required by developer and tooling are more or less covered in the platform amount (C4).

Another criterion that was missing is cost. This criterion has been intentionally removed as the tool should give the best implementation platform based on requirements by the client. The total costs (licenses, development costs, support, etc) are a result of this choice. It is better to keep this out of the scope of the tool, or reflect on this afterwards when discussing the result of the tool.
The last remark on the criteria was that he missed the organizational criteria like: bringing in the required skills for the app (development, management, support); and the continuity of an app. These are very valid, and only partially covered in the current criteria. A counterargument is that most of these are far beyond the choice for an implementation platform, and thus out of the scope of the tool. Nevertheless they are criteria that could be more visible or incorporated into the tool.

The second expert thinks that the tool is useful for clients that are in the scoping phase for an app. Although it is designed on an application level, not an organizational level where multiple mobile apps should be considered and maintained.

6.3 Tool analysis

For further analysis of the tool, a java program was developed to calculate all possible outcomes of the prototype based on the simple additive weighting value function. The code can be found in APPENDIX L: TOOL ANALYSIS CODE , or on Github[19].

The program has a compact representation of the internal weighted decision matrix, in the form of a 16 x 4 matrix. Each row of the matrix is a criterion with the relative criterion weight and the weight per platform (native, hybrid, and web). The output of the program shows how many times an implementation platform is favored.

In short the program generates all possible sequences of enabled criteria. Internally it generates strings of 16 characters is an abstraction for enabled or disabled criteria. For example the string “0000 0000 0000 0000” has no enabled criteria; while “1100 0010 0100 0001” has C1, C2, C7, C10, and C16 enabled. For each sequence of criteria the total outcome and preference is calculated; this is then stored in a global variable. There are four valid outcomes: favor native; favor hybrid; favor web; no favor. No favor means there is a tie in score.

An optional feature is to set a minimum amount of criteria that should be enabled. On default a minimum threshold of five is handled; as it is unrealistic that a client would have less than five requirements for an app.

As an extra feature the inverse norm can also be selected, this mean not a binary choice (enabled or disabled of a criterion) but also a ternary choice (enabled, disabled, inversed). This is possible as for each criterion there is a norm, but also an inverse norm formulated. When the requirement meets the inverse norm a “-1” is entered into the linear equation.

[19] https://gist.github.com/c8e3699a46c629b5cbdb/efc9afd6a130fa491c090ce7233073457b36f62d

Note that the relative weights are in their default settings but these can be easily adjusted in the tool. It would increase the complexity exponentially, and also the result would be very hard to interpret.

6.3.1 Results

The output of the program can be seen in Table 11 and Table 12. The table rows show the total of criteria and the total amount of combinations. In the in the binary variant this is 2^16 which is 65.536; in the ternary variant this is 3^16 which is 43.046.721. Then the amount of combinations that is favorable toward each platform is listed. The second column expresses this number in a percentage. This is also done when the constraint of a minimum of 5 enabled criteria ("Min 5C's"), although this has only a minor impact.

In the model where the criterion only complies with the norm or not (binary choice), the model polarizes towards the native implementation with 59%. Striking are the small amounts for the rest of the platforms: hybrid (11%) and web (17%). Although this can be explained as the majority of norms tend to favor native application developments. When the constraint of at least 5 criteria is maintained then 2517 combination drop off; which corresponds to roughly 4% of the total. The impact of enforcing this only makes a small difference.

In the model where the norm and inverse norm can be chosen (ternary choice), it has a more balanced outcome. This was also expected, as the binary was more polarized. The amount for hybrid stays more or less the same with 10%, while native and web each have around 40%. The enforcement of a minimum of 5 criteria only excludes 34113 combinations which is only 0.08%, this makes no difference for the result.

These results have been visualized in Figure 14.

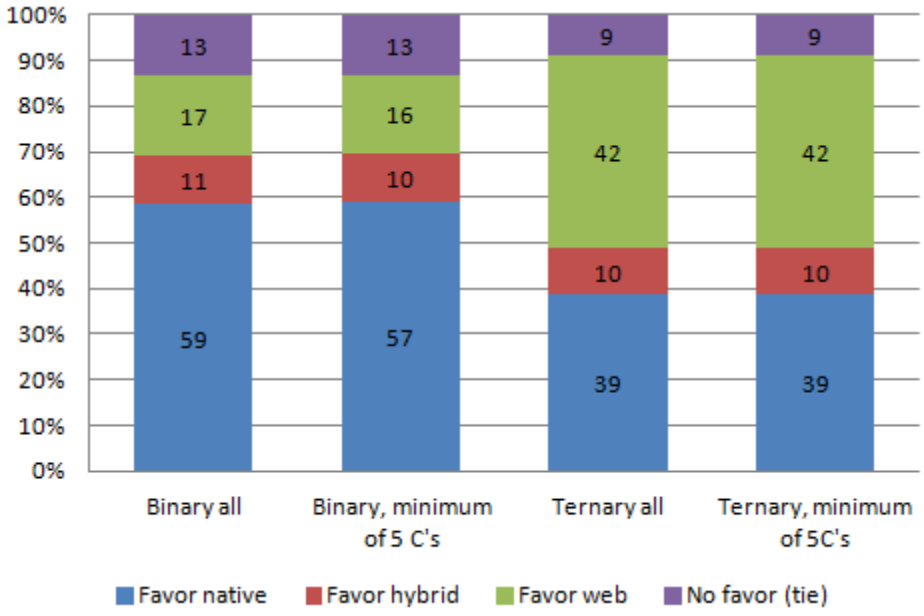


Figure 14: All possible outcomes of the tool, displayed in percentages

BINARY	No minimum of criteria	% of total		Minimum of 5 criteria enabled	% of 65536	Difference with no mimimum
Favor native	38360	59		37150	57	1210
Favor hybrid	6960	11		6731	10	229
Favor web	11396	17		10729	16	667
No favor (tie)	8820	13		8409	13	411
Total:	65536	100		63019	96	2517

Table 11: Output for the binary choice (norm or disabled)

TERNARY	No minimum of criteria	% of total		Minimum of 5 criteria enabled	% of total	Difference with no mimimum
Favor native	16635006	39		16622515	39	12491
Favor hybrid	4361872	10		4358913	10	2959
Favor web	18145746	42		18131918	42	13828
No favor (tie)	3904097	9		3899262	9	4835
Total:	43046721	100		43012608	100	34113

Table 12: Output for the ternary choice (norm, disabled, or inversed norm)

6.4 Results of validation

The experiment has shown that both prototypes could work in a practical environment. It also showed room for improvement:

- **Overlap of criteria**: Some criteria demonstrated great overlap, like heterogeneity, screensize support and user coverage. All three are about the range of devices the app will support, and could create a bias towards favoring a single platform.
- **Criterion names**: When comparing the criteria in pairs for AHP, it is very important that the names easy to compare. An example was comparing “info channel” with “user coverage”. These comparisons also made it more clear which criteria overlap
- **Criterion norms**: Instead of having an question, it should be norm that can be measured or explained in an easy way
- **Weights of criteria**: It seems that some criteria need a bit of tweaking in the scale

7 CONCLUSION

The research questions presented in chapter 2.3 (page 18) have been leading this research through the previous four chapters. First all sub questions will be answered, followed by answering the main research questions.

Q1: What are the implementation platforms in the current mobile ecosystem?

As long as there are multiple smartphone platforms, like iOS and Android, then there is the need to have mobile applications that run on all these platforms. There are several implementation styles that can be used to achieve these cross-platform mobile apps: native, hybrid, and web.

The native platform utilizes the mobile operating system SDK and API's which provides the most control and performance, and also utilizes the distribution channel of the vendor. The disadvantage is that the mobile app has to be developed for each mobile OS, which is essentially developing the same app multiple times, increasing the time and resources involved.

A web app is a mobile-optimized website which requires the browser and connection to the internet for access. The platform independent web app will render the same on the smartphone, using web techniques like HTML, CSS, and JavaScript. The downside is that a web app cannot use any native smartphone functionality, only use features that the browser provides. Also the web app cannot be downloaded through the normal application store from the vendor, as it is a website.

The hybrid app combines the web and native approach by using a native wrapper and does the rendering in a chromeless browser. The core consists of a web app using web techniques, but can use native functionalities through the wrapper which facilitates programmable bindings for native functions. The advantage is that the app can be downloaded from the application stores and contain platform independent elements.

The implementation platforms are discussed in more detail in chapter 3.

Q2: What are key criteria for mobile applications according to mobile experts?

By interviewing mobile experts and using qualitative data processing techniques the research has resulted in a total of 16 mobile app criteria. Each criterion has a weight to indicate the impact or rank in respect to other criteria. The six most important criteria, with their norm, are:

1. Connectivity: The app operates without internet connectivity
2. Device sensors: The app uses 1 or more device sensors
3. Market: The app requires distribution through an application store
4. Platform amount: The app is deployed to multiple mobile operating systems
5. Mechanisms: The app requires native functionality to function correctly, which cannot be replaced by web techniques.
6. Content: The app shows simple content like text and images.

The other 10 criteria are: updatability; personalization; responsiveness; native UI; screensize; heterogeneity of devices; user expectation; encrypted data; market fee; and regulations. Chapter 4 presents all 16 criteria, and discussed the interviewing and data processing techniques used to uncover these criteria.

Q3: How do the mobile criteria relate to the mobile implementation platforms?

The criteria extracted from the interviews all have norms, which can be compared to the (dis)advantages of each implementation platform. Each criterion has a score per implementation

platform, which indicates how much an implementation platform is preferred. The scale goes from not possible (like native mechanism on web apps), to no support for this feature on this platform, to neutral or limited support, to fully supported and/or very favorable for the implementation platform.
Native and web implementation tend to polarize to the extreme; it either takes full advantage for the criterion or is not favorable at all. The hybrid approach stays in between, tagging along on the advantages from the native or web implementation.

For the six key criteria the relation is as follows. Connectivity (1) favors the native implementation platform due to the fact that offline access is not possible for a web app, and limited for hybrid. The most controls for access to device sensors (2) is through the native API; web apps have no or limited access; hybrid have usually generalized interfaces. Distribution (3) through the application store is possible for native and hybrid app, yet impossible for web apps. The more mobile platforms (4) the app has to support, the better it suited for a web app. For required native mechanisms (5), the only options are native and hybrid implementation platforms, where native would be preferred above hybrid. Apps that only show simple content (6) like text and images can be perfectly rendered by a web app, without requiring a native app.
The full list with all criteria relations towards the implementation platforms is shown in chapter 4.5.

Q4: What decision support system methods are suited for linking mobile app criteria with the implementation platforms?
Decision support systems that determine the most optimal alternative based on multiple criteria; are called multi-criteria decision making systems. Multiple methods and frameworks are available for these types of decision systems. The two most popular are multi-attribute value functions and analytical hierarchy process. The multi-criteria value function is a simple method where each criterion has a weight and the highest scoring alternative is the best. Analytical hierarchy process also includes priority of the criteria, which consumes more time but also gives a finer granularity than the multi-criteria value function.
The core for both methods is the matrix of relations between mobile app criteria and mobile implementation platforms; which is derived from the interviews and (dis)advantages of the implementation platforms. Chapter 5 has a more in-depth view into these two decision support systems.

Q5: Which decision support system can be transformed into a prototype decision tool?
The research provided two prototypes. The first is based on the multi-criteria value function, where direct causal relations are reflected in the model. The simplicity makes the model understandable for the client, where enabling a criterion or changing weights has a direct impact on the outcome. The second prototype is based on analytical hierarchy process (AHP) approach, which requires more time due to the pairwise comparisons. Although the AHP prototype worked equally well it is less transparent but has finer granularity. In chapter 5.4 both prototypes are discussed in more detail.

Q6: What is an appropriate way to validate the design?
The research employed multiple ways to validate the prototype. The first is a validation experiment, the second is an interactive session with mobile experts, and the third is to analyze all possible outcomes of the tool (sensitivity analysis). The validation experiment at a customer of Deloitte

proved to be very valuable in determining the practical impact. It showed that both prototypes can provide an advice towards implementation platforms based on the (business)criteria. The sessions with experts validated the findings that it is very practical; although it can only be used in the planning phase of app development. According to the experts the list of criteria is quite complete and is applicable with customers. The sensitivity analysis revealed that the prototype tool tends to favor a native implementation; this is a direct result from the criteria descriptions and their norms.
The full validation experiment, reviews, and sensitivity analysis are found in chapter 6.

Which leads up to the main research question:
How to transform mobile application criteria into decisions for mobile implementation platforms using a decision support system?
Using a simple linear model provides the insight and transparency, on which criteria and choices favor a certain platform. The developed decision support system was designed to provide this insight to make it practically applicable. The DSS uses a multi-criteria value function that calculates the score adding the weight of each criterion per implementation platform. It is imperative that the criteria and their weight per implementation should be carefully chosen and validated.
To determine the criteria, a qualitative analysis has been performed after a series of semi-structured interviews with mobile experts. By hierarchically structuring the topics for mobile apps a total of 16 criteria were synthesized. The last step is to relate each criterion to the implementation platforms (native, web, hybrid).

The main contribution is determining the set of criteria and the corresponding (relative) weights towards the finite set of implementation platforms. This contribution has been presented in chapter 4.

8 DISCUSSION

This research provides useful guidelines and a baseline for decision support system. Yet there are some aspects that need further research, or should be further investigated.

The first is the limited amount of interviews for data collection for determining the mobile app criteria. Although I am confident in the produced criteria, it can still be biased by the selected group of interviewees. When selecting participants for the research, I did take into account that several perspectives were represented. These included: app developers, consultants, and app managers.

Another point is that the tool could "age" relatively quickly, making it less usable. This has to do with the rapid evolving mobile technology and new smartphones that are released each year. Also new releases of mobile OS's are frequently and are included with new options and features. This requires a periodical update to the criteria and weights per implementation platforms, to keep it relevant and up-to-date.

The used scale in the prototypes for weights is relatively small, from -1 to 3. This gives no room for fine granularity or tweaking of the weights. Also the total scores between alternatives outcomes can be perceived as being small. An increase in scale could potentially tackle both problems. The main reason that this has not been done is because the primary focus of the tool was to validate the models and criteria first.

Some argue(Slashdot 2011) that native apps and web apps are not mutual exclusive. Native apps have always been able to use a WebView in order to render web apps. This is indeed true, but this is not the point of a web app. A mobile-optimized site is not optimized for a chromeless browser, but indeed for the normal regular mobile browser. The hybrid approach, in which the native app uses a WebView for the presentation layer requires that the web app is designed for this, including wrappers for approaching native sensors. To wrap it up, a native app is not exclusive towards the web app; but a web app is excluded from native functions if no wrapper is used.

8.1 Future research

In an empirical study the model and tool could be more extensively studied. A questionnaire, instead of a semi-structured interview, has a better certainty interval to verify the listed criteria. A larger pool of participants also means that there is less bias and more completeness in the list of criteria. This can also be applied to the validation and practical use. Statistical models can be applied to prove the validity with more certainty and the relations that it claims.

Although this study has focused solely on apps for smartphones, the research area can be expanded to include tablets. Most tablets are marketed like smartphones, which would mean a large overlap with them. Even further widening the area, it could be applied to interactive television sets, like AppleTV and GoogleTV, which can also show webapps.

The current study has reviewed only two kinds of multi-attribute decision making (MADM) methods: analytical hierarchy process (AHP) and simple additive weighting (SAW). There are more methods

that can be investigated for this research that are suitable to multi-attribute decision making, examples are TOPSIS, multiplicative exponential weighting, and ELECTRE.

9 REFERENCES

Alter, S. (2004). "A work system view of DSS in its fourth decade." Decision Support Systems **38**(3): 319-327.

Apple. "iOS Human Interface Guidelines." Retrieved 22 August 2012, from http://developer.apple.com/library/ios/#DOCUMENTATION/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html.

Arnott, D. and G. Pervan (2005). "A critical analysis of decision support systems research." Journal of Information Technology **20**(2): 67-87.

Belton, V. (1986). "A comparison of the analytic hierarchy process and a simple multi-attribute value function." European journal of operational research **26**(1): 7-21.

Budgen, D. and P. Brereton (2006). Performing systematic literature reviews in software engineering, ACM.

Charland, A. and B. Leroux (2011). "Mobile application development: web vs. native." Communications of the ACM **54**(5): 49-53.

Deloitte (2012). "So many apps - so little to download." TMT Predictions.

Dyer, J. S. and R. K. Sarin (1979). "Measurable multiattribute value functions." Operations Research **27**(4): 810-822.

Eisenhardt, K. M. (1989). "Building theories from case study research." Academy of management review: 532-550.

Er, M. (1988). "Decision support systems: a summary, problems, and future trends." Decision Support Systems **4**(3): 355-363.

Facebook (2012). "Introducing New Mobile Ads For Apps." Retrieved 14 August 2012, from https://developers.facebook.com/blog/post/2012/08/07/introducing-new-mobile-ads-for-apps/.

Finlay, P. N. (1994). Introducing decision support systems, Blackwell Pub.

Forrester (2011). Western European Mobile Forecast, 2011 To 2016.

Gartner (2012). "Gartner says worldwide sales of mobile phones declined 2 percent in first quater of 2012; previous year-over-year decline occured in second quarter of 2009." Retrieved 10 August 2012, from http://www.gartner.com/it/page.jsp?id=2017015.

GeoChalkboard (2012). "Anatomy of a hybrid mobile GIS application." Retrieved 24 August 2012, from http://www.geospatialtraining.com/blog/?p=1897.

Ginzberg, M. J. and E. A. Stohr (1982). "Decision support systems: issues and perspectives."

Goepel, K. (2012). "New AHP Excel template with multiple inputs." from http://bpmsg.com/new-ahp-excel-template-with-multiple-inputs/.

Google. "User Interface Guidelines." Retrieved 22 August 2012, from http://developer.android.com/guide/practices/ui_guidelines/index.html.

Google. "Web Apps Overview." 24 August 2012, from http://developer.android.com/guide/webapps/overview.html.

Google (2013). "Platform Versions." Retrieved 1 March 2013, from http://developer.android.com/about/dashboards/index.html.

Hevner, A. R., S. T. March, et al. (2004). "Design science in information systems research." Mis Quarterly **28**(1): 75-105.

Hove, S. E. and B. Anda (2005). Experiences from conducting semi-structured interviews in empirical software engineering research, IEEE.

IDC (2012). "Android and iOS surge to new smartphone OS record in second quarter, according to IDC." Retrieved 20 August 2012, from https://www.idc.com/getdoc.jsp?containerId=prUS23638712.

Intelligence Group (2012). "Facebook en whatsapp meest populaire apps onder Nederlandse beroepsbevolking." Retrieved 22 August 2012, from http://www.intelligence-group.nl/nl/actueel/augustus-2012/nieuws/facebook-en-whatsapp-meest-populaire-apps-onder-nederlandse-beroepsbevolking.

Işıklar, G. and G. Büyüközkan (2007). "Using a multi-criteria decision making approach to evaluate mobile phone alternatives." Computer Standards & Interfaces **29**(2): 265-274.

Keeney, R. L. and D. von Winterfeldt (2009). "Practical value models."

Kitchenham, B. (2004). "Procedures for performing systematic reviews." Keele, UK, Keele University **33**: 2004.

Kitchenham, B., O. Pearl Brereton, et al. (2009). "Systematic literature reviews in software engineering-A systematic literature review." Information and Software Technology **51**(1): 7-15.

McLeod, R. and G. Schell (2001). Management Information Systems 8/e, Prentice-Hall, Inc.

Microsoft. "User Experience Design Guidelines for Windows Phone." Retrieved 22 August 2012, from http://msdn.microsoft.com/en-us/library/hh202915%28v=VS.92%29.aspx.

Miles, M. B. and A. M. Huberman (1994). Qualitative data analysis: An expanded sourcebook, Sage Publications, Incorporated.

Myers, M. D. and M. Newman (2007). "The qualitative interview in IS research: Examining the craft." Information and organization **17**(1): 2-26.

Nokia (2011). "Nokia outlines new strategy, introduces new leadership, operational structure." Retrieved 24 August 2012, from http://press.nokia.com/2011/02/11/nokia-outlines-new-strategy-introduces-new-leadership-operational-structure/.

Padley, R. (2011). "HTML5–bridging the mobile platform gap: mobile technologies in scholarly communication." Serials: The Journal for the Serials Community **24**: S32-S39.

Saaty, T. L. (1980). Analytic hierarchy process, Wiley Online Library.

Saaty, T. L. (1990). "How to make a decision: the analytic hierarchy process." European journal of operational research **48**(1): 9-26.

Saunders, M., P. Lewis, et al. (2009). Research methods for business students, Prentice Hall.

Slashdot (2011). "Native apps are dead, long live native apps." Retrieved 1 August 2012, from http://developers.slashdot.org/story/11/06/27/2131216/native-apps-are-dead-long-live-native-apps.

