Development of Mobile Applications in Regional Companies
Status Quo and Best Practices

Tim A. Majchrzak and Henning Heitkötter
Department of Information Systems, University of Münster, Münster, Germany

Keywords: App, Business App, Mobile Application, Best Practice.

Abstract: Within a few years, smartphones and tablet computers have spread dramatically. While their hardware has become more powerful, the versatility of devices is driven by applications for them. These so-called apps are increasingly developed for business purposes. We wanted to find out whether apps are interesting for all businesses, how they think apps can be useful for them, in which way they consider to use them, and what kinds of challenges and problems they see. Therefore, we worked with local companies and conducted interviews. By analysing the transcripts we got detailed insights of the practial relevance of using apps. We present the status quo of app development by local companies, highlight early best practices described by them, and discuss our findings. Thereby, we raise topics for future research.

1 INTRODUCTION

A couple of years ago, the only mobile devices widely used were mobile phones, which focused on (speech) communication. In theory, the functionality of mobile phones could be extended by e.g. installing small applications written in Java ME. This was seldom done, though. Other devices such as personal digital assistants (PDAs) have to be seen as niche products. Within a few years, smartphones and tablet computers have spread dramatically. They have mostly replaced PDAs as well as other mobile devices. With their increase in popularity, a paradigm shift can be observed. Communication is only *one* of many functions. While the hardware has become very powerful, it is the software of smartphones and tablets that makes them successful. Mobile applications – *apps* – make them versatile and adaptable yet easy to use.

At first, apps were developed either by vendors of mobile platforms such as iOS (Apple) and Android (Google) or by enthusiastic individuals. Slowly but steadily, enterprises are now exploring the possibilities of using apps. The number of apps developed with a commercial interest in mind grows. We wanted to find out *whether* apps are interesting for all businesses, *how* they think apps can be useful for them, *in which way* they consider to use them, and *what* kinds of challenges and problems they see. Business adoption is too recent to get insights from quantitative studies. Thus, we worked qualitatively in a project with regional companies. While this seems like a limitation, it enabled us to keep close contact and conduct very detailed interviews.

This paper presents results from our project. Whenever possible, general implications of our findings are given along with a discussion. Our work makes several contributions. Firstly, it depicts the status quo of app development as derived from a regional sample. Secondly, it presents early best practices learned from the companies. Thirdly, it discusses aspects of app development that pose particular challenges. Fourthly, it theorizes about these findings and points to general ramifications. Findings are specifically suited for businesses just beginning to examine mobile scenarios.

Section 2 of this paper motivates the importance of business apps and explains the background of our project. The status quo of app development as derived from our business partners is described in Section 3. In Section 4, early best practices for business app development and usage are compiled. Section 5 presents a discussion of our findings. An overview of related projects is given in Section 6. Finally, Section 7 draws a conclusion.

2 PROJECT BACKGROUND

Apps are, generally speaking, lightweight, easy-to-use, and optimized for the particularities of mobile

Majchrzak T. and Heitkötter H..
Development of Mobile Applications in Regional Companies - Status Quo and Best Practices.
DOI: 10.5220/0004359303350346
In *Proceedings of the 9th International Conference on Web Information Systems and Technologies* (WEBIST-2013), pages 335-346
ISBN: 978-989-8565-54-9

devices, such as screen size and prevalence of touchscreens. They are distributed via a mobile platform's app store. Demand for apps from consumers and businesses alike is very strong. Often, customers expect companies to be present on mobile devices in one way or another. They want to