Thomas, G. (2011). "A typology for the case study in social science following a review of definition, discourse, and structure." Qualitative Inquiry **17**(6): 511-521.

Verschuren, P. J. M. and H. Doorewaard (2007). Het ontwerpen van een onderzoek, Lemma.

W3 Cubes (2011). "Anatomy of a HTML5 mobile application." Retrieved 24 august 2012, from http://w3cubes.com/blog/2011/10/26/anatomy-of-a-html5-mobile-application/.

Wasserman, T. (2010). "Software engineering issues for mobile application development." FoSER 2010.

Wieringa, R. (2009). Design science as nested problem solving, ACM.

Wieringa, R. J. (1996). "Requirements engineering: frameworks for understanding."

Yin, R. K. (2009). Case study research: Design and methods, Sage publications, INC.

Zanakis, S. H., A. Solomon, et al. (1998). "Multi-attribute decision making: A simulation comparison of select methods." European journal of operational research **107**(3): 507-529.

Zelkowitz, M. V. and D. R. Wallace (1998). "Experimental models for validating technology." Computer **31**(5): 23-31.

10 APPENDIX A: EXAMPLES OF APPS

The NU.nl app on different OS's, from left to right: Android, iOS, WindowsPhone, and Blackberry

11 APPENDIX B: NATIVE CODE

In this appendix we show a native code example for each native platform. The programs has to display the text “Hello World”, which the classic example when a programmer starts in an unfamiliar development environment. Although the complexity is not in question, it is to show the differences between native SDK and programming language. Auto-generated files are omitted, the links provided with each example leads to the original tutorial with extended explanations of the code.

Android

Based on a tutorial provided by: http://androidcodemonkey.blogspot.nl/2010/01/hello-world-your-first-android.html (retrieved on 30 August 2012). This code shows a toast message.

HelloWorld.java

```
package com.android.test;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class HelloWorld extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button button = (Button) findViewById(R.id.Button01);
        button.setOnClickListener(new OnClickListener() {
           @Override
           public void onClick(View v) {
            Toast.makeText(HelloWorld.this, "Hello World",
Toast.LENGTH_SHORT).show();
           }
         });
    }
}
```

iOS

Based on a tutorial provided by: http://idevzilla.com/2010/09/06/ios-helloworld-example/ (retrieved on 30 August 2012). It shows a plain label on the screen.

HelloWorldViewController.m

```
- (void)loadView {
      UIView *mainView = [[UIView alloc] initWithFrame:[UIScreen
      mainScreen].bounds];
      CGRect labelFrame = CGRectMake(50, 50, 200, 30);
      UILabel *helloLabel = [[UILabel alloc] initWithFrame: labelFrame];
      helloLabel.text = @"Hello World";
      [mainView addSubview:helloLabel];
      [helloLabel release];
      self.view = mainView;
      [mainView release];
}
```

WindowsPhone

Based on a tutorial provided by: http://www.codeproject.com/Articles/312963/Hello-World-Application-in-Windows-Phone-7 (retrieved on 30 August 2012). It shows a button with the text "hello world" on the screen.

MainPage.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;

namespace HelloWorldApp
{
     public partial class MainPage : PhoneApplicationPage
     {
          // Constructor
          public MainPage()
          {
              InitializeComponent();
          }

          private void button1_Click(object sender, RoutedEventArgs e)
          {
              textBlock1.Text = "Hello World!!!";
          }
     }
}
```

12 APPENDIX C: INTERVIEW PROTOCOL

The protocol describes the type of interview, the resources used and a general outline of an interview.

Type of interview

The interview conducted will be qualitative in nature, in the form of a conversation. A semi-structured interview has the advantage to set out the big picture, while leaving room for (social) interaction: deviation on topics and/or further questioning on specific topics. A structured interview (or questionnaire) is most likely too limited for our goal to gain insight, where we need the flexibility to ask further, go into details, or deviate from the set of questions. An unstructured could work, but is not preferred as we would like to loosely compare answers on several topics which require general questions.

Resources

The following resources will assist with the interview, and make it easier to record and process the information of the interviews:

- **Recording device**: in this case a smartphone that records the conversation)
- **Notebook** for comments and annotations during the interview
- **Laptop**
- **Background information**: A document with a summarized description of the thesis
- **Interview script**: The script with questions

Approach

The first contact will be by email or by phone to explore the possibility of an interview with the potential candidate. If the candidate is Dutch, then the language will be in Dutch; otherwise English will be used.

Initial contact consists of the following elements:

1. Introducing myself
2. Introduction and goal of the thesis
3. Explanation and goal of interviews
4. Ask for participation and/or contact

The initial contact is only the first step in the process.
A complete overview of steps is:

1. **Initial contact**: Introduction, question to participate
2. **Reaction**: Does the participant want to accept the interview
3. **Make appointment**: Setting the date and time for the interview
4. **Interview**: Take the interview (no more than 60 minutes)
5. **Summary**: Sent a summary back to participant for approval
6. **Approval**: Interviewee approves the summary
7. **Sent final public thesis**: When thesis is done, send a digital copy

Outline

The outline of the interview with a time estimation of each action is presented in the table below. Duration is expected time in minutes.

Duration	Action	Description
5	Opening	Introduce myself: Name – Study – Experiences – Permission for recording - Ask how data should be treated
5 – 10	Introduction	Introduce main topic and goal of thesis – purpose of interview
30 – 40	Key questions	Getting answers on the main questions
10	Follow ups	Return to interesting topics and/or details
5	Close	Ask for a follow-up interview/contact

Outline with time indication per action

Leading questions

1. What is your experience with mobile applications and implementation platforms?
2. What are (common) business drivers behind mobile apps?
3. What are (common) implementation requirements for mobile applications
4. When in an implementation platform preferred above another platform?
5. How do company competencies impact the choice for a mobile platform?

Questions with follow up questions

The general questions are:

1. What is your experience with mobile applications and implementation platforms?
    a. What mobile devices/equipment do you differentiate? Why?
    b. Are you familiar with Native/Web/Hybrid platforms?
2. What are (common) business drivers behind the mobile apps?
    a. How do the cost-drivers affect the app and implementation platform?
    b. How do cost-drivers relate to each other?
    c. How do support and licenses of an implementation affect the choice?
    d. How high is the necessity to develop and maintain the app in-house?
    e. For business applications: Does MEAP (mobile enterprise application platform) and/or BYOD (bring your own device) determine the mobile platform?
3. What are (common) implementation requirements for applications?
    a. How important is the security aspect?
    b. Usage of smartphone specific sensors? (Camera / GPS / Accelerometer / Touch)
    c. What is the impact of the implementation language (Java / .Net / objective-C / web)
    d. What is the role of the IDE (Eclipse / Visual studio / Xcode)
    e. For multiplatform: Is a single codebase preferred?
4. When is an implementation platform preferred above another platform?
    a. Could certain requirements and/or drivers be linked to implementation platforms?
    b. What requirements/drivers would exclude an implementation platform?
5. How do company competencies impact the choice for a mobile platform?
    a. Inhouse expertise // preferentials // Internal used software
    b. How do they work? // development process
    c. How is the quality of an app measured

13 APPENDIX D: SUMMARIES OF INTERVIEWS

Eight of the nine interviews were conducted in Dutch, and thus also transcribed and summarized in Dutch. The summaries are presented in the chronological order in which the interviews were conducted.

Interview #1

Interview informatie

Datum: 14 June 2012

One-liners

- Als je geen app hebt, dan tel je niet mee
- De kwaliteitsnorm wordt bepaald door wat de verwachtingen van de end-user
- Mobiele OS'en worden steeds beter, behalve bij Android
- Native is the way to go voor kwalitatieve (en complexe) apps

Samenvatting interview

Na de introductie van mezelf, introductie van scriptie, doel van interview en introductie van Ewald begon het interview.

Voor onderscheidt in devices stelt Ewald duidelijk dat het de grote mobiele OS'en die domineren: iOS, Android, Windows Phone, en eventueel BlackBerry. Waarbij hij de voorkeur geeft aan iPhone/iPad met iOS van Apple (first choice); gevolgd door smartphones/tablets met Android.

Hij ziet het meeste heil in de Native SDK van elk platform, omdat deze leiden tot de hoogste kwaliteit van apps. Voor hybrid oplossing (zoals PhoneGap / Appcelerator) heb je vaak meer tijd nodig om te programmeren, omdat het niet altijd werkt als van te voren gedacht. Daarnaast wil de documentatie van de verschillende HTML5+Hybrid approach nog weleens te wensen overlaten. Een voorbeeld is de Google Maps Api, die werkt uitstekend in Android maar werkt omslachtiger in iOS.

Over business drivers kunnen we kort zijn: Als je geen app hebt, dan tel je niet mee. Apps zijn de nieuwe websites, het is de presentatie van je bedrijf.
Drivers als budget drukken natuurlijk wel een stempel op je product, een simpele app (extensie van website bijvoorbeeld) is prima te maken in HTML5/Hybride vorm. Maar bij interactie en gebruik van sensoren is Native aan te raden, ook al brengen die vaak (iets) hogere kosten met zich mee. Ook dit valt en staat met de kwaliteit die het bedrijf en consument verwacht bij de app.
Support en SLA's horen zeker bij apps, vooral bij B2C hoort een goede(continue) support en vaak korte release cycles / update cycles. Apps in markets/stores/Play worden heel snel afgerekend als iets niet (helemaal) werkt zoals de consument verwacht.
Een goede driver voor een app is als je een idee/creatief idee hebt. Het liefste iets dat ondersteund of aansluit bij een continue menselijk proces. Voorbeeld is een app voor zwangere vrouwen.
Als laatste vroeg ik welke rol opensource kan spelen. Ewald gelooft hier niet zo in, het is mooi voor hobby projecten en/of ideële projecten. Maar voor serieuze business is het beter om de app code

closed te houden, onder andere om je business model te beschermen tegen het kopiëren van app-logica en app-code.

Een belangrijke pijler in de hedendaagse IT landschap is security, dit geldt dus ook voor apps. Om dit direct te relativeren wil Ewald toevoegen dat je wel moet nagaan in hoeverre je (gevoelige) data op je device/app hebt staan, en hoeverre je dit kunt en wilt veiligstellen/beschermen. Hierbij gaat het om het gebruikersgevoel van veiligheid bij de app. Bij geldtransacties wil je de allerhoogste beveiliging en wil de gebruiker het gemak opofferen voor extra security stappen; maar bij een berichtje plaatsen is de gebruiksvriendelijkheid van een app veel belangrijker (er valt niets te halen uit de app / niet belangrijke info).
Het is een afweging tussen hoever je de security wilt hebben, en in welke mate dit belangrijk is voor de app. Dit betekend echter niet dat basis principes als HTTPS/SSL mogen ontbreken!

Op het gebied van sensoren zijn de GPS en de camera duidelijk het belangrijkste, hoewel je dit wel bij voorkeur in native implementeert aangezien hybride niet werkt (of niet consistent werkt). De extra features van de smartphones door middel van sensoren zijn de echte added value van een app ten opzichte van een (mobiele) website.

De vraag in hoeverre programmeertalen en IDE's een rol spelen blijkt minder groot. Het is zaak om mensen direct van de Hogeschool of Universiteit te rekruteren, waarbij ze een aantal weken op (spoed)cursus gaan om de taal te leren en gedachtegang achter apps onder de knie te krijgen. In alle gevallen geldt dat de limitatie niet bij de programmeur ligt, maar bij de smartphone. Een voorbeeld is dat hele zware apps een telefoon zo drainen dat de accu direct leeg is, terwijl de light-variant juist niet voldoet aan wensen van de gebruiker qua interface.

Over de verschillende versies van de OS'en is Ewald best te spreken, waarbij het bij elke release cycle beter wordt. Echter is een observatie dat Android heel versnipperd raakt (relatief vaak updates) mede doordat het heel veel devices, hardware en schermgroottes wil ondersteunen (in tegenstelling tot iOS en WP7). Uiteindelijk gaat het erom dat de code voor een OS er op elk device er hetzelfde uitziet, en dit wil bij Android nog weleens tegenvallen (grote verscheidenheid aan hardware en screensizes).

Terugkomend op de markets, hier is Apple wellicht te strikt voor developers. De andere kant van de medaille is dat de consument een gegarandeerde werkende app krijgt, waarbij de app werkt zoals de gebruiker zou verwachten (middels HIG en andere guidelines). Android is te los hierin, daarbij kan iedereen een app (zonder keuring) in de market zetten.

We concludeerden het gesprek met de verbanden tussen de besproken onderwerpen en de voorkeur voor implementatieplatformen. Uiteindelijk komt het neer op het budget + kosten en de ROI (return of investment); maar ook in hoeverre er kwaliteit van de app verwacht wordt door de end-user en wat het bedrijf er voor over heeft.
Echte kwaliteit voor apps kan alleen bereikt worden door native solutions, zeker in combinatie met het gebruik van sensors op de telefoon. Wanneer men bereid is om met mindere kwaliteit genoegen te nemen, dan zijn hybride vormen ook een optie.

Interview #2

Interview informatie

Datum: 15 juni 2012

One-liners

- De added value van een mobiele app behaal je door de mogelijkheden te benutten van het device // Je app moet meer zijn dan een veredelde website
- De keuze van hybride middleware komt voort uit de eisen futureproof en multiplatform
- Preferred platform of choice heeft altijd invloed op de keuze

Doel interview

Aan de ene kant probeert het interview mijn vermoedens over de business drivers en technische requirements te bevestigen, en aan de andere kant probeer ik via de gesprekken inzicht te krijgen hoe ze tot elkaar verhouden. Daarnaast dienen de interviews om verbanden inzichtelijk te maken tussen de drivers/requirements en de mobiele implementatie platformen (native / web / hybride).
Hierbij zijn de leidende vragen:

- Wat is uw ervaring en achtergrond met mobiele applicaties en implementatie platformen?
- Wat zijn de business drivers achter mobiele applicaties?
- Wat zijn technische requirements voor mobiele applicaties?
- Wanneer en waarom prefereert u een platform boven een ander platform?

In het interview met Jeroen kwam goed naar voren welke selectie criteria ze hebben gehanteerd en hoe ze de verschillende alternatieven hebben afgewogen. Deze procedure geeft inzicht in de belangrijkste criteria voor het hybride platform boven andere implementatie platformen.

Samenvatting interview

Na de introductie van mezelf, introductie van scriptie, doel van interview en introductie van Jeroen begon het interview.

De business drivers komen vooral naar voren uit de business propositie; hierin staat beschreven wat de toegevoegde waarde is en wat het “nut”. In dit geval gaat het om het verkorten van reistijd en efficiënter omgaan met administratieve handelingen, in beide gevallen maken we gebruik van de sensoren op het device. Middels de GPS kun je navigatie doen in je app, maar ook effectiever routeplannen (voor de centrale, doordat de locatie bekend is). Voor de administratieve handelingen geldt dat de touch interface een goede en snelle manier is van interactie met de flow voor de afhandeling van het proces dat engineers op de locatie moeten uitvoeren. De camera draagt hierbij toe dat er fotos gemaakt kunnen worden, maar ook om barcodes te scannen. Dit laatste kan ook gebruikt worden om de inventaris van de service-bus bij te houden.

Het budget is voortgekomen uit de besparing die de app moet gaan opleveren. Hierbij wordt gekeken wat de benefits zijn en welke ROI (return of investment) je daarbij wilt halen. Er wordt zeker wel

gekeken naar kosten als drivers, maar hierbij weet de klant dat een degelijk product (app) de nodige kosten met zich mee brengt. Dit wat ook een hoofdreden voor de keuze van een third party die de implementatie doet. Deze third party heeft een eigen middleware platform voor mobiele applicaties waardoor er cross-platform ontwikkeld kan worden. Daarnaast heeft die ervaring met de backoffice producten en services wat een erg belangrijke pré is.

De kwaliteit van de mobile app is erg belangrijk waarbij de selectie van de implementatie partner gekeken is naar ervaring met apps, eigen platform, support, en ervaring met de backoffice. Het bouwen van een app is namelijk 1 ding, maar de integratie binnen het huidige systeem is de echte uitdaging. De keuze van outsourcing van ontwikkeling lag voor de hand, omdat er intern geen ervaring is met apps.

Een andere driver is de support en licenties voor het platform. Het huidige licentie model is pay-per-user, waardoor er flexibiliteit ontstaat in piektijden voor de app. Daarnaast is support op de applicatie een belangrijke requirement voor het continue proces.

Hoewel de klant zelf nog weinig ervaring heeft met apps, hebben ze wel voorwaarden bij security zoals HTTPS verbindingen en encrypted data op het device. Daarnaast hebben ze graag MDM oplossing (Mobile Device Management) waarbij ook een remote wipe gedaan kan worden.

De features die een mobiele smartphone kan bieden bij de app, zijn het echte selling point en added value. Deze app gebruikt het touchscreen, gps en camera actief in het process; daarnaast wordt ook SMS gebruikt voor notificaties. Omdat features zo belangrijk zijn, is het ook interessant te kijken naar de selectie procedure naar het juiste device. Er is uiteindelijk gekozen voor een smartphone en niet voor een tablet. Een engineer heeft namelijk al een laptop en telefoon bij zich, een tablet zou een derde device zijn. De laptop is namelijk onmisbaar bij diagnosticeren en updaten van firmware. Hierdoor lag de keuze van een smartphone (met groot scherm) voor de hand, zodat ze enkel een laptop en telefoon bij zich hoeven te hebben. Het voordeel van de smartphone is dat deze altijd aanstaat en snel benaderbaar (in tegen stelling tot een laptop).
Bij de keuze van de smartphone is er er onder andere gekeken naar schermgrootte, battery life, snelheid, en OS. Er is een voorlopige keuze gemaakt, maar de techniek gaat zo snel (zowel hardware als software) dat het belangrijk was dat de applicatie op meerdere platformen zou kunnen draaien.

Bij het opstellen voor de app is er de top-down approach gebruikt waarbij de business drivers (project drivers) als eerste benoemd werden, vervolgens een platform gekozen. Uit de drivers kwam naar voren dat een handheld device prima een aantal drivers ondersteunde. Toen kwam er al snel uit dat er een hybride app gemaakt zou worden, vanwege de ondersteuning van vele devices (in meerdere landen).

Uiteindelijk is android de preferred platform of choice geworden omdat deze vele devices had en apps eenvoudiger aan te passen zijn. Bij Apple zit je namelijk direct aan 1 toestel vast (iPhone/iPad), en heeft de platformhouder (Apple) vele restricties voor apps, inclusief een strenge goedkeuring die enkele weken kan duren.

Ten slotte nog een korte toelichting op consumer smartphones vs specialized devices. Speciale ruggedized devices zijn vaak erg duur, is minder keuze qua toestellen en lopen qua techniek/OS vaak achter op de consumenten varianten. Tevens zijn er voor consumenten smartphones genoeg ruggedized cases te vinden, waardoor het toestel een stuk beter beschermd wordt. De kosten wegen dan niet tegen de baten.

Interview #3

Interview informatie

Datum: 18 Juni 2012

One-liners

- De app begon als experiment omdat we vonden dat er in het mobiele platform erg veel potentie zat
- HTML5 veranderd niet zoveel, CSS3 en JavaScript zullen een grotere impact hebben op de “snappyness” van User Interface (en kwaliteit) van de hybride apps
- Op dit moment zijn we iPhone only; Android brengt hogere development kosten vanwege device range, lagere overall kwaliteit, en ontwikkelaars zijn moeilijk aan te trekken

Doel interview

Aan de ene kant probeert het interview mijn vermoedens over de business drivers en technische requirements te bevestigen, en aan de andere kant probeer ik via de gesprekken inzicht te krijgen hoe ze tot elkaar verhouden. Daarnaast dienen de interviews om verbanden inzichtelijk te maken tussen de drivers/requirements en de mobiele implementatie platformen (native / web / hybride).
Hierbij zijn de leidende vragen:

- Wat is uw ervaring en achtergrond met mobiele applicaties en implementatie platformen?
- Wat zijn de business drivers achter mobiele applicaties?
- Wat zijn technische requirements voor mobiele applicaties?
- Wanneer en waarom prefereert u een platform boven een ander platform?

In het interview met Edwin kwam naar voren dat niet alles vooraf bedacht was omdat de app begon als een experiment, achteraf gezien bleken de gemaakte keuzes toch in de juiste richting te zijn gemaakt voor MoneyBird. Uit het interview bleek dat de voornaamste drivers achter de app kwaliteit, user experience en ROI zijn. De ROI is in dit geval het aantal nieuwe gebruikers/aanmeldingen dat de app genereerd voor MoneyBird.

Samenvatting interview

Na de introductie van mezelf, introductie van scriptie, doel van interview en introductie van Edwin begon het interview.

Doordat MoneyBird groter werd, hadden ze gekozen voor voor een app omdat daar erg veel potentie in het mobiele platform zat. De app werd initieel voor iPhone gemaakt als experiment, omdat Berend ervaring had met C en de overstap naar Objective C een relatief eenvoudige stap was. De overige ontwikkelaars binnen MoneyBird programmeren voornamelijk in Ruby on Rails.

De insteek van de app was om facturen onderweg benaderbaar te maken en ook om nieuwe facturen te kunnen creëren. Uit analyse bleek dat een gemiddelde factuur evenveel tekst bevat als SMS, waardoor de feasibility van het aanmaken van facturen zeer hoog ligt. Later kwam de feature erbij

om ook bonnetjes te kunnen “scannen”, wat in feite een foto maakt van een bon en deze vervolgens als attachment upload in een nieuw inkoopfactuur.

De leuke reacties van klanten, de (positieve) media aandacht voor de bonnetjes scanner, goede zichtbaarheid in de appstore, en het feit dat er vele nieuwe gebruikers voortkwamen zorgde er voor dat de app steeds belangrijker werd.

Op de vraag of er ook nog een overstap gemaakt wordt naar Android, komt een genuanceerd antwoord waarbij Edwin vertelt dat een Android versie veel gevraagd is maar er nog niet gaat komen. Android is lastig vanwege hogere development kosten (grotere range aan devices en bijbehorende schermresoluties) en het afwezig zijn van voldoende inhouse Android kennis. Het aantrekken van Android programmeurs blijkt problematisch, en outsourcen creëert een afhankelijk zonder garanties voor (snelle) updates, support en overall kwaliteit. Verder is zijn mening dat de Android minder kwaliteit “afdwingt” vanwege de vele devices die het moet supporten, ziet het er allemaal net wat minder gelikt uit als bij iOS.

Kwaliteit is een belangrijke pijler voor MoneyBird, en hybride oplossing biedt nog niet de gewenste “snappyness” en kwaliteit. Hij verwacht wel dat als HTML5/PhoneGap/Appcelerator beter gaat worden dat er dan nogmaals besloten moet worden of een hybride (of web only) app ook een alternatief is. De verbeteringen van browsers in combinatie met javascript, css, html en nieuwe functies, die bijvoorbeeld mogelijk maken om direct foto’s te maken met de camera en die up te loaden, vormen een cruciale rol om hybride apps een betere positie te geven ten opzichte van Native apps.
Bij script to native compilers (hybrid) geld dat je altijd met het OS rekening moet houden, zoals de backbutton die wel in Android zit maar niet in iOS. Als je logica moet inbouwen voor elk platform (if android then xx; if ios then yy), dan is de vraag of je niet beter native kan ontwikkelen.

Wel gaat de app de overstap maken naar de iPad, waarbij de extra schermruimte goed benut kan worden voor meer details bij rapportages.

Als we kijken naar de impact van de app op de releasecycle, dan zie je dat de web app met 1 druk op de knop compleet up-to-date is, terwijl je bij app dit niet kunt afdwingen. Daarnaast moet de app nog goedgekeurd worden (max 3 weken door Apple), en vervolgens moeten de gebruikers hem nog updaten. Dit heeft implicaties dat er mogelijk verschillende API’s en koppelingen moeten blijven werken. Dus een app creëert een nieuwe afhankelijkheid op het bestaande web platform, welke nu voor lief genomen wordt omdat de app veel nieuwe klanten oplevert.

Als ze nu een bewuste keuze moesten maken om een app te maken, dan zouden ze weer als eerste voor een native iPhone app zijn gegaan. Dit heeft te maken de zakelijke gebruikersgroep (klein MKB en ZZP’ers) die toch veelal een iPhone gebruiken. Het installeren van apps en het gebruik van apps ligt hoger op iPhone dan op Android. Toch vind Edwin het jammer dat ze een groot deel van de gebruikers (Android) buiten sluiten, maar aan de andere kant is hij bang dat de kwaliteit van de app op Android het niet gaat halen bij hun huidige app. In zijn ogen loop iOS voorop qua ontwikkelingen op smartphone gebied.

Op de vraag hoe het zit met open source en licenties, is hij behoorlijk duidelijk: We werken met Ruby on Rails, waarbij opensource handig is. Echter hoeft closed source geen probleem te zijn, mits goed gedocumenteerd. Daarnaast werken we altijd al met Mac en iPhone, dus licensies en hardware aanschaf voor iPhone development was geen issue.

Security is een punt waar veel aandacht aan wordt besteedt: Extra pincode op facturen (binnen de app), HTTPS verbinding met de server, inloggen via OAuth methode, internetverbinding vereist om nieuwe facturen aan te maken. Ook worden er preventieve security scans gedraaid om eventuele lekken te achterhalen voordat ze bekend zijn. Je moet altijd meer security inbouwen dan er verwacht wordt, een data-lek en negatieve media aandacht kan funest zijn voor het imago van een bedrijf.
Overigens verbaasd het hem dat de gebruikers wel bewust zijn van (gevoelige) data op hun device, maar bij support is het geen probleem om toestemming te krijgen om in de data te mogen meekijken.

Qua sensors heeft de app enkel de camera mogelijkheden nodig, en mogelijk gaat dit in de volgende versie van iOS native ondersteund worden vanuit de browser. In dat geval zou het ook een web app kunnen worden. Via local storage wordt er data gesynched wordt zodat er ook data beschikbaar is wanneer er geen internetverbinding is.
Verder wil hij geen sensors gebruiken die enkel als gadget dienen, zoals gps geotagging, dropbox koppelingen etc; dit zou de kwaliteit enkel omlaag halen, het moet echte added value hebben.

Op de vraag voor koppeling tussen drivers en implementatie platformen schets hij het volgende beeld:

- Native = kwaliteit + marketing waarde van appstore (sign ups uit halen)
- Web = uniforme werkwijze tov apps (monitoring + deployment + workflow)
- Hybride = Beste van beide werelden, hoewel de kwaliteit niet kan tippen aan native. Workflow kan probleem zijn, veel platformen en hoe kan ik de releases op elkaar afstemmen (launch zowel android + iOS en anderen). Je blijft niet 1 ding hebben.

Technische requirements hebben minder invloed dan business drivers. Kwaliteit is kritiek: Liever 1 device die 100% kwaliteit heeft, dan multiple devices waar het net niet werkt.Kosten maken minder uit, ze halen een substantieel deel inkomsten uit de app. Verder willen ze inhouse ontwikkelen.
In de toekomst verwacht hij meer kansen voor hybride- en web app: Niet zozeer HTML5, maar CSS3 is wel belangrijk (en ondersteuning in de browser). CSS is namelijk belangrijker voor de User Interface en de Experience (CSS + Javascript). De kwaliteit is belangrijkste, de snappiness.

Interview #4

Interview informatie

Datum: 20 Juni 2012

One-liners

- iOS is het belangrijkste, Blackberry is dood, Android en WP7 zitten er tussen in
- De APIs die beschikbaar zijn voor native OS zijn gewoon beter dan wat er beschikbaar is voor hybrid/web
- Korte lijntjes en communicatie: Vandaag iets bedenken en morgen een prototype te hebben
- Opensource is heel mooi en heeft zo zijn nut en functie; het is alleen niet altijd toepasbaar

Doel interview

Aan de ene kant probeert het interview mijn vermoedens over de business drivers en technische requirements te bevestigen, en aan de andere kant probeer ik via de gesprekken inzicht te krijgen hoe ze tot elkaar verhouden. Daarnaast dienen de interviews om verbanden inzichtelijk te maken tussen de drivers/requirements en de mobiele implementatie platformen (native / web / hybride).
Hierbij zijn de leidende vragen:

- Wat is uw ervaring en achtergrond met mobiele applicaties en implementatie platformen?
- Wat zijn de business drivers achter mobiele applicaties?
- Wat zijn technische requirements voor mobiele applicaties?
- Wanneer en waarom prefereert u een platform boven een ander platform?

Het was een vlot gesprek waarbij er zeer open gesproken werd over de verschillende implementatie platformen. Hierbij was het interessant te horen hoe er omgegaan werd met de verschillende OS'en, welke links er gelegd worden tussen drivers en platformen, en een blik op de toekomst.

Samenvatting interview

Bekend met alle implementatie platformen (web, native, hybrid), maar bij Repudo werken ze enkel native apps. De markt is nog erg jong waarbij de beste ervaring gegeven wordt via native programmas. De keuze hiervoor komt voort uit verschillende drivers waarbij de user experience voorop staat. Daarnaast zijn er nog drivers als (wereldwijde) distributie, monetisatie (conversie), documentatie(API), en gebruik van sensors.

Middleware is niet interessant, zeker niet voor ons. Repudo wil echt de techniek ontwikkelen, we gaan iets moeilijks maken. Dat betekend dat het niet overal op werkt (dat kan niet), ook geeft de native SDK meer (en mooiere) opties om met device sensors te doen. Voor Repudo gebruiken we de GPS en camera.
Er zijn meerdere manieren om conversie (monetisatie) te bereiken, via businessmodellen die verschillen tussen de campagnes en doelgroep. Binnen de app zelf vinden er geen monetaire transacties plaats, mocht dit ooit veranderen dan zouden ze een payment module hiervoor gebruiken. Als je zoiets dergelijks gaat bouwen, dan moet het ook meteen je core business zijn.

Volgens Niels en Lucas is de volgorde van belangrijkste markten (qua volume en volwassenheid):

- iOS, het meest volwassen platform voor apps
- Android, heeft het grootste markt aandeel maar is totaal versnipperd en heeft veel goedkope toestellen. Geen controle op de market betekend veel malware
- WP7, met als aantekening dat ze zeer benieuwd zijn naar de ontwikkelingen van windows phone 8. Dit kan potentieel een groot platform worden.
- BlackBerry, welke in feite al niet meer meetelt in het app landschap

Repudo ontwikkeld tegenwoordig alles inhouse en enkel met de native SDK's. Ze hebben outsourcing (met hybride modellen) geprobeerd maar dit bleek verre van ideaal te werken. Ze willen namelijk snelle communicatie en rapid prototyping. Met outsourcing naar India bleek dit niet mogelijk. Het boven op het product zitten, korte lijntjes, en het snel realiseren van ideeën zijn drivers om het vooral inhouse te houden.
Programmeurs binden blijkt een groot probleem te zijn. Ze geven de programmeurs veel vrijheid in werken zoals ze dat willen (kapitein op eigen scihp), waarbij er zoveel mogelijk nieuwe technieken toegepast/uitgeprobeerd worden.

Bij ontwikkelingen is het vaak als eerste voor de iPhone, omdat dit platform het meest volwassen is. De beperkingen door Apple blijken eigenlijk geen (groot) probleem te zijn, zoals de HIG en strenge app controle. Ook loven ze over de support afhandeling van de app controle; hierbij kan Microsoft nog wat leren en merk je dus dat ze nog moeten groeien.

Support is een zeer belangrijk aspect van een app. Hierbij proberen ze directe feedback te beantwoorden van klanten (ongeacht of het positief of negatief is), en directe support te verlenen bij problemen. Ze merken hierbij op dat support voor iPhone het makkelijkst is omdat er relatief weinig toestellen zijn, en de meesten updaten direct naar de laatste versie. Bij Android blijkt dit toch een stuk moeilijker, want hier heeft slechts 7% de nieuwste versie en blijven er veel op Android 2.1/2.1/2.3 steken. Support voor Blackberry is min of meer gestaakt, en het was ook een "crime" om te ondersteunen.

Opensource is goed voor bepaalde doelen en frameworks, maar je bent wel afhankelijk van de activiteit van die community. Verder hebben ze geen probleem met closed source (Repudo is ook closed source), zolang de documentatie maar uitgebreidt en verzorgd is.

Verbanden tusen platformen en de drivers: user experience, security models en volwassenheid geeft het volgende rijtje:

- iPhone: Scoort het hoogste op alle punten
- Windows Phone
- Android: Geen controle, dat zet de deur open voor malware
- Blackberry: doet niet meer mee

Verder moeten er bij apps gekeken worden naar de doelgroep en de demografie ervan. Android bedient een andere markt (veel goedkope toestellen, populairder onder mensen die gewoon en smartphone willen); terwijl iPhone een high-end markt bediend (grotere interesse in apps).

Tenslotte nog een korte blik op de toekomst: Ze verwachten een nog diepere integratie met verschillende platformen zoals met social media platformen en standaardisatie in betalingen.

Interview #5

Interview informatie

Datum: 21 Juni 2012

One-liners

- Het allerbelangrijkste voor een app is de user-experience, de app moet het leven makkelijker of leuker maken van mensen.
- PhoneGap is een oplossing als je geen(of weinig) tijd en/of resources (budget/kennis) hebt om je app native te maken
- Ik ben ervan overtuigd dat een app vooral 1 ding goed moet doen, wat je app clean, lean en mean houdt

Doel interview

Aan de ene kant probeert het interview mijn vermoedens over de business drivers en technische requirements te bevestigen, en aan de andere kant probeer ik via de gesprekken inzicht te krijgen hoe ze tot elkaar verhouden. Daarnaast dienen de interviews om verbanden inzichtelijk te maken tussen de drivers/requirements en de mobiele implementatie platformen (native / web / hybride).
Hierbij zijn de leidende vragen:

- Wat is uw ervaring en achtergrond met mobiele applicaties en implementatie platformen?
- Wat zijn de business drivers achter mobiele applicaties?
- Wat zijn technische requirements voor mobiele applicaties?
- Wanneer en waarom prefereert u een platform boven een ander platform?

Het interview verliep zeer soepel, waarbij het meer een dialoog was dan een interview sessie. Door de goede technische kennis van het implemenatie platform en de business case voor mobile apps kwam er erg veel informatie uit dit gesprek. De structuur met de 4 leidende vragen hebben we niet geheel aangehouden, maar ik het gevoel dat juist er daardoor meer informatie naar voren kwam.
Opmerking is dat Diederik vooral ervaring heeft met native en web-development en niet hybrid.

Samenvatting interview

Volgens Diederik geldt bij apps (en eigenlijk altijd) dat er een trade-off is tussen budget enerzijds en haalbaarheid en functionaliteit anderszijds.

Diederik is erg benieuwd wat WindowsPhone gaat doen, nu Microsft ook een tablet gaat uitbrengen en windows 8 een verregaande integratie heeft tussen smartphone, tablet en laptop/pc (zoals Apple dat ook heeft). Het is zelfs goed mogelijk dat hij aan einde van dit jaar ook voor WindowsPhone gaat ontwikkelen, omdat het kan zijn dat Microsoft een (groot) aandeel zou kunnen pakken binnen de corporate wereld. Dit zou verder gesteund kunnen worden door platform of choice, opgelegd vanuit een bedrijf; een andere beweging zou juist richting Bring Your Own Device (BYOD) zijn waarbij het niet te voorspellen valt wel platform de voorkeur zou krijgen.

Zelf denkt hij niet aan hybride oplossingen/frameworks, native bevalt hem goed. Het is iets meer werk, maar als je anticipeert op requirement changes en hergebruik van code dan is het slechts 1 keer een zware exercitie en vervolgens een rustiger vaarwater van sustainable engineering.
Veel (hybride) frameworks en/of libraries zijn opensource en worden onderhouden door de community. Een reden om WEL aan hybride te doen, zou gebrek aan (inhouse) programmeer kennis zijn: webdevelopers kunnen bijvoorbeeld gewoon hun eigen taal blijven gebruiken, maar toch apps creëren. PhoneGap is een oplossing als je geen tijd en/of resources (budget/kennis) hebt om je app native te maken.
Wat Diederik wel doet is native een webview gebruiken in combinatie met platform onafhankelijke data/content structuren. Voor multi-platform kan de content op deze manier herbruikt worden, wat weer scheelt in de kosten/ontwikkeltijd, verder komt dit de portability ten goede! Wat hij hiermee beschrijft is in principe een eigen hybride variant van een app.

Kosten van een app bestaan uit kennis, tijd, ontwikkeltijd, functionaliteit, en het maken van dingen. In hybride modellen komt veel functionaliteit van de browser, bij native moet je alles zelf maken.

Het allerbelangrijkste voor een app is de user-experience, de app moet het leven makkelijker of leuker maken van mensen. Hij denkt dat dit los staat van de keuze tussen native en hybride implementaties. Wel merkt hij op dat webdevelopers op hybrid apps vaken hun unieke signatuur/stempel willen drukken, wat niet altijd ten goede komt voor consistentie tussen apps voor de gebruiker (het is niet altijd direct duidelijk). Zowel developers als consumenten hebben er baat bij om de gestandaardiseerde manier te gebruiken voor bepaalde dingen (zoals een tweetfeed implementatie), zeker voor een klein scherm.

Diederik is ervan overtuigd dat een app vooral 1 ding goed moet doen, wat je app clean, lean en mean houdt. De added value van een app liggen bij de device capabilities (sensoren), maar ook het persoonlijke gebruik is iets om rekening mee te houden. Een voorbeeld is sociale media: Gebruikers hebben greep op hun content, waarbij personificatie en personaliseren key-drivers zijn. Juist die integratie van personaliseren is beter te realiseren via native code dan met browser-tech.

Security is een kort verhaal, omdat zijn apps geen gevoelige data bevatten. Maar hij is wel blij met de diepere integratie binnen het OS met social media en andere zaken. Verder gebruikt hij OAuth als protocol om contact te maken, en omdat Twitter en Apple dit zelf ontwikkelen is dit eenvoudig te integreren.

Over de sensoren op een telefoon (GPS / Camera etc) denk hij dat het beter is om deze te benaderen via native-code dan via javascript-libraries. Via de native manier heb je meer controle en mogelijkheden, terwijl de javascript-api juist een standaard manier brengt (non specifiek). Verder vraag hij zich af hoe je de (unit)testen wilt doen voor de hybride oplossingen.

Andere business drivers zijn doelgroep en voorkeur voor een platform. Verder geeft hij aan dat het waarde heeft om in de app-store te staan, een stukje prestige voor een bedrijf (dat ze meegaan met de digitalisering). Web is meer voor het algemene publiek

Interview #6

Interview informatie

Datum: 27 Juni 2012

One-liners

- In Nederland waren er destijds weinig recruitment apps waar je mee uit de voeten kon. Slechte resultaten (het werd niet gebruikt) en technische mankementen (stroeve koppeling met back-end CRM), hebben geleid dat we de stekker eruit trokken
- User experience is het allerbelangrijkste, maar ook wat ze over de web app te zeggen hebben (feedback)
- Mobiel wordt straks de belangrijkste manier van internetgebruik
- Ik zou niet voor een bepaald platform gaan, bij bv een web app met zeer sterke visuele ervaring (conceptueel moet heel top zijn, dan beter native). Eenvoudig en lage kosten dan ten alle tijden gebruik maken van een web app

Doel interview

Aan de ene kant probeert het interview mijn vermoedens over de business drivers en technische requirements te bevestigen, en aan de andere kant probeer ik via de gesprekken inzicht te krijgen hoe ze tot elkaar verhouden. Daarnaast dienen de interviews om verbanden inzichtelijk te maken tussen de drivers/requirements en de mobiele implementatie platformen (native / web / hybride).
Hierbij zijn de leidende vragen:

- Wat is uw ervaring en achtergrond met mobiele applicaties en implementatie platformen?
- Wat zijn de business drivers achter mobiele applicaties?
- Wat zijn technische requirements voor mobiele applicaties?
- Wanneer en waarom prefereert u een platform boven een ander platform?

Dit was de tweede keer dat ik Marco interviewde, dus er was vrij veel voorkennis aanwezig. Een belangrijk speerpunt was om te achterhalen wat de business drivers achter de mobiele website zijn, en hoe de keuze voor dit platform tot stand is gekomen. Hierbij wist ik van te voren dat er niet te diep in de technische requirements kon duiken. In de loop van een gesprek kreeg ik een goed idee hoe de keuze voor het web platform tot stand is gekomen.

Samenvatting interview

Het interview loopt de vier vragen in chronologische volgorde af, waarbij er soms enige overlap ontstaat.

Ervaring en achtergrond met mobiele apps

De ervaring met mobiele apps komt vanuit experimenten om de mobiele markt gebruiken binnen recruitment. Twee jaar geleden hadden ze een app laten maken (toen waren er nog weinig recruitment apps), welke zeer slechte resultaten opleverden en technische mankementen vertoonde. Dit was reden genoeg om de stekker uit de app te halen.

Daarna de focus verlegd naar gemak, de gebruiker, de koppeling tussen app en CRM, en kosten. Dit resulteerde in de mobiele website (web app) die ook meteen cross-platform was.

Business drivers
De business case achter de web app is het verkopen van vacatures, het werven van sollicitanten en informatie verspreiden. Een duidelijk driver is het gemak voor de gebruiker, maar ook de belevenis zoals ze die hebben op het web willen we ze ook geven op de mobiele variant.

Een andere belangrijke driver zijn de kosten, als Talent (HR) heb je een jaarlijks budget voor online activiteiten dus de het effectief inzetten van dit budget is erg belangrijk. Destijds bleek uit een onderzoek dat de kosten voor native apps dermate hoog lag dat dit niet binnen de mogelijkheden lag (meerdere platformen en meerdere koppelingen). We wilden gemak voor de sollicitant maar ook voor de recruiter, daarnaast moest er ook een koppeling zijn met CRM systeem.
Ook eerdere slechte ervaring en kritiek op onze app heeft een rol gehad in het keuze proces.

User experience is zeer belangrijker, maar ook alles wat men zegt over je app (imago). Bij de web app hebben we ook eerst een focusgroep laten testen; soms komen daar dingen uit waarvan je het niet direct verwacht zoals: veel snack-content (kort) maakt het saai. Dus feedback via een focusgroep en andere (interne) kanalen is cruciaal zodat je de app kunt verbeteren, hierbij ben je volledig afhankelijk van de gebruiker. Als je het echt goed wilt doen dan kun je het met een extern onderzoek laten testen, met meerdere alternatieven zodat men niet weet voor wie het onderzoek is.

De app moet een groot bereik hebben en dus veel devices ondersteunen. Ondanks dat een overgroot deel van de bezoekers iPhone users zijn, hebben we niet gekozen voor een iPhone app. Dit heeft mede te maken dat je niet weet hoe de smartphonemarkt eruit ziet over een jaar; een web app is futureproof.

Technische requirements
Deloitte heeft vrij strenge eisen voor DB en CRM, vandaar dat de koppeling naar het eigen CRM een requirement was het beslistraject. Deze koppeling wordt gemaakt door de leverancier van het CRM, zodat er geen extra partijen (en/of middleware) in het proces zitten.
Om dezelfde ervaring van de website voort te zetten in de mobiele site, is de front-end gemaakt door hetzelfde reclamebureau; deze doet trouwens ook ondersteuning, strategie, en search engine optimalisatie. Hierdoor hebben ze zo weinig mogelijk implementatie partners, en slechts 1 CRM systeem; wat de efficiëntie ten goede komt. Inhouse ontwikkeling hebben ze wel overwogen, maar woog niet op tegen de baten die de huidige partners konden geven.

Integratie tussen het sociale platform LinkedIn en de web app, maakt het ook weer een stuk prettiger voor de sollicitant. Wij kunnen door de integratie makkelijk een speciale pagina per event kunnen aanmaken in het CRM, en op deze pagina kunnen studenten/sollicitanten zich direct inschrijven, waarbij hun gegevens direct in ons CRM verwerkt is.
Een ander requirement is dat er qua sensoren weinig gebruikt werd, behalve locatie bepaling. Dat laatste kan via de gps en via de locatie-gegevens vanuit de browser(HTML5).

Security is altijd belangrijk, zeker voor het imago; of eigenlijk andersom slechte beveiliging kan zorgen voor een slecht imago. De web app wordt daarom ook nagelopen door security experts (inhouse hackers), die audits uitvoeren.

Platform preferentie

Op de vraag of mensen andere verwachten hebben van mobiele website dan native apps, denk hij van wel. Native apps moeten sneller zijn en je hebt meer vrijheid qua creatief/interactief design. Echter kun je bij web apps veel (online) technologien vaak snel en eenvoudig integreren. Hij denkt dat mobiel internet de belangrijkste manier van internetgebruik wordt, zodra het sneller wordt dan worden de mogelijkheden steeds beter voor web apps.

Native is beter voor zeer sterke visuele ervaringen (zoals games) en wanneer er veel sensors gebruikt moeten worden. Als de nadruk ligt op die ervaring, dan spelen kosten vaak een minder grote rol.

Bij de meer eenvoudige website zou hij altijd kiezen voor web apps, waarbij kosten veel lager liggen. Met hybride heeft Marco geen ervaring, en kan hij er geen uitspraken over doen.

Platform of choice / inhouse preference bleek geen invloed te hebben gehad op de platformkeuze. Een analyse tussen de alternatieven, de eerdere opgedane ervaring en kosten; gaf aan dat een web app de beste keuze was.

Interview #7

Interview informatie

Datum: 03 Juli 2012

One-liners

- Elke klant is anders, en je moet kijken naar het issue dat de klant heeft en welke implementatie platformen hier het best bij past
- Mensen zien een shift komen van website naar mobile toe, een mobile-optimized website voldoet in sommige gevallen wel, maar niet altijd. De experience in een app is vaak beter, een marktaandeel verschuift richting mobile
- Ervaring over de afgelopen jaren helpt erg met de ontwikkeltijd voor Android, en waar je extra aandacht aan moet schenken.

Doel interview

Aan de ene kant probeert het interview mijn vermoedens over de business drivers en technische requirements te bevestigen, en aan de andere kant probeer ik via de gesprekken inzicht te krijgen hoe ze tot elkaar verhouden. Daarnaast dienen de interviews om verbanden inzichtelijk te maken tussen de drivers/requirements en de mobiele implementatie platformen (native / web / hybride).
Hierbij zijn de leidende vragen:

- Wat is uw ervaring en achtergrond met mobiele applicaties en implementatie platformen?
- Wat zijn de business drivers achter mobiele applicaties?
- Wat zijn technische requirements voor mobiele applicaties?
- Wanneer en waarom prefereert u een platform boven een ander platform?

Het was een zeer prettig gesprek, waarbij er veel vergelijkingen en technische zaken aan bod kwamen. Hiermee werden de vragen zeer concreet beantwoord en geeft het goede input voor mijn thesis.

Samenvatting interview

Het interview loopt de vier vragen in chronologische volgorde af, waarbij er soms enige overlap aanwezig is.

Ervaring en achtergrond met mobiele apps

Joshua heeft al heel wat jaren ervaring met mobiele applicaties. Iets hem opviel is dat het meeste werk zit in het maken van views, het is niet echt moeilijk maar het kost veel tijd. Daarnaast is DOM-tree manipulatie erg langzaam in de webview. Om een oplossing hiervoor te bieden hebben ze een eigen platform gemaakt: Spicy. Hierin wordt alle logica, interactie en opbouw native afgehandeld en op de weergave op het laatst gerenderd in een webview.
Verder voornamelijk ontwikkelt voor iOS en Android, en in minder mate met Windows Phone en BlackBerry. Blackberry is vooral interessant voor als de doelgroep tussen de 15 en 18 jaar is. WindowsPhone heeft de potentie om groter te worden, zeker omdat Microsoft een grote slagkracht heeft om dingen er doorheen te drukken.

Business drivers

De belangrijkste reden voor bedrijven om mobiele apps te willen, is dat ze een plekje willen hebben in de broekzak van de consument, het feit dat je het altijd bij je hebt. Je moet zichtbaar zijn in de marketstores (appstore/play). Er treedt een shift op waarbij een mobile-optimized site niet meer voldoende is, de experience op een app is vaak beter en de markt steeds groter. Verder wordt er gekeken naar wat de concurrentie doet, en als die een app hebben dan moet jij ook een app hebben.
De waarde van een appstore is belangrijk omdat iPhone apps nog nieuwswaarde hebben, terwijl een nieuwe mobiele site niet echt nieuwswaardig is. Een goede mobiele website hoeft niet goedkoper te zijn dan een goede app.

Kwaliteit van een app definieert hij aan de hand van wat de user meekrijgt (user experience en expectation). Het verschil tussen het web en een app is dat een app in de appstore vindbaar is en dat het gebruik kan maken van de sensors (GPS/Camera/kompas etc) van de telefoon. Een native app zal over het algemeen sneller aanvoelen dan een web app.

Als je kijkt naar de algehele integratie (vendor lock-in) tussen de aanbieders dan zie je dat Apple hier heel sterk in is, zodra je meerdere apparaten van Apple hebt(iPhone/iPad/iPod/iMac) dan werkt de lock-in zeer sterk. Microsoft zal deze truc ook toepassen met Windows8 waarbij de integratie tussen PC + Tablet + telefoon + Xbox uitgebuit zal worden. De insteek van Google is heel anders, in plaats van een closed eco-system bouwen zij voort op het opensource principe en cloudbased oplossingen.
Verder kijken ze nog even de kat uit de boom qua WindowsPhone8 en Android tablets.

Een belangrijke andere driver is de support op apps, hierbij doen zij niet aan licenties maar bieden ze continue support aan op de producten. Alleen change-requests worden extra belast.

Technische requirements

Security probeert hij te matchen met privacy. Developers moeten de best-practices zoals SSL en databaseframeworks gebruiken om security op orde te hebben. Verder staat er zoveel mogelijk informatie op de servers en niet op het device. Tracking is tot een minimum beperkt en gevoelige data nooit plaintext over de lijn of ergens opgeslagen. MDM is een leuke feature, maar je moet er niet afhankelijk voor zijn want je kunt het er relatief eenvoudig vanaf halen.

Voor de device capabilities/sensors test Peperzaken de snelheid, om te zien of deze geschikt zijn voor display via een webview of dat alles native moet. De info ophalen van de camera/gps gaat eigenlijk altijd native.

Qua restricties die die appStore van Apple heeft in combinatie met strak review proces, is af en toe lastig voor de developer maar dit komt ten goede van de kwaliteit en de ervaring voor de gebruiker.

Opensource is vaak een onderwerp bij overheid aanvragen, waarbij ze eigenlijk de broncode willen hebben (dit is niet geheel opensource zoals Joshua het kent, dat zijn de open projecten gehost op github en andere).

De verschillen tussen hoog- en laagvolume apps zijn dat hoogvolume apps (zoals nu.nl app) vrij sterke pieken kent en door de grote gebruikersgroep support een stuk belangrijker is. Als er 0.1% van de gebruikers een issue heeft met een hoogvolume app dan is daar meteen veel en intensief werk meegemoeit qua rapportages, afhandeling en user feedback.

Platform preferentie
Qua OS preferentie staat iOS ver bovenaan, daarna WindowsPhone en Android. De reden dat Android hekkensluiter is omdat er geen uniforme of consistente beleving is voor de gebruiker. Het is technisch gezien een heel mooi platform, maar op tragere android phones wil het nog weleens haperen en dat komt de experience/kwaliteit niet ten goede. WindowsPhone heeft een super interface, maar heeft met de aankondiging van WP8 laten weten dat het niet op de huidige smartphones draait, hiermee laat je de early adapters behoorlijk in de kou staan. Qua user experience en volwassenheid zie je dat Apple toch voorop loopt met iOS.

Verder benadrukt Joshua dat je eerst moet kijken naar het issue van de klant; niet alles heeft een app nodig. Een goede mobiele website volstaan prima als je contact formulier en blogposts wilt laten zien, hierbij gebruik je namelijk geen personalisatie van de content en/of sensors van de phone.
Als er dan een app gebouwd wordt, kijk dan kritisch welke stukken je native gaat doen en welke niet. Naarmate je meer platformen moet ondersteunen neigt je richting hybride oplossingen. Als er slechts 1 platform ondersteund hoeft te worden dan is de keuze voor native voor de hand liggend.

Elke klant is anders, en sommige hebben zelf een sterke voorkeur over het implementatie platform.

Wat je ziet is dat zaken die goed werken op het web en daarna goed geport worden naar mobile apps, dat dit goed werkt! Als je kijkt naar dingen die alleen goed doen qua mobile zoals Angry Birds die gebruik je alleen in de wachtkamer (tijddoder).

Interview #8

Interview informatie

Datum: 17 Juli 2012

One-liners

- Vanuit functioneel perspectief kun je stellen dat dat mobiele apps veel beter past bij de informatiebehoefte van de gebruiker, dan bij een mobiele website
- De belangrijkste business driver voor bedrijven is om de behoefte aan contact momenten met de gebruiker te vermeerderen via het mobiele kanaal
- Kwaliteit van een app wordt bepaald door het feit of de gebruiker de app blijft gebruiken

Doel interview

Aan de ene kant probeert het interview mijn vermoedens over de business drivers en technische requirements te bevestigen, en aan de andere kant probeer ik via de gesprekken inzicht te krijgen hoe ze tot elkaar verhouden. Daarnaast dienen de interviews om verbanden inzichtelijk te maken tussen de drivers/requirements en de mobiele implementatie platformen (native / web / hybride).
Hierbij zijn de leidende vragen:

- Wat is uw ervaring en achtergrond met mobiele applicaties en implementatie platformen?
- Wat zijn de business drivers achter mobiele applicaties?
- Wat zijn technische requirements voor mobiele applicaties?
- Wanneer en waarom prefereert u een platform boven een ander platform?

Ik kon direct merken dan Wim al lang in deze sector zit, wat de diepgang van de antwoorden alleen maar ten goede kwam. Verder gaf hij vaak concrete voorbeelden om de antwoorden te illustreren.

Samenvatting interview

Het interview loopt de vier vragen in chronologische volgorde af, waarbij er soms enige overlap aanwezig is.

Ervaring en achtergrond met mobiele apps

Binnen Service2Media is er vanaf het begin gekozen voor het applicatie paradigma, waarbij de focus ligt op mobiel internet. Vanuit functioneel perspectief kun je stellen dat dat mobiele apps veel beter past bij de behoefte van de gebruiker, dan bij een mobiele website. Als je mobiel bent moet je snel en betrouwbaar, binnen een paar klikken, de gewenste informatie kunnen bereiken.

Wim geeft aan dat hoewel het M2Active platform volgens mijn definitie een hybride vorm is, het toch meer weg heeft van een native app. De werking van het M2Active implementatie platform werkt door dat er LUA-script omgezet wordt naar byte-code, die vervolgens gepackaged worden met de native runtimes. De smartphone executeert vervolgens, middels een virtual machine, de LUA-

code. Hiermee doelt hij op het feit dat alle interactie plaats vind via de API's van het OS, waarbij het OS het ook als een native app ziet. M2Active maakt niet gebruikt van een (chromeless) browser voor rendering.

Business drivers

De belangrijkste business driver voor bedrijven is om de behoefte aan contact momenten met de gebruiker te vermeerderen via het mobiele kanaal. In Nederland is het netwerk zo ver gevorderd dat het normaal is om continue online te zijn via je smartphone. Voor de gebruiker is er een behoefte om informatie/services te verkrijgen op elk moment van de dag, geprefereerd via een app.
Bedrijven proberen hierop in te springen, ook omdat ze bang zijn dat ze de boot anders missen.
Ondanks dat er veel vraag is, denkt Wim dat er nog niet veel sluitende business-cases zijn waarbij de revenue voldoende is om de kosten te dekken. Wel ziet hij een shift dat de nieuwe mobiele apps steeds professioneler aangepakt worden en een groter budget krijgt. Eerder was het vaak marketing die experimenteerde, maar nu mobiel steeds meer mainstream wordt zie je dat nu vaker vanuit IT komt. Dit zorgt ervoor dat apps betere aansturing krijgt met duidelijke doelen en beslispunten, maar ook opdrachten die uiteindelijk leiden tot continuïteit in dienstverlening. Het wordt een permanent kanaal dat men steeds verder ontwikkeld en via updates en regelmatige aanpassingen zodat de gebruiker getriggerd blijft.

De visibility op de mobiele markt zorgt voor toenemend gebruik van de apps, echter trekt dit mensen weg van de website. Een website is een belangrijk communicatiekanaal waarbij er veel meer informatie overgebracht kan worden aan de gebruiker. Een app moet ook (specifieke) informatie kunnen aanbieden, al zal het nooit een website 1-op-1 kunnen overzetten naar een app. De beeldvorming om gebruik te maken van het mobiele kanaal als dienstverlener of brand is nog steeds in ontwikkeling en nog geen uitgekristalliseerd proces.

De smartphone-markt is een snel groeiende markt, zeker als je dat vergelijkt met de featurephone. Daarin is Android de marktleider aangezien die meer dan 50% marktaandeel heeft. Apple heeft destijds de markt opengebroken met de touchscreen-gebaseerde telefoons, wat een knap staaltje engineering en marketing is geweest. Blackberry wordt ook ondersteund door het M2Active platform, maar WindowsPhone niet. Dit heeft te maken dat er geen JAVA virtual machine op draait, wat een gigantische trendbreuk was ten opzichte van WindowsMobile. De native app development omgeving van Windows Mobile is niet compatibel met die van Windows Phone 7/8. Service2media is bezig met het ondersteunen van Windows Phone 8 in haar M2Active App Lifecycle Platform. Wel verwacht hij dat Microsoft met Nokia een kritische massa zou kunnen bereiken (met WP8).

De spelers op de smartphone-markt proberen zich op hun eigen manier te onderscheiden. Dit kan door visie op gebieden als distributie, presentatie en openheid van het platform. Hoewel de smartphone-mark vrij divers is, is het wel mogelijk om stereotypes te verzinnen bij de verschillende OS'en. Verder zie je dat de zakelijke markt zich steeds comfortabeler voelt met iPhones en Android. Wim verwacht dat BYOD (bring your own device) vrij normaal gaat worden in de toekomst.

Technische requirements

Je moet niet meer specifiek native apps willen bouwen voor iOS en/of Android. Dit is gewoon veel te duur, kost veel tijd, en je moet de (specifieke) expertise in huis halen. Daarom in de aanpak van

build-once-deploy-to-many beter. Binnen het M2Active platform ontwikkelen ze de app via de SCRUM methode, waarbij app gradueel uitgebouwd wordt. De opvolgende iteraties worden op elk platform getest en gevalideerd. De app heeft 1 broncode, maar kan wel flexibel omgaan met specifieke interacties op bepaalde platformen. Een groot deel van de (platform specifieke) variaties kunnen ze onderbrengen in de native runtimes. Dit is belangrijk omdat elk platform zijn gebruikers een beetje conditioneert door hun eigen guidelines voor de interface gebruikt.

Platform preferentie

Hoewel Wim, vanwege het M2Active platform, geen onderscheidt maakt tussen hybrid en native; zou hij wel handvaten kunnen geven welk platform er bij een klant past. Als er een keuze gemaakt moeten worden, dan zou het beginnen met de vraag wat de klant wil bereiken met het mobiele kanaal. Apps hebben het voordeel dat ze gevonden kunnen worden in de app-store maar ook op je telefoon. Met de vele apps die er zijn, is het belangrijk dat je ook herkenbaar bent als app en/of als brand; tevens moet je continue aanwezig proberen te zijn. Een ander voordeel van apps zijn dat ze offline kunnen werken.

Niettemin zijn er genoeg opties voor web-apps, waarbij er voornamelijk info gepresenteerd wordt.

Security is wel een punt dat hij nog wil aanstippen, vanaf het web denkt iedereen wel twee keer na voordat hij iets installeert; echter iedereen installeert apps zonder nadenken. Bij ongecontroleerde stores zoals Android, is de kans op malware en trojans een stuk groter.

Een andere trend die hij ziet is een convergentie die optreedt tussen webtechnologie, cloudplatformen en het applicatie paradigma. Een voorbeeld hiervan is chrome, die web apps en desktop combineert. Voor de gebruiker draait hij een normale app, maar doet dit lokaal in een sandbox terwijl alle data via webtechnologie uitgewisseld wordt met de cloud.

Kwaliteit van een app wordt bepaald door gebruik en het blijven gebruiken van een app. Kijk naar de doelen, en stel vast of jouw klant/gebruiker de applicatie benut, op die manier dat jij voor ogen had. Haal onbenutte functies en langzame elementen eruit en probeer de kwaliteit te verhogen door meer personalisatie en gebruik te stimuleren.

Interview #9

Interview Information

Date: 17th of July, 2012

One-liners

- The ultimate answer is depends on the services and setup of platform. It will not be limited by device of render engine, but by the content you have and the way you can distribute it.
- At an early stage it became clear that not all functionality of desktop programs could be transferred to the iPad.
- For end-user it should not matter which implementation platform is used. Just use the platform that makes the most sense for the business case you are looking at

Purpose interview

The purpose of the interview is double-sided. Firstly it serves to validate the criteria behind mobile apps, and secondly to discover possible relations between these criteria and mobile implementation platforms (native / web / hybrid).
The leading of the interview questions are:

- What is your experience with mobile applications and implementation platforms?
- What are business drivers for mobile applications?
- What are implementation requirements for mobile applications?
- When and why is an implementation platform preferred above another platform?

The interview gave good answers on my main questions. Rowan has a clear vision that in the end it is all about the experience for the user, no matter which implementation platform is used. In the meantime he elaborated on how to do achieve this for the different platforms and which criteria he uses for this.

Summary interview

The interview will be summarized around the four main questions, where some overlap can occur.

Experience with mobile apps and implementation platforms

Before the iPhone came out, Rowan was already writing Mac software for a long time. They were already very focused on, and maintaining the principles of the HIG (Human Interface Guideline). When iOS (iPhone) came out the HIG did not change drastically, yet it did was more specific for the new technology.
At an early stage it became clear that not all functionality of desktop programs could be transferred to the iPad. They took the famous 80-20 rule (Pareto's principle[20]), and bent it to a 60-40 rule. There is more focus and features for that 60%, which worked better for the apps (especially if they were complementary to desktop apps).

[20] http://en.wikipedia.org/wiki/Pareto_principle

When looking to functionality, he quotes Antoine de Saint Exupéry[21]: “Engineering isn’t done when you can’t add something else; it is done when you can’t remove anything else”. Referring to the fact that developing an app involves critical thinking of implementing only the features you really need. The flipside is that the more you provide functionality that is (initially) hidden for the user, the more powerful it will be. An example is downloading data, in the background only when WiFi is available, so the user has an instant update without using the 3G connection.

His experience with Android is mostly related to design work. Android was a big mess in terms on how to design an app, as there were a lot of things floating around. Google improved a lot with the latest version where the documentation really propelled the platform forwards; it gives the developer a coherent way to design apps for phone and tablet. This can leverage the advantage of porting an Android phone-app to a tablet without many issues. While on Apple the iPhone and iPad require a very different design for each device.

When it comes down to programming, Rowan has extensive Objective-C experience but also used to program in JAVA. An important thing is not to think in JAVA programming but in Android programming, where the developer utilizes the built-in features that Android provides, like the a-sync tasks. It will lead to more powerful code, as it requires less code and automatically return the proper object while using features like caching. He thinks that the Android has an advantage over iOS, especially when looking to the views. iOS uses a more pixel-to-pixel implementation, while Android has more HTML-like approach with blocks, variable width and columns. This is a major advantage when porting an app to a different screensizes, like from phone to tablet.
When talking about screensizes, he does not necessarily thinks that it is a disadvantage for Android that is has to support many screensizes. It can be an advantage and flexible, as long as the developer plans for using the lay-out system in a proper way.

Business Drivers
When asking about the different implementation platforms (native, web, and hybrid) he immediately approached this from the perspective of the end-user. Rowan uses a totally agnostic approach; the important part is the services and setup of the platform. For end-user it should not matter which implementation platform is used. Regardless of native or HTML5, at the core is always about pushing some data through some request at some server. Just use the platform that makes the most sense for the business case you are looking at.
Each approach has it pros and cons, so I usually ask about a few things. To begin with if there is a need for offline access. Native apps handle offline access, and what you can do with it, a lot better. Android and iOS (and other platforms as well) have good mechanisms for caching, databases, syncing etc.
The choice of rendering (webview vs native) depends heavily on what you are displaying, and the speed it has to have. It should fit to get the performance and responsiveness it is designed to have. To illustrate this he refers to the LinkedIn app developers who have made this case: if you write a hybrid app, there are things that native does really well like scrolling in list; but everything else HTML does better

[21] “It seems that perfection is attained not when there is nothing more to add, but when there is nothing more to remove”

Another criteria would be if the app uses OpenGL or is an game, then native would be the wise choice.
When looking at business apps, it is always a question if you need an app or if a mobile optimize site is more appropriate. A lot of these apps do not require native functionality, and most could be well done on a site.
When using the sensors (like GPS and camera), but also the other things like caching, offline access, communication between apps on the device and opening certain links by your app; then consider native.

Personalization is also a factor that has significant impact on the user. When looking at social media, it is obvious that everybody should get a personalized dataset instead of everybody the same. The relevance increases for the bonding to the user. If this personalization is already present on the web, then it can make them strong in apps as well.

About visibility in the app-stores he is less enthusiastic. In the beginning it was simple, all apps were curated and all of them were fantastic. In the present app-stores it is hard to find the app that you already know to look for. It is important that the app-store is a trusted source that is properly handling the money-transfers, and supports in-app purchases.
About targetgroups (and stereotypes with certain OS'es) Rowan says that you cannot target specific groups, as mobile is everywhere and very ubiquitous.

Technical requirements
Rowan thinks there is no limit for the web-platforms, on the matter what you can do with it. The current trend is to use JavaScript libraries and frameworks. Although he thinks that the future of apps will have 2 things: storing more data in the cloud/servers; and more mature ways of programming and adapters for "plug and play" functionality.

A sidetrack was the question about total integration of hardware platforms; the phone combined with TV, laptop, desktop, mp3-players, etc. This creates a vendor lock-in, which is a powerful tool of customer binding. The current closed eco-systems by Apple, Microsoft and Google do this pretty well; but Rowan thinks that a service that just runs on any platform is more powerful then the closed eco-systems. Examples are Netflix and Spotify.

Requirements often come forth out of business perspective instead of the consumer. Examples are analytics, encryption, and tokenization of sorts.

Platform preference
Rowan likes to recommend the hybrid model, as examples have shown that it is viable and users respond well to it. The problem is that is does require some trade-offs in capabilities and design implications. What the customer wants and what is doable, usually results in native based apps. Yet he thinks that in a year or 2 the hybrid model will have more capabilities so that it could be the preference of/for customers.

The platform choices boil down to the trade-offs on what you want, what it can do, and what has/hasn't been tried yet. If any platform can do it, then take a look at how your app differentiates

itself: what is the unique selling point? In the end it also comes down to how much many you can spend on it.

The ultimate answer is depends on the services and setup of platform. It will not be limited by device of render engine, but by the content you have and the way you can distribute it.

14 APPENDIX E: CODING INTERVIEWS

Each of the interviews was compared with this list of criteria:

1) **User Interface**: Criteria that have requirements on the user interface part of an app
   a) Responsiveness: How snappy is the interface? Is this level acceptable?
   b) Animations and sophistication: The complexity of the UI, use of openGL or webGL
   c) Screensize: Does the app require a big screen?
2) **Offline access**: Does the app require internet, or should it stand-alone?
3) **Device capabilities**: Are there requirements for the device?
   a) Sensors (GPS / Camara / etc)
   b) Special needs (ruggedized)
4) **Visibility**: Visibility in stores/markets
5) **Multiplatform**: Does the app have to run on multiple platforms?
   a) Portability: The portability of apps between the platforms
6) **App development**
   a) Skillset required by developer
   b) Tooling / IDE
   c) In-house vs outsourcing
   d) Open source
   e) Updatability
7) **Business Model**
   a) Business case: The proposition, the ROI
   b) Budget: The overall budget
   c) Support & SLA: Maintainability of the app
   d) Purpose / goal of app
8) **Targetgroup**: Who is the targetgroup, and how is this defined?
   a) Heterogeneity: How
   b) BYOD (bring your own device)
   c) High-end market? Low-end market?
9) **Security**: What security features are required?
   a) Mobile Device Management
   b) Remote Wipe
   c) Encryption of data
10) **Platform features**: General features for Native/Web/Hybrid platforms)
   a) Ease of back-office integration
   b) Legal issues (patents)
   c) Licenses / Fees / TOS / InApp sells
   d) Content owning
11) **Future proof**
12) **Preference**

An overview can be found in the table below

FREQUENCY / INTERVIEWEE	#1	#2	#3	#4	#5	#6	#7	#8	#9
1. **User Interface**	2	1	3	1	1	2	2	1	2
a. Responsiveness			x			x	x	x	x
b. Animations and sophistication:	x		x	x	x	x	x		
c. Screensize	x	x	x						x
2. **Offline access**:					1		1	1	1
3. **Device capabilities**:	1	2	1	1	1	1	2	1	1
a. Sensors	x	x	x	x	x	x	x	x	x
b. Special needs		x					x		
4. **Visibility**	1		1		1	1	1	1	1
5. **Multiplatform**		1	1		1	1	1	1	1
a. Portability			x		x	x			
6. App development	3	2	4	3	2	2	3	2	1
a. Skillset required by developer	x		x	x	x			x	x
b. Tooling / IDE	x								
c. In-house vs outsourcing		x	x	x		x	x	x	
d. Open source	x		x	x	x		x		
e. Updatability		x	x			x	x		
7. Business Model	3	4	2	2	1	1	3	2	1
a. Business case		x	x	x			x	x	x
b. Budget	x	x			x	x	x	x	
c. Support & SLA	x	x		x			x		
d. Purpose / goal of app		x	x						
8. **Targetgroup**	1	1	1	2	1	1	1	1	
a. Heterogeneity		x	x	x		x			
b. BYOD (bring your own device)					x				
c. Hi-end or low-end market?	x			x					
9. **Security**	1	2	1	1		1	2	1	
a. Mobile Device Management		x					x		
b. Encryption of data		x	x	x			x		
10. Platform features		1	1	1		1	1		
a. Ease of back-office integration		x				x			
b. Legal issues (patents)									
c. Licenses / Fees / TOS			x						
d. Distribution				x			x		
11. Future proof		1	1	1	1			1	1
12. Preference	1	1	1	1	1	1	1	1	1

Interview #1

Explicit relations (criteria -> preference)

- User Interface: Android has many devices and screensizes that is likes to support
- Offline access
- Device capabilities: Sensors = native. Hybrid doesn't not work (consistent)
- Visibility: You need to be in the store
- Multiplatform
- App development:
    - Hybrid takes longer due to poor documentation or unexpected behavior, longer implementation rout
    - Open source: not of importance. Better closed source for business
    - Skills and Tooling: is less important. Recruit right from university and do crash course. Limitations are on the device, not the developer.
- Business model:
    - Budget: Low = simple html5/hybrid site. Native = expensive
    - Support & SLA = important for B2C
- Targetgroup: What does the user expect?
- Security: Important, but depends on app. Trade-off: security vs ease of use
- Platform features
- Future proof
- Preference: iPhone/iPad > Android

OTHER:

- High quality = Native
- Release cycle is getting better

Interview #2

Explicit relations (criteria -> preference)

- User Interface: Touch interface has advantages in administrative tasks
- Offline access:
- Device capabilities:
    - Sensors: GPS and Camera have to be used
    - Special needs: Ruggedized devices, special hardware or consumer hardware. Consumer hardware with protector was enough (cost-benefits)
- Visibility:
- Multiplatform: App should be cross-platform, although the focus was Android
- App development:
    - Outsource: Not enough inhouse knowledge
    - Updatability: Code changes and distribution should be fast, preferably in own control
- Business model:
    - Business case: The added value/proposition of the app
    - Budget: What are the costs? How is the ROI?
    - Support & SLA: Continuous support was a requirement at outsourcing
    - Purpose: Improve efficiency in the process, shortening travel time.
- Targetgroup: Unclear what device to support.
- Security:
    - Mobile Device Management: The customer is looking into possibilities of MDM where remote wipe is one of the main criteria
    - Encryption: Secure data management is important
- Platform features
    - Back office integration: 1 of requirements on partner selection
- Future proof: Not native to maximize future application. Technology goes so rapid
- Preference: Preferred platform of choice is Android, due to its open character, many devices, and no vendor lock-in

OTHER:

- Main concern was native vs hybrid
- Tablet vs Smartphone -> Favored the smartphone for size, screensize, battery life, speed and OS

Interview #3

Explicit relations (criteria -> preference)

- User Interface:
  - Responsiveness: Hybrids don't have the snappyness yet that native does
  - Sophistication: iOS just looks better, Android is less strict due to support of wide array of devices. iOS > Android
  - Screensize: Fixed screens for iPhone works great, difficulty at Android. iPad offers bigger screen which can be used
- Offline access
- Device capabilities:
  - Sensors: Usage of camera for scanning invoices
- Visibility: Visibility in the appstore has had user-value, they found us there.
- Multiplatform:
  - Portability: Android is more expensive to develop for, needs
- App development:
  - Skillset developer: Difficult to attract (good) Android programmers
  - Inhouse: Outsourcing creates dependency, which is unwanted.
  - Opensource: Ruby = opensource, but closed source is used for own app
  - Updatability: Web apps have the advantage of instant updates, without required updates. Hybrid/Native apps can have dependencies on (old) API's.
- Business model:
  - Business case: Get new customers through mobile channel
  - Goal: Bring invoices to mobile
- Targetgroup:
  - Heterogeneity: Targets business users, these have mostly iPhones. Although Android request are also coming in.
- Security:
  - Encryption: All best practices are used: SSL, oAuth, extra pincodes. Most data on server. Preventive scans
- Platform features
- Future proof: Hybrid (and web) solutions do have potential to be the future when browser and webtech improves (CSS)
- Preference: Develops native applications
  - Native = quality + marketing value of appstore
  - Web = uniform method for monitoring, deployment and workflow
  - Hybrid = a little les quality but all the advantages

OTHER:

- Mobile market has lots of potential (future)
- Positive feedback from users and media was a driver to continue development
- Script-2-native hybrids (like appcelerator), why not native if you have to put OS-specific code in there
- Rather have 1 app that is 100%, then less costs and have 80% quality

Interview #4

Explicit relations (criteria -> preference)

- User Interface:
  - Sophistication: Native user interface experience has the most quality
- Offline access
- Device capabilities:
  - Sensors: Usage of camera and GPS
- Visibility:
- Multiplatform
- App development:
  - Skillset: Each programmer has freedom to experiment and use own technology. Still difficult to attract specialized developers.
  - Inhouse: Outsourcing with hybrid did not work, fast communication and prototyping is essential.
  - Open source: This creates dependencies on communities. There is nothing wrong with closed source if maintained (by a company) .
- Business model:
  - Business case: Monetization for app is done by campaigns, although the user will not notice this
  - Support: Direct feedback and dedicated support for apps is an important pillar.
- Targetgroup:
  - Heterogeneity: Targets specific usergroups, but has to support every major platform
  - Hi-end: iPhone users; budget = Android
- Security:
  - MDM/Encryption: Native has most to offer for security.
- Platform features:
  - Distribution: Worldwide distribution
- Future proof: Native will get more
- Preference: order in volume and maturity
  - iOS: most mature
  - Android: Biggest player, most fragmented, no app management in Play
  - WindowsPhone: Has the potential to become a very big player
  - Blackberry: declining, expected to disappear

OTHER:

- Middleware is not interesting. Native is the way to go!
- Fragmentation of Android is a big problem (almost nobody on newest version)

Interview #5

Explicit relations (criteria -> preference)

- User Interface:
  - Sophistication: App should make it easier for the user, no matter the platform. Webdevelopers do have the tendency to put their stamp/mark on the app.
- Offline access
- Device capabilities:
  - Sensors: They are the added value for the mobile platform; though best approached on a native way. Javascript api = more generic (standard)
- Visibility: It matters to be found in the appstore, much as branding and prestige
- Multiplatform:
  - Portability is enhanced by using platform-independent datastructures and using webviews
- App development:
  - Skillset: Lack of native skills could drive towards hybrid framework
  - Opensource: creates dependencies which
- Business model:
  - Budget: Lower budgets favor web and hybrid solutions
- Targetgroup:
  - BYOD: This could be a driver for corporate organizations (no platform of choice)
- Security:
- Platform features
- Future proof
- Preference: Native >> hybrid.

OTHER:

- App is trade-off between Budget ←→feasibility & functionality
- Costs are knowledge, development time, and functionality
- Browser gives a lot of functionality for free (in native you need to program everything)
- App should be clean, lean and mean
- Personalization and personification are key for app

Interview #6

Explicit relations (criteria -> preference)

- User Interface:
    - Responsiveness:
    - Sophistication: Easy integration with other web-bases tech
- Offline access
- Device capabilities:
    - Sensors: None were used
- Visibility: Feedback comes in many forms, do you really need to be in the store?
- Multiplatform:
    - Portability: Website is directly cross-platform
- App development:
    - Outsource: Costs outweighs the advantage of inhouse development.
    - Updatability: The website can be instantly updated
- Business model:
    - Budget: There is no ROI, but targets and budget is restricted
- Targetgroup:
    - Heterogeneity: With a mobile website you can serve all smartphones
- Security: The app is checked and audited
- Platform features:
    - Backoffice integration: major decision for outsource partner
- Future proof: Mobile
- Preference: Mobile website

OTHER:

- Previous negative experience with native influenced the choice for mobile website heavily
- Native is better for very visual apps (like games), and when sensors are used
- Simpler app, should consider a mobile website/web app or hybrid due to costs and compatability.

Interview #7

Explicit relations (criteria -> preference)

- User Interface:
    - Responsiveness: Domtree manipulatie in webview == expensive
    - Sophistication: Native can employ faster views
- Offline access:
- Device capabilities:
    - Sensors: Native apps are more suitable to deal with sensors. They test the performance for rendering the output in Native or Webview mode.
    - Special needs: Some environments needs certification, and not every (consumer) product have those. These limit the potential platform and possibilities for apps.
- Visibility: Required at marketstore, apps have news value
- Multiplatform: More platform favors hybrid
- App development:
    - Inhouse: Even though developers are hard to find, they develop inhouse.
    - Opensource: Governmental bodies, do require this but only want to have the source-code (and keep it private) which is actually not opensource.
    - Updatability: Apple review process is annoying at times, but improves the quality, look&feel, and experience for the user
- Business model:
    - Business case: Companies want to have a place at the customer
    - Budget: Mobile sites aren't always cheaper than mobile apps
    - Support & SLA: Support (even to customers of your client) is important and should be continuous. It is a vital element of the app
- Targetgroup:
- Security:
    - MDM: A potent feature, but can be easily worked around (factory reset).
    - Encryption of data: This is matched by the level of privacy needed.
- Platform features:
    - Distribution: Appstores and markets are good distribution platforms, although sometimes the review process can slow it down.
- Future proof:
- Preference: Native/Hybrid combination.

OTHER:

- Employs an own hybrid framework, where the logic = naitve, but the view = webview
- Mobile-optimized sites aren't cutting it anymore, the user requires an experience that is better suited through apps
- High volume apps have a stronger focus on support, as a small bug will affect many more phones/users then with low volume apps.
- If you support only 1 platform, go native
- A good mobile website is fine for non-personalized content and no use of sensors.

Interview #8

Explicit relations (criteria -> preference)

- User Interface:
- Offline access:
- Device capabilities:
    - Sensors can be a reason to go native, but in the framework it is captured via native runtimes
- Visibility:
    - Presence in appstores is essential for organizations.
- Multiplatform:
    - Portability: Code once, deploy to many
- App development:
    - Skillset: Too expensive to have specialist developers on each platform
    - Inhouse: They develop the framework inhouse
- Business model:
    - Business case: almost no proposition that cover the costs
    - Budget: Hybrid is cheaper than native
- Targetgroup: Not only how to target your users, but also let the users keep using it
- Security: Installing something from the internet is not done (Trojans etc) but installing apps is a no brainer
- Platform features:
- Future proof: Native will become too expensive, hybrid and web will dominate
- Preference: Hybrid

OTHER:

- From a functional view, apps are better suited then a mobile website in the mobile channel
- Mobile apps are getting more professional, with bigger budget, clearer goals and milestones
- Websites can never be ported 1-on-1 to mobiles
- Personalize the app for better user commitment

Interview #9

Explicit relations (criteria -> preference)

- User Interface:
    - Responsiveness: Renderer in native: scrolling in lists, everything else use HTML
    - Screensizes: Android has good portability for phones and tablets (with same view), like an HTML page
- Offline access: Native handles offline access better
- Device capabilities:
    - Sensors: If you don't need sensors, consider web/hybrid apps
- Visibility: Even when you are in a store, you still need to be found
- Multiplatform:
    - Android has good portability from phone to tablet
- App development:
- Business model:
    - Business case: a lot of requirement are business sided and not focused on the consumer (like analytics and tokenization)
- Targetgroup:
- Security:
- Platform features
- Future proof: The future will most likely have services that run on all devices instead of the vendor lockin
- Preference: Hybrid

OTHER:

- Apps aren't about what features/functionality you can add, but which one you can remove/hide.
- Use an agnostic approach to which services and setup of a platform you are going to use.
- Business sites should ask if a mobile-optimized site is more appropriate

Personalization of content has a significant impact on the user.

15 APPENDIX F: PROTOTYPE SAW

A screenshot of the simple additive weighting based prototype, worked out in Microsoft Excel. It requires the input in the blue colored columns. A screenshot of the sheet can be seen in the image below.

Adjust these

		Criteria	Enabled	Criteria weight	Description		Weight native	Weight Hybrid	Weight Web
Major criteria	C1	Offline Access	0	2	App should be usable without (internet) connection	C1	2	1	-1
	C2	Device Sensors	0	2	App uses device sensors like GPS, camera, compas, etc	C2	2	1	-1
	C3	Market Distribution	0	2	App visibility in market store, using that distribution channel	C3	2	2	0
	C4	Multi Platform	0	3	App should be deployed on multiple mobile OS'en	C4	0	1	2
	C5	Native Mechanisms	0	2	App employs native functionality like push notifications, native ui	C5	2	1	-1
	C6	Info channel	0	2	App only displays text and multimedia	C6	0	1	2
Normal criteria	C7	Cost	0	1	Total costs of app; in other words the available budget	C7	0	2	2
	C8	Updatability	0	1	Instant update of program / content	C8	0	1	2
	C9	Personalization	0	1	Personalized & Dynamic content based on location/preference etc?	C9	2	1	0
	C10	Responsiveness	0	1	Responsiveness of the user interface (for interactions)	C10	2	1	0
	C11	Complex / native UI	0	1	Complexness and/or usage of native UI	C11	2	0	0
	C12	Screensize	0	1	Support for multiple screensizes	C12	0	1	2
	C13	Heterogeneity devices	0	1	Heterogeneity of devices, even within 1 OS (BYOD)	C13	0	2	2
	C14	High expectations	0	1	Serve high-end users with high expectations	C14	2	1	0
	C15	User coverage	0	1	Reach all users, no matter the device	C15	0	1	2
	C16	Encrypted data	0	1	Storing encrypted date on device/cloud	C16	2	1	1
	C17	Market fee	0	1	3rd party fees for monetization of apps and in-app transactions	C17	0	0	2
	C18	Regulations	0	1	Impact of restrictions (Terms of Service) from OS vendors	C18	0	0	2
						Total weight	18	18	16

Outcome	Native	Hybrid	Web
Platform Score	0	0	0
Preferred platform:	**No decision**		

Legend & Explanation

Enabled: Did the customer mention this criteria? 0 = No, 1 = Yes
Criteria weight: How strong is the preference for this criteria. Scale is from 1 till 3 (higher = stronger)
Major criteria: These criteria have a high impact on the outcome. Preference is set to 2 or 3
Normal criteria: These criteria have a normal impact on the outcome. Preference is set to 1
Platform weights: The scale for weight are from -1 to 2.

-1	Not possible, create a negative impact for this platform for the specified criteria
0	Not favorable or no support
1	Neutral favorable or limited (functional) support
2	Very favorable or fully supported

16 APPENDIX G: PROTOTYPE AHP

A screenshot of the Analytical hierarchy process prototype, worked out in Microsoft Excel. It requires the input in the blue colored columns. A screenshot of the sheet can be seen in the image below.

Adjust these

	Criteria	Enabled	Criteria weight	Description
C1	Offline Access	1	0,05	App should be usable without (internet) connection
C2	Device Sensors	1	0,04	App uses device sensors like GPS, camera, compas, etc
C3	Market Distribution	1	0,22	App visibility in market store, using that distribution channel
C4	Multi Platform	1	0,22	App should be deployed on multiple mobile OS'en
C5	Native Mechanisms	1	0,05	App employs native functionality like push notifications, native ui
C6	Info channel	1	0,27	App only displays text and multimedia
C7	Cost	1	0,16	Total costs of app; in other words the available budget
C8	Updatability	0	1	Instant update of program / content
C9	Personalization	0	1	Personalized & Dynamic content based on location/preference etc?
C10	Responsiveness	0	1	Responsiveness of the user interface (for interactions)
C11	Complex / native UI	0	1	Complexness and/or usage of native UI
C12	Screensize	0	1	Support for multiple screensizes
C13	Heterogeneity devices	0	1	Heterogeneity of devices, even within 1 OS (BYOD)
C14	High expectations	0	1	Serve high-end users with high expectations
C15	User coverage	0	1	Reach all users, no matter the device
C16	Encrypted data	0	1	Storing encrypted date on device/cloud
C17	Market fee	0	1	3rd party fees for monetization of apps and in-app transactions
C18	Regulations	0	1	Impact of restrictions (Terms of Service) from OS vendors

	Native	Hybrid	Web
Platform Score	0,28931499	0,36995057	0,340734435
Preferred platform:	**Hybrid**		

	Weight native	Weight Hybrid	Weight Web
C1	7	5	1
C2	7	5	1
C3	7	7	3
C4	3	5	7
C5	7	5	1
C6	3	5	7
C7	3	7	7
C8	3	5	7
C9	7	5	3
C10	7	5	3
C11	7	3	3
C12	3	5	7
C13	3	7	7
C14	7	5	3
C15	3	5	7
C16	7	5	5
C17	3	3	7
C18	3	3	7

	Total weight	90	90	86
	Major	37	39	27

	Weight native	Weight Hybrid	Weight Web
C1	0,538461538	0,384615385	0,076923077
C2	0,538461538	0,384615385	0,076923077
C3	0,411764706	0,411764706	0,176470588
C4	0,2	0,333333333	0,466666667
C5	0,538461538	0,384615385	0,076923077
C6	0,2	0,333333333	0,466666667
C7	0,176470588	0,411764706	0,411764706
C8	0,2	0,333333333	0,466666667
C9	0,466666667	0,333333333	0,2
C10	0,466666667	0,333333333	0,2

Enabled: Did the customer mention this criteria? 0 = No, 1 = Yes

Criteria weight: How strong is the preference for this criteria. Scale is from 1 till 3 (higher = stronger)

Major criteria: These criteria have a high impact on the outcome. Preference is set to 2 or 3

Normal criteria: These criteria have a normal impact on the outcome. Preference is set to 1

Platform weights: The scale for weight are from -1 to 2.

1	Not possible, create a negative impact for this platform for the specified criteria
3	Not favorable or no support
5	Neutral favorable or limited (functional) support
7	Very favorable or fully supported

17 APPENDIX H: VALIDATION EXPERIMENT PROTOCOL

The protocol describes the purpose of the experiment and general outline.

Goal

The goal of the interactive session is to validate two prototype models. The research provided several mobile app criteria based on literature and expert interviews. The impact of these criteria has been weighted in relation to the mobile implementation platforms, and molded into decision support tools. The experiment is to validate the two prototype tools on the following three points:

- **Practical applicability**: The extent to which the models are applicable in practice
- **Correctness of the models**: Does the sessions deliver new insights to improve the models? But also to which extent the results match the expected results
- **Preference of a model**: A personal preference by the participant for a model

Setup

The experiment is an interactive session with several components. It is qualitative in nature, which creates flexibility to improvise during the session and deviate from the pre-set script.

The four components are:

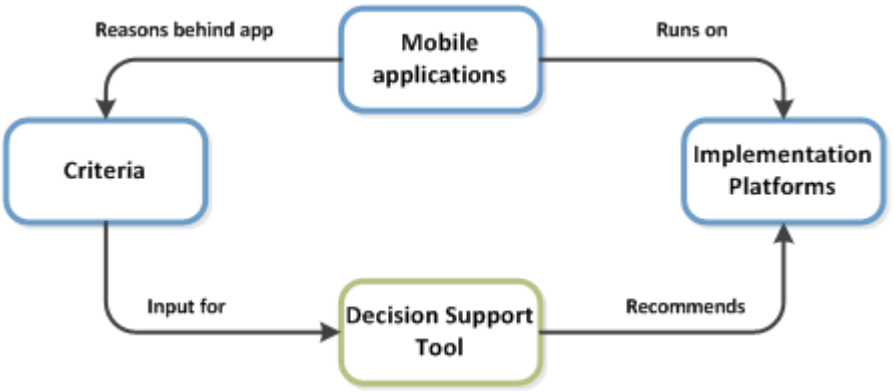
5. **Introduction**: An introduction of myself, the thesis, and goal of the session. Also the time for a disclaimer on the results
6. **Interview**: A short interview that is based on open questions asked from a business perspective. It retrieves the criteria on a qualitative manner. Each question is formulated in a way that it could answer if 1 or more criteria are used for the mobile app that the client wants to realize. After the interview it is necessary to discuss the found criteria, to confirm understanding of the requirements on the app.
7. **Model testing**: The found criteria are entered in the Excel prototypes. The prototypes are based on a Weighted decision matrix (WDM) and Analytical hierarchy process (AHP).
8. **Reflection**: Afterwards the results are discussed, but also the suggestions for improvement. The last question is which model has the preference of the client.

Introduction

First of all, the participant should be asked permission for recording (audio only) the session. Another matter that should be addressed is the disclaimer on the results of the session, as the models are prototypes and still in development. The results are subject to change and are not binding advice.

The introduction has the following elements:

- Introduce myself
- Explain what the goal of my research is; see also the image which is a visual aid in explaining the research in general terms
- Explain the goal of this session:
  - Validation of practical application
  - Scientific validation of the models
- Introduction by the participant



Interview

The interview is based on the same constructs as the semi-structured interviews that were conducted with the mobile experts, see also APPENDIX C: INTERVIEW PROTOCOL on page 62 and APPENDIX D: SUMMARIES OF INTERVIEWS on page 66. The reason for the qualitative approach is that the participant can express him/herself in his/her own words and terms. The interviewer needs to be alert if these match any of the criteria, and adjust or further inquire when it is not clear what the participant means. The interview will have a maximum duration of 40 minutes.

There are four leading questions:

5. What is your experience with mobile applications and devices?
6. What is the purpose of the mobile application?
7. What are the (business) drivers behind the mobile app?
8. Are there (technical) requirements for the mobile app?

The first question is to get a feeling for the understanding of mobile apps that the participant has. The last three are mainly to determine what kind of app the customer has in mind. Possible follow up questions are listed below, with the possible mentioning of criteria between parentheses. The criteria are summed up, at the end of this protocol.

1. What is your experience with mobile application and devices?
2. What is the purpose of the mobile application?
   - Who are the target users? (market / user coverage / multiplatform)
   - What expectation do you have from the app? (expectations)
   - What is the app process and integration for the app? (complexity / sensors)
3. What are the (business) drivers behind the mobile app?
   - What are the considerations between budget and quality/functionality/capabilities? (costs)
   - What content would the app offer? (dynamic content / regulations / info channel)
   - Will the app offer a personalized experience, or does everybody get the same? (personalization)
   - Will the app include a profit model? (market distribution / fee / regulation)
4. Are there (technical) requirements for the mobile app?
   - How should (personal) data be treated / protected (encryption)

- Should the app work without internet access? (offline access)
- Will the app use GPS, camera, or any other device sensor? (sensors)
- Does the content have to be displayed instantly? (responsiveness / offline access)
- Do you expect many updates and/or versions of the app? (updatability)
- Will the app comply to the native OS style, or will it have its own look, feel and branding? (native UI)

After finishing the interview, the interviewer will provide the participant with a list of found criteria. Together they go over the list, and confirm by each criterion the participation in the models. By doing so the participant gains insight into the criteria that are sought; and the interview can confirm if he/she understood the requirements of the app.

Criteria

The descriptions of the criteria, as they were at the time of experiment, are listed in the table below.

Number	Criteria name	Description
C1	**Offline Access**	App should be usable without (internet) connection
C2	**Device Sensors**	App uses device sensors like GPS, camera, compas, etc
C3	**Market Distribution**	App visibility in market store, using that distribution channel
C4	**Multi Platform**	App should be deployed on multiple mobile OS'en
C5	**Native Mechanisms**	App employs native functionality like push notifications, native ui
C6	**Info channel**	App only displays text and multimedia
C7	**Cost**	Total costs of app; in other words the available budget
C8	**Updatability**	Instant update of program / content
C9	**Personalization**	Personalized & Dynamic content based on location/preference etc?
C10	**Responsiveness**	Responsiveness of the user interface (for interactions)
C11	**Complex / native UI**	Complexness and/or usage of native UI
C12	**Screensize**	Support for multiple screensizes
C13	**Heterogeneity devices**	Heterogeneity of devices, even within 1 OS (BYOD)
C14	**High expectations**	Serve high-end users with high expectations
C15	**User coverage**	Reach all users, no matter the device
C16	**Encrypted data**	Storing encrypted data on device/cloud
C17	**Market fee**	3rd party fees for monetization of apps and in-app transactions
C18	**Regulations**	Impact of restrictions (Terms of Service) from OS vendors

Criteria table

This table can be used during the interview to keep track of which criteria the participant had mentioned. It has also room for new criteria that he participant mentions, but it not listed as a criteria; which can be useful for evaluation purposes.

		Criteria	Mentioned
Major criteria	C1	Offline Access	
	C2	Device Sensors	
	C3	Market Distribution	
	C4	Multi Platform	
	C5	Native Mechanisms	
	C6	Info channel	
Normal criteria	C7	Cost	
	C8	Updatability	
	C9	Personalization	
	C10	Responsiveness	
	C11	Complex / native UI	
	C12	Screensize	
	C13	Heterogeneity devices	
	C14	High expectations	
	C15	User coverage	
	C16	Encrypted data	
	C17	Market fee	
	C18	Regulations	
Own criteria	C19		
	C20		
	C21		
	C22		
	C23		

Model testing

There are two models that are tested in the experiment. Both prototypes of the tools are built in Excel, which provided immediate feedback as the criteria are entered into the Excel-sheets. The first model is based on a Weighted decision matrix (WDM) approach, while the second is based on Analytical hierarchy process (AHP). See also APPENDIX F: PROTOTYPE SAW on page 92 and APPENDIX G: PROTOTYPE AHP on page 104.

Evaluation

The reflection is the opportunity to have a critical look at how the goals are reached. The goals are defined to which extend the models proved to be applicable in a practical environment, and where improvements may lie. The results of the models are discussed and further clarified if needed.

A final question is which model has the preference of the participant, and why.

18 APPENDIX I: VALIDATION EXPERIMENT SUMMARY

Validatie experiment informatie

Datum: 9 Augustus 2012

Disclaimer: De resultaten genoemd in dit verslag zijn een indicatie van modellen die nog in ontwikkeling verkeren. De modellen zullen nog verder worden ontwikkeld.

Doel van sessie

In mijn scriptie worden de verbanden onderzocht tussen mobiele app criteria aan de ene zijde, en mobiele implementatie platformen aan de andere zijde. Hierbij worden mobiele criteria gedestilleerd uit interviews met experts. Vervolgens worden er verbanden gelegd en modellen opgesteld. Het uiteindelijke doel is om een tool op te leveren op basis van 1 van de modellen, die als hulpmiddel gebruikt kan worden om een keuze te maken voor de verschillende implementatie platformen.

Het doel van de interactieve sessie is om de twee modellen te valideren. Dit gebeurd op drie verschillende punten:

- **Praktische toepasbaarheid**: De mate waarin de modellen toepasbaar zijn in de praktijk.
- **Correctheid van modellen**: Komen er tijdens de sessie nog nieuwe inzichten naar voren waarbij de modellen kunnen worden verbeterd? Maar ook de mate waarin de verwachting overeenkomt met de uitkomsten van de modellen.
- **Voorkeur voor een model**: Een persoonlijke voorkeur van de participant voor een model.

KvK

De sessie vond plaats bij de Kamer van Koophandel (KvK), met Ina van Beemdelust.
De Kamer van Koophandel is een onafhankelijke dienstverlener die zich inzet om ondernemers verder te laten komen met ondernemen[22]. Dit doen ze door ondersteuning en informatie te bieden aan ondernemers over alle aspecten van het ondernemerschap. Daarnaast bieden ze bedrijfsinformatie over elk bedrijf en/of branche, waarbij het Handelsregister als bron gebruikt wordt.

Ina is al lange tijd werkzaam binnen de Kamer van Koophandel waarbij ze diverse functies heeft bekleedt. De laatste zeven jaar is ze Hoofd Uitgeverij, bij de marketing & verkoop afdeling. Hierbij is ze verantwoordelijk voor de landelijke websites en papieren uitingen; waaronder de website kvk.nl, de online ondernemers community Hallo!, het magazine “Eigen Bedrijf”, en drukwerk zoals folders, flyers en posters. Ze geeft leiding aan een team van 20 personen waarbij het merendeels gaat om mensen die bezig zijn met web gerelateerde media. Ook is ze actief betrokken bij het opzetten van het mobiele kanaal.

Op dit moment is de KvK aan het oriënteren op het mobiele communicatie kanaal. Hierbij willen ze graag een app ontwikkelen die aansluit bij de diensten en producten die de KvK levert. Veel van de functionaliteit voor de app bestaat al in de website (kvk.nl) . Tevens zijn er meerdere concepten van de app gemaakt en een enquête onder de gebruikersgroep over welke functionaliteit ze graag zouden terug zien.

[22] Citaat van missie van de KvK: http://www.kvk.nl/over-de-kvk/organisatie/missie/

De functionaliteit die de app zou kunnen hebben: zoekfunctie, informatie over bedrijven, koppelingen tussen bedrijven/personen en linkedin, direct contact opnemen met bedrijf/contactpersonen, zoeken van bedrijven in de buurt, uittreksel van Handelsregister. Een nieuwe feature zou het aanbieden van mutaties zijn. Uiteraard moeten er aan het stringente beleid van de KvK voldaan worden, vooral regels omtrent security, content transfer, en transacties

Met de eerste versie van app wil de KvK de basisfunctionaliteit neerleggen waarbij er incrementeel nieuwe features en updates worden uitgevoerd. De doelgroep bestaat uit de ondernemers en ZZP'ers die nu al gebruik maken van de mobiele kanalen.

Opzet

Het experiment is een interactieve sessie met verschillende onderdelen. Het heeft een kwalitatief karakter, waardoor er flexibiliteit is om af te wijken van het vooraf bepaalde script.

De sessie heeft 4 onderdelen:

1. **Introductie**: De introductie bestaat uit een voorstelronde van zowel mezelf als de participant. Verder wordt het doel van de sessie uitgelegd en achtergrond informatie over de thesis gegeven.
2. **Interview**: Het interview gedeelte is er op gericht om op kwalitatieve wijze te achterhalen welke criteria van belang zijn in de huidige situatie. De open vragen zijn zo opgesteld dat er per vraag 1 of meer criteria beantwoord kunnen worden. Deze vragen zijn opgesteld met een business perspectief in gedachten.
3. **Modellen testen**: De gevonden criteria uit het interview worden langsgelopen om te bevestigen dat ze wel/niet toegepast moeten worden. Vervolgens worden de criteria ingevuld in de Excel modellen. Het eerste model is gebaseerd op een Weighted decision matrix. Het tweede model is gebaseerd op Analytical hierarchy process (AHP).
4. **Evaluatie**: Hier wordt gereflecteerd op de uitkomsten van de modellen, maar ook punten die mogelijk verbeterd kunnen worden. Tenslotte wordt er gevraagd welk model de voorkeur geniet.

Resultaten

Tijdens het interview werden er vragen gesteld om te achterhalen of bepaalde criteria wel of niet meegenomen dienen te worden in de modellen. In onderstaande tabel staan alle 18 criteria opgesomd, inclusief of deze zijn toegepast in de modellen. Voor model 2 staat er ook bij welk gewicht eraan is gegeven.

		Criteria	Antwoord interview	Toegepast in Model 1	Toegepast in Model 2	Model 2 gewicht
Major criteria	C1	Offline Access	Nee	Nee	Nee	-
	C2	Device Sensors	Nee	Nee	Nee	-
	C3	Market Distribution	Ja	Ja	Ja	8%
	C4	Multi Platform	Ja	Ja	Ja	17%
	C5	Native Mechanisms	Nee	Nee	Nee	-
	C6	Info channel	Ja	Ja	Ja	22%
		Criteria	**Mee genomen**	**Model 1**	**Model 2**	**Model 2 gewicht**
Normal criteria	C7	Cost	Ja	Ja	Ja	6%
	C8	Updatability	Ja	Ja	Ja	12%
	C9	Personalization	Nee	Nee	Nee	-
	C10	Responsiveness	Nee	Nee	Nee	-
	C11	Complex / native UI	Weet niet	Nee	Nee	-
	C12	Screensize	Ja	Ja	Nee	-
	C13	Heterogeneity devices	Ja	Ja	Nee	-
	C14	High expectations	Weet niet	Nee	Nee	-
	C15	User coverage	Ja	Ja	Ja	21%
	C16	Encrypted data	Ja	Ja	Ja	14%
	C17	Market fee	Nee	Nee	Nee	-
	C18	Regulations	Nee	Nee	Nee	-

Model 1: Weighted decision matrix

Een Weighted decision matrix (WDM) geeft aan elk criteria een gewicht en score per platform. WDM telt vervolgens alle scores bij elkaar op om zodoende tot een totaalscore per platform te komen. Het platform met de hoogste score zou het beste passen aan de hand van de opgegeven criteria.

Het model geeft de volgende verdeling: Web app (21), Hybrid (17), Native(6).
Deze verdeling geeft een voorkeur voor een webgebaseerde app, al is het verschil met een hybride app niet al te groot. Dit wordt verklaard omdat bijna alle criteria van waarbij web het beste scoort meegenomen worden.

Model 2: Analytical hierarchy process

Analytical hierarchy process (AHP) wordt gebruikt om een prioritisering te maken tussen de aangegeven criteria. Dit gebeurd op basis van paarsgewijze vergelijkingen. Hierbij geeft de participant aan in welke mate het ene criteria een voorkeur heeft boven een andere. De granulariteit van de onderlinge criteria is veel fijner dan bij de Decision Matrix.
Verder wordt er gebruik gemaakt van dezelfde gewichten als die in de Weighted decision matrix (WDM), deze zijn al vooral ingesteld zodat de gebruiker zich enkel hoeft bezig te houden met de criteria vergelijking.

Voor dit model werden de 7 belangrijkste criteria tegen elkaar afgewogen.
Deze waren: Market visibility, Multiplatform, Info channel, Costs, Updatability, User coverage, en Encryption. Deze criteria werden paarsgewijs met elkaar vergeleken, waarbij er aangegeven moest worden in welke mate de ene belangrijker was dan de andere. Hierbij kwam de volgende rangschikking naar voren:
Info channel(22%), User coverage(21%), Multiplatform(17%), Encryption(14%), Updatability(12%), Market visibility(8%) en Costs(6%).

Deze waarden werden vervolgens verder ingevoerd in het model dat lijkt op de weighted decision matrix, waarbij de volgende verdeling eruit kwam: Web app(0,42), Hybrid (0,34), Native(0,25).
Hierbij komt nogmaals de web app als geprefereerd platform uit de bus.

Reflectie

Hierbij wordt inhoudelijk gekeken naar de doelen die eerder geformuleerd waren: praktische toepasbaarheid, correctheid van modellen, en de voorkeur voor een model.

De opzet om eerst een kort interview te doen, waarbij er open (business) vragen gesteld worden, werkte erg prettig. Ina gaf openhartig en duidelijk antwoorden waardoor het voor ons eenvoudig was om bepaalde criteria juist wel of juist niet mee te nemen in de modellen. Ook gaf het de mogelijkheden om nuances aan te brengen in bepaalde criteria.
De stap om na het interview nog een keer alle criteria af te lopen om te bevestigen of deze wel of niet meegenomen diende te worden blijkt een noodzakelijke controle stap. Hierbij werd het ook voor Ina duidelijk waar we op hadden gelet tijdens het interview, tevens konden we toelichting geven op de criteria die we hanteren als input voor de modellen

Het invullen in de Weighted decision matrix (WDM) was een fluitje van een cent, waarbij er vrijwel direct terugkoppeling gegeven kon worden. Dit geeft meteen een ongenuanceerd beeld in welke richting er gezocht moet worden.
Het tweede model met AHP bleek een stuk langer te duren, dit komt omdat er 21 paarsgewijze vergelijkingen ingevuld dienen te worden. Dat er maximaal 8 criteria aan mee mochten doen bleek niet een probleem. Het werken in 2 bestanden is suboptimaal en kan verbeterd worden.

De sessie heeft wel een aantal van verbeterpunten opgeleverd:

- **Overlap criteria**: Sommige criteria vertonen te veel overlap, zoals User coverage en Heterogeneity of devices. De criteria zullen opnieuw bekeken moeten worden op overlap,

zodat er niet een scheef beeld ontstaat waarbij 1 feature meerdere keer geteld wordt in de modellen

- **Criteria naamgeving**: Bij het vergelijken van criteria voor AHP bleek het niet altijd even makkelijk omdat naamgeving af en toe te ambigue was. Een voorbeeld is Info Channel vs User Coverage. Bij de vergelijkingen vielen overlappende criteria (zie vorige punt) meer op
- **Gewichten aan criteria**: De gewichten aan bepaalde criteria zullen nog nadere tweaking nodig hebben.

Persoonlijk zou ik een hybride oplossing aanraden, in plaats van de web app die de huidige modellen aangeven. Dit heeft mede te maken met de wens van market visiblity en distribution. De modellen geven een mobiele website aan, omdat de meeste features die gewenst zijn goed uit de verf komen met een website en niet (native)app-specifiek zijn. Een hybride variant gebruikt dezelfde technieken als de web app, maar voldoet dan wel de wens om specifiek van het mobiele distributie kanaal en market gebruik te maken.

19 APPENDIX J: VALIDATION SESSION PROTOCOL

Validation review

This document gives a summary of the mobile app criteria and a prototype tool that has been developed during this master thesis.

Disclaimer: The results from the prototypes are only indicators; please bear in mind that these models are still in development. This document is for review purposes only, and may not be redistributed or copied without the author's written consent.

Author: Ernst Fluttert (e.h.fluttert@alumnus.utwente.nl)

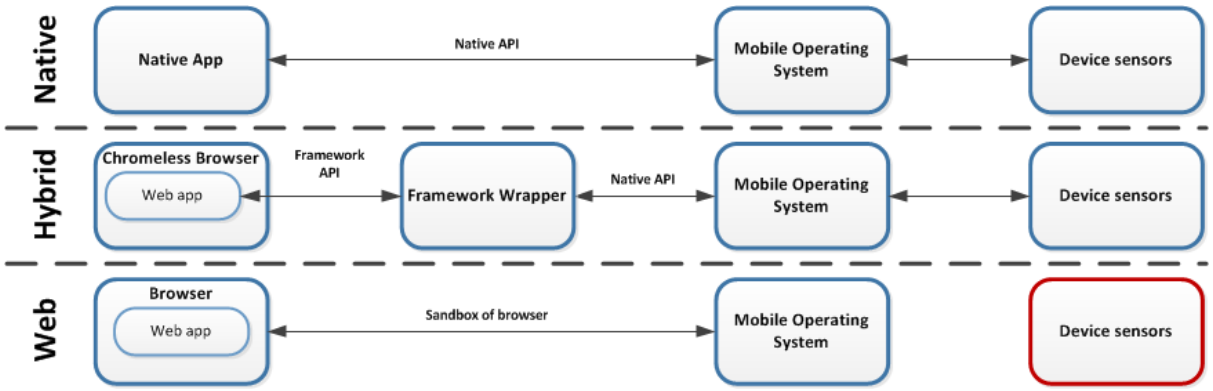
Research overview

Goal

The focus of the research is to link mobile app criteria to implementation platforms. The ultimate goal is to create a decision support tool that can determine the preferred platform based on criteria that the mobile app should have.

Implementation platforms

The platforms used in this research are: native, web, and hybrid. To further illustrate this, see image below. The distinction is made explicitly on the presentation layer. Native apps run directly on the mobile OS (Android, iOS, WindowsPhone, BlackBerry etc), using the native API's and SDK's. A web app is a mobile-optimized website, which is limited by the mobile's browser for presentation and functionality. An intermediate variant is the hybrid; an extra wrapper is used to provide extra (but often basic) native functionality that is offered to the web app. Hybrid and web apps are considered to be cross-platform, while native apps are limited to their respective mobile OS. A well-known hybrid framework is PhoneGap. Frameworks that provide native code (like Appcelerator) are considered native, as the result is a native app.



Mobile app criteria

The gathered criteria are presented for review purposes and will be discussed in detail later in this document. Most of the criteria come from interviews with experts, and only those that influence the implementation platform are included.

Review questions

Before presenting the criteria and the tool, I would to present the questions that are relevant for validation:

1. Is the definition of the implementation platforms clear?
2. Are the criteria and the corresponding norms clear?
3. Are there any missing criteria?
4. Would you agree with the weights of each criterion with respect to the implementation platform?
5. What do you think of the tool, where the criteria and weights have been incorporated into one package?
6. What is your opinion about the practical usage of the tool?

Could you (briefly) address each question in your answer?

Criteria

Criteria are part of the core research which is the reason to give more insight in how they were formed and selected. First all the raw and uncategorized criteria from interviews are shown in section “First list after interviews”. From this extensive list a final list was made based on relevance for the three implementation platforms: native, web and hybrid; this is shown in section “Final set of criteria”. Finally in section “Criteria Weights” all weights per criterion are shown.

First list after interviews

Each of the interviews was compared with this list of criteria:

1. **User Interface**: Criteria that have requirements on the user interface part of an app
    a. Responsiveness: How snappy is the interface? Is this level acceptable?
    b. Animations and sophistication: The complexity of the UI, use of openGL or webGL
    c. Screensize: support for multiple screensizes (tabs/pads and smartphone)?
2. **Offline access**: Does the app require internet, or should it stand-alone?
3. **Device capabilities**: Are there requirements for the device?
    a. Sensors (GPS / camera / compass / etc)
    b. Special needs (ruggedized)
4. **Visibility**: Visibility in stores/markets
5. **Multiplatform**: Does the app have to run on multiple mobile OS’s?
    a. Portability: The portability of apps between the mobile OS’s
6. **App development**
    a. Skillset required by developer: The technical skills required to create the app
    b. Tooling / IDE: The required and preferred IDE and other tools for app building
    c. In-house vs outsourcing: The choice of developing in-house versus outsourcing
    d. Open source: What role does open source play for apps?
    e. Updatability: How easy is it to update the app, within the mobile infrastructure?
7. **Business Model**
    a. Business case: The proposition; the ROI; the earning model
    b. Budget: The overall budget for the app for development and quality
    c. Support & SLA: Maintainability of the app, after delivering the app
    d. Purpose / goal of app: What gap does it fill for the user?
8. **Target group**: Who is the target group, and how is this defined?
    a. Heterogeneity: What devices does it have to support (even within 1 mobile OS)
    b. BYOD (bring your own device): Does BYOD affect choice of platform?
    c. User expectation: High-end market? Low-end market?
9. **Security**: What security features are required?
    a. Mobile Device Management: Will features like MDM work?
    b. Remote Wipe: Does remote wiping apply for apps?
    c. Encryption of data: What are proper ways to store sensitive data?
10. **Platform features**: General features for NATIVE/WEB/Hybrid platforms)
    a. Ease of back-office integration: Does a platform perform better than others?
    b. Licenses / Fees / TOS / In-app transaction: What is the current status?
    c. Content owning: Who owns the content, once published?
11. **Future proof**: Are the platform future proof?
12. **Preference**: How does preference for a platform weigh on the choice?

Final set of criteria

The extensive list of possible criteria went through a selection procedure and sorted out the relevant topics that are linked to choices for implementation platforms. The next phase was to transform these topics into a list of criteria and norms.

The table below shows the final list of criteria that is used in the prototype. The first 6 (C1-C6) are considered major criteria and have double (or triple) the weights of the rest (C7-C16).
The norm is the indicator if the criteria should be applied; while the inverse norm is when the opposite of the normal norm should be applied. Example: The criteria “Device sensors” has the norm that at least 1 or more sensors should be used in the app; the inverse norm is true when no sensors are used.

#	Criteria	Norm	Inverse Norm
C1	**Connectivity**	App is used without internet connectivity (offline access)	Device is always connected to internet
C2	**Device sensors**	App uses 1 or more device sensors	App uses no sensors
C3	**Application store**	App must be in an application store	No explicit need to be in an application store
C4	**Platform amount**	App is to be deployed on more than 1 mobile OS	App only runs on a specific mobile OS
C5	**Mechanisms**	App requires native functionality to function correctly. It cannot be built with web technology (like advanced encryption / total control of sensors / etc.)	App can run with web technology (html/CSS/JavaScript)
C6	**Content**	App shows only text and images	App has interactive presentation of content.
C7	**Updatability**	There are multiple versions planned for the app	App is final (bug fixes only)
C8	**Personalization**	App show personalized content for used (based on user and/or location)	App shows uniform content
C9	**Responsiveness**	App has to respond immediately on input	Response of more than 1 second is acceptable.
C10	**Native UI**	App requires native elements	App has simple interface (no requirements)
C11	**Screensize**	App has to support many screensizes (phone + tablet + desktop)	App supports limited amount of screensizes
C12	**Heterogeneity devices**	App supports many devices	App is made for limited amount of devices
C13	**User expectation**	User has high expectations, both visually and performance wise	User has normal expectations.
C14	**Encrypted data**	Storing encrypted data on device	Storing NO sensitive data on device

C15	**Market fee**	App will use in-app transactions	App does not use transactions
C16	**Regulations**	Impact of restrictions (Terms of Service) from OS vendors	App has explicit and/or age restricted content

Criteria Weights

After establishing the relevant criteria and norms in section "Final set of criteria"; it is imperative to assess their impact towards implementation platform (native, web, and hybrid). The table, shown below, represents the weights for each criterion. For example, criteria C3 "Market" with the norm that the app must be visible in the appstore/marketplace, is fully supported by native and hybrid platforms but is not possible for web apps.
A special note for criteria C1, C2, and C5: These list a negative impact for web apps (weight of -1), as these explicitly require support from native API's which is cannot be implemented in web apps.

Scale and legend for the table:

-1	Impossible, create a negative impact for this platform for the specified criteria
0	Not favorable or not supported / not possible
1	Neutral favorable or limited (functional) support
2	Very favorable or fully supported

Table showing weights per criterion per implementation platform

	Criteria	Weight native	Weight Hybrid	Weight Web
C1	**Connectivity**	2	1	-1
C2	**Device Sensors**	2	1	-1
C3	**Application store**	2	2	0
C4	**Platform Amount**	0	1	2
C5	**Mechanisms**	2	1	-1
C6	**Content**	0	1	2
C7	**Updatability**	0	1	2
C8	**Personalization**	2	1	0
C9	**Responsiveness**	2	1	0
C10	**Native UI**	2	0	0
C11	**Screensize**	0	1	2
C12	**Heterogeneity devices**	0	2	2
C13	**User expectations**	2	1	0
C14	**Encrypted data**	2	1	2
C15	**Market fee**	0	0	2
C16	**Regulations**	0	0	2

From client to tool

The process of deriving criteria from a client, that wants an app, is done via a qualitative interview. It is the part which a consultant, advisor or expert in mobile apps can transform the wishes and reasons of the client into criteria that fit in the tool.

Interview

The criteria are extracted from a client using a semi structured interview. Most are open questions to ensure that the client is telling the purpose of the app, without bias from the interviewer.

There are four leading questions:

1. What is your experience with mobile applications and devices?
2. What is the purpose of the mobile application?
3. What are the (business) drivers behind the mobile app?
4. Are there (technical) requirements for the mobile app?

The first question is to get a feeling for the understanding of mobile apps that the participant has. Most of the participants are familiar with one mobile OS; this might impact the general view on apps. The last three are mainly to determine what kind of app the customer has in mind. Potential follow up questions are listed below, where they could mention 1 or more criteria in the answer; some are noted between parentheses. Note that this list is not exhaustive, and many more can be asked to get a clearer picture of the app and the corresponding criteria.

1. What is your experience with mobile applications and devices
2. What is the purpose of the mobile application?
    a. Who are the target users? (market / multiplatform)
    b. What expectation do you have from the app? (expectations)
    c. How do you visualize the app, what is the general process? (complexity / sensors)
3. What are the (business) drivers behind the mobile app?
    a. What content would the app offer? (content / regulations)
    b. Will the app offer a personalized experience, or does everybody get the same? (personalization)
    c. Will the app include a profit model? (market / market fee / regulations)
4. Are there (technical) requirements for the mobile app?
    a. Is there sensitive or personal data? How is it protected? (encryption)
    b. Should the app work without internet access? (Connectivity)
    c. Will the app use GPS, camera, or any other device sensor? (sensors)
    d. Does the content have to be displayed instantly? (responsiveness / connectivity)
    e. Are there (incremental) updates planned, or is this app already final? (updatability)
    f. Will the app comply to the style of the OS, or does it have its own branding? (native UI)

After finishing the interview, the interview will provide the participant with a list of found criteria. Together they go over the list, and confirm by each criterion the participation in the models. By doing so the participant gains insight into the criteria that are sought; and the interview can confirm if he/she understood the requirements of the app.

Tool

All the information of the criteria can be put into a weighted decision matrix. This is a linear model where a score per platform is calculated. The platform with the highest score is the most preferable.

Score per criterion per platform is expressed as “Enabled (0 or 1)” x “Criteria weight (1 to 3)” x “Weight for platform(-1 to 2)”. This results in three scores, one for each platform (native, web, and hybrid). Then all criteria are summed up per platform, which is the total score per platform.

In the attachment you can find the prototype of the tool (Excel spreadsheet).
As an example two criteria (C3 and C4) are enabled.

Usage:

- Enabled column (aqua blue), can have one of these values:
  - 0 = Is not relevant for customer
  - 1 = Complies to norm of this criterion
  - -1 = Complies to inverse norm of this criterion (optional)
- Criteria weight (light blue): The relative weight of the criterion, compared to other criteria. These are preset, but can be adjusted if necessary.

A screenshot:

			Fill these columns		
		Criteria	**Enabled**	**Criteria weight**	Norm
Major criteria	**C1**	**Connectivity**	0	2	App is used without internet connectivity (offline access)
	C2	**Device Sensors**	0	2	App uses 1 or more device sensors
	C3	**Market**	1	2	App must be in a market/store
	C4	**Platform Amount**	1	3	App is to be deployed on more than 1 mobile OS
	C5	**Mechanisms**	0	2	App requires native functionality to function correctly
	C6	**Content**	0	2	App shows only text and images
	C7	**Updatability**	0	1	There are multiple versions planned for the app
	C8	**Personalization**	0	1	App show personalized content for used (based on user and/or location)
	C9	**Responsiveness**	0	1	App has to respond immediately on input
	C10	**Native UI**	0	1	App requires native elements
	C11	**Screensize**	0	1	App has to support many screensizes (phone + tablet + desktop)
	C12	**Heterogeneity devices**	0	1	App supports many devices (like for BYOD)
	C13	**User expectations**	0	1	User has high expectations, both visually and performance wise
	C14	**Encrypted data**	0	1	Storing encrypted data on device
	C15	**Market fee**	0	1	App will use in-app transactions
	C16	**Regulations**	0	1	Impact of restrictions (Terms of Service) from OS vendors
Outcome			Native	Hybrid	Web
		Platform Score	4	7	6
		Preferred platform:	**Hybrid**		

Legend & Explanation		
Enabled:	Did the customer mention this criteria? 0 = No, 1 = Yes	
Criteria weight:	How strong is the preference for this criteria. Scale is from 1 till 3 (higher = stronger)	
Major criteria:	These criteria have a high impact on the outcome. Preference is set to 2 or 3	
Normal criteria:	These criteria have a normal impact on the outcome. Preference is set to 1	
Platform weights:	The scale for weight are from -1 to 2.	
	-1	Not possible, create a negative impact for this platform for the specified criteria
	0	Not favorable or no support
	1	Neutral favorable or limited (functional) support
	2	Very favorable or fully supported

20 APPENDIX K: VALIDATION SESSION SUMMARIES

A prototype of the tool and the underlying research has been presented to a few mobile experts to get their opinion and comments. This appendix shows the summaries of these meetings.

Summary validation review #1

Date: 4th of September 2012

Review session setup

The session has four parts:

5. **Introduction**: Introduction of the research, starting with a general overview of the researched topics. It also explains the chosen implementation platforms: native, web, and hybrid.
6. **Criteria**: The process of selecting criteria for mobile apps is discussed, where a total list is presented with all criteria. As not all criteria are necessary affecting the implementation platform, only those that are relevant are included to a final list. Then the weights per platform per criteria are introduced, which is the base of the tool.
7. **Tool**: When all criteria are clear, a test case will be done in the tool. This will confirm or disprove the validity of the tool. Furthermore it gives insight in the workings and practical application of the tool.
8. **Evaluation**: The last part of the review is the reflection of the criteria and tool. Any remarks and comments are reviewed. This is also where the practical use of the tool is assessed.

Review summary

When I explained the implementation platform, he immediately noticed that this was based on the presentation layer but was not explicitly stated. Another factor that could play a role is the how data is retrieved. He describes three states for the communication channel:

- **Full offline**: App is downloaded once, no data exchange with external server afterwards. The app is intended for offline use.
- **Full online**: Data is continuously downloaded and exchanged with an external server. The app only works when an internet connection is available.
- **Hybrid**: The synchronization with the server is periodically, where the app can be used without a continuous connection.

This explanation also corresponds, but nuances the first criteria (C1 connectivity).

Remarks and comments on the criteria are:

- **Connectivity (C1)**: Defined too strong, there are multiple ways (full online, full offline, and hybrid) that an app can use the connection for online and offline activities. The question (or norm) could better the rephrased to: can the app be used without internet?
- **Content (C6) and Personalization (C8)**: Very important criteria which shows how the app will interact with the user.

- **Responsiveness (C9)**: Is a non-criterion. An app should always be as fast as possible, whenever the app should retrieve something from a server then a user expects to wait for it. Perhaps this is a combination of the criteria “connectivity” (C1) and “User expectations”(C13)
- **Native UI (C10)**: Include the question if phone-buttons are used (like the menu-button from android).
- **Screensize (C11) and heterogeneity of devices (C12):** A good thing to keep these separated. Screensize support goes beyond the smartphone platform, while heterogeneity of devices is about the diverse smartphones.
- **Encrypted data (C14)**: Storing sensitive data on the device itself is as strong as the used implementation option (and thus up to the developer). But native app can make use of the standard implementation by the vendor, or choose something more advanced. If sensitive data is stored on servers and downloaded through a secure connection than this is also very safe. Perhaps adjust the weights to 1 : 0 : 1 (Native : Hybrid : Web).

After going over the criteria, we looked at the prototype of the tool. Just a few instructions were enough to let Wim fill the criteria, where he thought of an field mobility app that should be able to work offline and has a barcode scanner. It was clear to enable a criterion, and to add a relative weight to this criterion. For the exact details, please see the image on the next page; there the criteria and weights are shown.

The total score per platform were: native (26); hybrid (18); and web (2). The tool indicates a clear preference towards a native implementation of the app, although a hybrid approach could also work. Web is clearly not an option. Wim told me that the outcome of the tool corresponds with the expectation.

When reflecting upon the criteria and the tool, he definitely sees the practical use. Wim states that all criteria that really matter for an app-implementation-choice are listed. This list can be used when asking a client/customer what kind of app they have in mind. He is also positive about the weighted decision matrix tool, where the linear connection between criteria and outcome can be easily observed. This gives more insight in the dynamics of the criteria influence.

A remark is that the extraction of the criteria from a client depends heavily on the skill of the interviewer (usually a consultant or advisor). Another remark is that the criteria should be updated periodically, because smartphones and new mobile OS versions come out on a regular base. A final comment is that some of the criteria are at best an estimated guess, which is also a fairly good reason that the tool should be adaptable and simple.

Overall Wim thinks that the criteria list and tool has practical use, and he is quite positive about it.

			Adjust these		
		Criteria	**Enabled**	**Criteria weight**	Norm
Major criteria	**C1**	**Connectivity**	1	3	App is used without internet connectivity (offline access)
	C2	**Device Sensors**	1	2	App uses 1 or more device sensors
	C3	**Market**	0	1	App must be in a market/store
	C4	**Platform Amount**	0	2	App is to be deployed on more than 1 mobile OS
	C5	**Mechanisms**	1	3	App requires native functionality to function correctly
	C6	**Content**	0	2	App shows only text and images
	C7	**Updatability**	1	1	There are multiple versions planned for the app
	C8	**Personalization**	1	3	App show personalized content for used (based on user and/or location)
	C9	**Responsiveness**	0	1	App has to respond immediately on input
	C10	**Native UI**	1	2	App requires native elements
	C11	**Screensize**	1	2	App has to support many screensizes (phone + tablet + desktop)
	C12	**Heterogeneity devices**	1	2	App supports many devices (like for BYOD)
	C13	**User expectations**	0	1	User has high expectations, both visually and performance wise
	C14	**Encrypted data**	0	1	Storing encrypted data on device
	C15	**Market fee**	0	1	App will use in-app transactions
	C16	**Regulations**	0	1	Impact of restrictions (Terms of Service) from OS vendors
Outcome			Native	Hybrid	Web
		Platform Score	26	18	2
		Preferred platform:	**Native**		

Legend & Explanation		
	Enabled:	Did the customer mention this criteria? 0 = No, 1 = Yes
	Criteria weight:	How strong is the preference for this criteria. Scale is from 1 till 3 (higher = stronger)
	Major criteria:	These criteria have a high impact on the outcome. Preference is set to 2 or 3
	Normal criteria:	These criteria have a normal impact on the outcome. Preference is set to 1
	Platform weights:	The scale for weight are from -1 to 2.
	-1	Not possible, create a negative impact for this platform for the specified criteria
	0	Not favorable or no support
	1	Neutral favorable or limited (functional) support
	2	Very favorable or fully supported

Summary validation review #2

Date: 6th of September 2012

Review session setup

The session has four parts:

1. **Introduction**: Introduction of the research, starting with a general overview of the researched topics. It also explains the chosen implementation platforms: native, web, and hybrid.
2. **Criteria**: The process of selecting criteria for mobile apps is discussed, where a total list is presented with all criteria. As not all criteria are necessary affecting the implementation platform, only those that are relevant are included to a final list. Then the weights per platform per criteria are introduced, which is the base of the tool.
3. **Tool**: When all criteria are clear, a test case will be done in the tool. This will confirm or disprove the validity of the tool. Furthermore it gives insight in the workings and practical application of the tool.
4. **Evaluation**: The last part of the review is the reflection of the criteria and tool. Any remarks and comments are reviewed. This is also where the practical use of the tool is assessed.

Review summary

The session started with a summary of the results from the first interviews, and an update on the research. Then we went through the list of criteria and their norms. He had the following comments:

- **Market (C3)**: The market is more than just visibility, it is also about findability. Being in the market is not enough if you cannot be found by your users.
- **Platform amount (C4)**: Perhaps rename this criterion to “market reach” as this can be seen from a user target perspective; instead of relying on mobile OS alone.
- **Content (C6)**: The formulation could be improved, towards passive versus (inter)active app. Or to one-way communication versus two-way communication for content. This depends on the service and organization that offers the app.
- **Updatability (C7)**: Include a roadmap and/or lifecycle in the description and/or norm. This is much clearer than “versions”.
- **Personalization (C8)**: A very important criterion, as the smartphone is a personal device. In Wim’s opinion this should be a focus from the start for any app.
- **Responsiveness (C9)**: Also a very relevant aspect of any app, yet difficult to formulate a norm. It really depends on the context of the app and the user.
- **Native UI (C10)**: Users are used to User Interface (UI) patterns that belong to each mobile OS. Examples are look and feel of buttons; usage of (physical) buttons; screen position of elements (like menus, back-button and tab-bars).
- **Screensize (C11)**: Next to screensizes for smartphones and tablets, it could also be another factor that is considered: screen-rotation. So landscape or portrait modus should be included when considering screensizes.
- **Heterogeneity of devices (C12)**: Make explicit that this is about the amount of phones (hardware differences on a platform). An example for this is the amount of Android phones; they all run Android but vary in CPU, RAM, storage space, GPU etc.

- **User expectations (C13)**: Expectations differ per market segment. There is a difference between the consumers and the enterprise market. Consumers are more focused on design and branding, while the enterprise market is concentrating on functionality and reliability. Although Wim acknowledges that the enterprise apps are also getting more and more attention for the design and branding part, next to the functionality.

Wim misses a criterion that covers development, which in the initial criteria list covered the following topics: skill set required by developer; tooling and IDE; in-house versus outsourcing; open source; and updatability. None of these have an impact on the choice for the implementation platform, except updatability which is included as a criterion (C8). The skill set required by developer and tooling are more or less covered in the platform amount (C4).
Another criterion that was missing are costs. This criterion has been intentionally removed as the tool should give the best implementation platform based on requirements by the client. The total costs (licenses, development costs, support, etc) are a result of this choice. It is better to keep this out of the scope of the tool, or reflect on this afterwards when discussing the result of the tool.
The last remark on the criteria was that he missed the organizational criteria like: bringing in the required skills for the app (development, management, support); and the continuity of an app. These are very valid, and only partially covered in the current criteria. A counterargument is that most of these are far beyond the choice for an implementation platform, and thus out of the scope of the tool. Nevertheless they are criteria that could be more visible or incorporated into the tool.

After extensively discussing the criteria, we looked at the prototype of the tool. Wim thought of a banking app where the most important features were performance and cryptographic possibilities. The exact criteria can be found on the next page. The results of the tool are: native (16); hybrid (14); and web (7). So the preference is strong for a native solution, which corresponds to the expectation of Wim. The tool indicates that hybrid is also a good candidate.

Wim thinks that the tool is useful for clients that are in the scoping phase for an app. Although it is designed on an application level, not an organizational level where multiple mobile apps should be considered and maintained.

			Adjust these		
		Criteria	**Enabled**	**Criteria weight**	Norm
Major criteria	**C1**	**Connectivity**	1	2	App is used without internet connectivity (offline access)
	C2	**Device Sensors**	0	2	App uses 1 or more device sensors
	C3	**Market**	1	2	App must be in a market/store
	C4	**Platform Amount**	1	3	App is to be deployed on more than 1 mobile OS
	C5	**Mechanisms**	0	2	App requires native functionality to function correctly
	C6	**Content**	-1	2	App shows only text and images
	C7	**Updatability**	1	1	There are multiple versions planned for the app
	C8	**Personalization**	1	1	App show personalized content for used (based on user and/or location)
	C9	**Responsiveness**	1	1	App has to respond immediately on input
	C10	**Native UI**	1	1	App requires native elements
	C11	**Screensize**	1	1	App has to support many screensizes (phone + tablet + desktop)
	C12	**Heterogeneity devices**	1	1	App supports many devices (like for BYOD)
	C13	**User expectations**	0	1	User has high expectations, both visually and performance wise
	C14	**Encrypted data**	1	1	Storing encrypted data on device
	C15	**Market fee**	0	1	App will use in-app transactions
	C16	**Regulations**	0	1	Impact of restrictions (Terms of Service) from OS vendors
Outcome			Native	Hybrid	Web
		Platform Score	16	14	7
		Preferred platform:	**Native**		

Legend & Explanation	Enabled:	Did the customer mention this criteria? 0 = No, 1 = Yes
	Criteria weight:	How strong is the preference for this criteria. Scale is from 1 till 3 (higher = stronger)
	Major criteria:	These criteria have a high impact on the outcome. Preference is set to 2 or 3
	Normal criteria:	These criteria have a normal impact on the outcome. Preference is set to 1
	Platform weights:	The scale for weight are from -1 to 2.
	-1	Not possible, create a negative impact for this platform for the specified criteria
	0	Not favorable or no support
	1	Neutral favorable or limited (functional) support
	2	Very favorable or fully supported

Summary validation review #3

The third validation review is based on the review package only, without an interview. The reaction was:

“Overall the tool looks good. I plugged in a recent client and it came back with the answer I expected.

A few things to point out though related to C1/C2. There is the possibility for offline and sensor input via HTML5, but the implementations and usage are not as well documented and vary greatly between devices/OS versions/platforms. While I don't think that changes the overall tool, it may impact the weighting a bit, and should probably be called out.

What might be helpful is to include a few sample company use-cases to show the results along with the tool. For instance look at 3 different use cases where Native/Hybrid/Web are chosen for a given sample client to give people an idea of how to check that their outcome lines up with reality.

But yeah, having this list of criteria and being able to use this with prospective client will be a great addition to our toolkit.”

21 APPENDIX L: TOOL ANALYSIS CODE

This code is also available on https://gist.github.com/c8e3699a46c629b5cbdb

```
package SA;

/** current output
SA: Starting Application
=== Binary options (0, 1)
Total amount of criteria: 16
Total amount of combinations: 65536
Favor Native: 38360
Favor Hybrid: 6960
Favor Web: 11396
No Favor: 8820

Now with minimum of 5 criteria (total of: 16)
Total amount of combinations: 63019 (total of: 65536)
Favor Native: 37150 (total of: 38360)
Favor Hybrid: 6731 (total of: 6960)
Favor Web: 10729 (total of: 11396)
No Favor: 8409 (total of: 8820)

=== Ternary option (-1, 0, 1)
Total amount of criteria: 16
Total amount of combinations: 43046721
Favor Native: 16635006
Favor Hybrid: 4361872
Favor Web: 18145746
No Favor: 3904097

Now with minimum of 5 criteria (total of: 16)
Total amount of combinations: 43012608 (total of: 43046721)
Favor Native: 16622515 (total of: 16635006)
Favor Hybrid: 4358913 (total of: 4361872)
Favor Web: 18131918 (total of: 18145746)
No Favor: 3899262 (total of: 3904097)
SA: Ending Application
 */


/**
 * class SA :: Sensitivity Analysis
 * This class takes a weighted decision matrix, and calculates all possible
scenarios; by enabling or disabling criteria
 * The summary shows which platform is favored, based on the weights of the
criteria
 * @Author: Ernst Fluttert
 * @Date: 29 august 2012
 */
public class SA {

     // outcome variables
     private int outcomeWeb;
     private int outcomeWebMC;
     private int outcomeNative;
     private int outcomeNativeMC;
     private int outcomeHybrid;
     private int outcomeHybridMC;
```

```
	private int outcomeUndecided;
	private int outcomeUndecidedMC;
	private int outcomeTotal;
	private int outcomeTotalMC;

	private int minimumCriteria = 5;	// only count with minimum amount of
involved criteria

	// list of criteria, each with 4 weights (impact, native, hybrid, web)
	private int[][] criteria;	// the list of criteria
	private String[] criteriaNames;	// name of the criteria

	// constructor, initializes the numbers and criteria
	public SA(){
		// put counters on 0
		resetCounter();

		criteria = new int[16][4];

		// === criteria (weight, native, hybrid, web)
		// C1 connectivity (offline access?)
		criteria[0] = new int[]{2, 2, 1, -1};

		// C2 Device Sensor usage
		criteria[1] = new int[]{2, 2, 1, -1};

		// C3 Market (app must be in market)
		criteria[2] = new int[]{2, 2, 2, 0};

		// C4 Amount of platforms (more than 1?)
		criteria[3] = new int[]{3, 0, 1, 2};

		// C5 Mechanisms (do you need native functions)
		criteria[4] = new int[]{2, 2, 1, -1};

		// C6 Content (text and image only?)
		criteria[5] = new int[]{2, 0, 1, 2};

		// C7 Updatability (multiple versions planned?)
		criteria[6] = new int[]{1, 0, 1, 2};

		// C8 personalisation (personalized content)
		criteria[7] = new int[]{1, 2, 1, 0};

		// C9 responsiveness (immediate vs delayed)
		criteria[8] = new int[]{1, 2, 1, 0};

		// C10 native ui elements like hardware buttons and look and feel
		criteria[9] = new int[]{1, 2, 0, 0};

		// C11 Screensize, also support pads?
		criteria[10] = new int[]{1, 0, 1, 2};

		// C12 heterogeneity device number of hardware devices
		criteria[11] = new int[]{1, 0, 2, 2};

		// C13 user expectation, high expectation?
		criteria[12] = new int[]{1, 2, 1, 0};
```

```
            // C14 Encrypted data?
            criteria[13] = new int[]{1, 2, 1, 2};

            // C15 Market fee: use transactions?
            criteria[14] = new int[]{1, 0, 0, 2};

            // C16 market regulations
            criteria[15] = new int[]{1, 0, 0, 2};

      }

      // reset the counters to zero
      public void resetCounter(){
            outcomeWeb = 0;
            outcomeWebMC = 0;
            outcomeNative = 0;
            outcomeNativeMC = 0;
            outcomeHybrid = 0;
            outcomeHybridMC = 0;
            outcomeUndecided = 0;
            outcomeUndecidedMC = 0;
            outcomeTotal = 0;
            outcomeTotalMC = 0;
      }

      // function run: starts the test. Calculates all possible mutations, and
passes this off for analysis
      public void run(){
            // determine number of criteria
            int numberOfCriteria = criteria.length;

            // First Run: binary choices ( 0 , 1)
            binaryChoice(numberOfCriteria);
            System.out.println("=== Binary options (0, 1) ");
            stats();

            // reset stats before second run
            resetCounter();

            // Second Run: Ternary choice ( -1 , 0, and 1)
            ternaryChoice(0, numberOfCriteria, "");
            System.out.println(" ");
            System.out.println("=== Ternary option (-1 , 0, 1) ");
            stats();
      }

      // all possibilities for a binary option (yes or no)
      public void binaryChoice(int numberOfCriteria){
            // to calculate every combination of criteria (either on or off)
            // This is done by taking taking 2 to the power of number of criteria ( 3
criteria would yield 8 possibilities, 5 criteria would yield 32 etc)
            for (int i = 0; i < Math.pow(2, numberOfCriteria); i++) {
                  // the magic step is to create the binary representation of the
number
                String binaryRep = Integer.toBinaryString(i);
                while (binaryRep.length() < numberOfCriteria){
// is it the specified length?, output should be
                  binaryRep = "0" + binaryRep;
// add leading zero's
```

```
            }
            calcRow(binaryRep, true);                                    // process
the criteria as binary
        }
    }

    // all possibilities for a ternary option ( inverse norm - doesnt apply -
norm)
    public void ternaryChoice(int depth, int maxdepth, String row){
        if(depth == maxdepth){
            // maximum depth reached please calculate the values per platforms
            calcRow(row, false); // process criteria as ternary
            return;         // done with recursion :P
        }
        else{
            String origRow = row;
            for(int i = 0; i<3; i++){
                //System.out.println("i="+i+"  depth:"+ (depth+1) + "
maxdepth:"+maxdepth);
                // continue with recursion
                ternaryChoice(depth+1, maxdepth, origRow+i);
            }
        }
    }

    // function calcRow: Takes a binary string (like "0011101010"), and calculates
the weight per platform, based on the weight specified in the constructor
    public void calcRow(String cr, boolean binary){
        // convert the string to a character array
        char[] crt = cr.toCharArray();
        int correction = 0;
        if(!binary){correction = 1;}
        // cr = current row. This is a string with 0 or 1 for each criteria
        // cr[0,1,0] = only the second criteria is enabled. Please calculate the
result of this for each platform
        int on = 0;          // native platform
        int oh = 0;          // hybrid platform
        int ow = 0;          // web platform
        int crl = crt.length; // length of row
        int ec = 0;     // number of enabled criteria

        // determine if minimum amount of criteria is applied
        for(int z=0; z<crl; z++){
            int curval = (Integer.parseInt( Character.toString(crt[z]) ) -
correction);
            if(curval== -1 || curval == 1 ){ec++;}
        }

        // process the string again for values
        for(int z=0; z<crl; z++){
            //System.out.println("crl: "+ crl +" z:" + z + " on:" + on + "
crt(z):" + (crt[z])  + " (int)crt[z]: " + Integer.parseInt(
Character.toString(crt[z]) ) );
            // SUM the enabled criteria * weight of criteria * impact on
implementation platform
            int enabled = (Integer.parseInt( Character.toString(crt[z]) ) -
correction);
            on += ( enabled * criteria[z][0] * criteria[z][1]);
            oh += ( enabled * criteria[z][0] * criteria[z][2]);
```

```
                ow += ( enabled * criteria[z][0] * criteria[z][3]);
            }
            //System.out.println(cr + " == "+ on + "  "  + oh + "  "+ ow );

            // analyse and process the score for this sequence of criteria
            analyseResult(on,oh,ow, ec);
      }

      /** function analyseResult :: compares the scores per platform and decides
which has the highest score
       * @param on : OutcomeNative , the score of the native platform
       * @param oh : OutcomeHybrid , the score of the hybrid platform
       * @param ow : OutComeWeb , the score of the web platform
       * @param ec : EnabledCriteria , the amount of enabled criteria
       */
      public void analyseResult(int on, int oh, int ow, int ec){
            // process normal results
            if(on > oh && on > ow){outcomeNative++;}       // native has highest
score
            else {
                  if(oh > on && oh > ow){outcomeHybrid++;}  // hybrid has highest
score
                  else{
                        if(ow > on && ow > oh){outcomeWeb++;}     // web has highest
score
                        else{
                              // there is no highest, so it is undecided
                              outcomeUndecided++;
                        }
                  }
            }
            outcomeTotal++;                                          // check if all
possibilities are checked (double check feature)

            // process if minimum amount is satisfied
            if(ec >= minimumCriteria){
                  if(on > oh && on > ow){outcomeNativeMC++;}           // native has
highest score
                  else {
                        if(oh > on && oh > ow){outcomeHybridMC++;}     // hybrid has
highest score
                        else{
                              if(ow > on && ow > oh){outcomeWebMC++;}   // web has
highest score
                              else{
                                    // there is no highest, so it is undecided
                                    outcomeUndecidedMC++;
                              }
                        }
                  }
                  outcomeTotalMC++;                                                //
check if all possibilities are checked (double check feature)
            }
      }

      // print the stats
      public void stats(){
            System.out.println("Total amount of criteria: " + criteria.length);
            System.out.println("Total amount of combinations: " + outcomeTotal);
```

```
            System.out.println("Favor Native: " + outcomeNative);
            System.out.println("Favor Hybrid: " + outcomeHybrid);
            System.out.println("Favor Web: " + outcomeWeb);
            System.out.println("No Favor: " + outcomeUndecided);
            if(minimumCriteria > 0){
                  System.out.println("");
                  System.out.println("Now with minimum of " + minimumCriteria + "
criteria (total of: " + criteria.length + ")");
                  System.out.println("Total amount of combinations: " + outcomeTotalMC
+ " (total of: " + outcomeTotal + ")");
                  System.out.println("Favor Native: " + outcomeNativeMC + " (total of:
" + outcomeNative + ")");
                  System.out.println("Favor Hybrid: " + outcomeHybridMC + " (total of:
" + outcomeHybrid + ")");
                  System.out.println("Favor Web: " + outcomeWebMC + " (total of: " +
outcomeWeb + ")");
                  System.out.println("No Favor: " + outcomeUndecidedMC + " (total of:
" + outcomeUndecided + ")");
            }
      }

      /**
       * public method main :: Starts the application
       * @param args
       * @return void
       */
      public static void main(String[] args){
            System.out.println("SA: Starting Application");// print to console that
program starts
            SA analysis = new SA();
            analysis.run();
            System.out.println("SA: Ending Application");  // print to console that
program starts
      } // end main

}
```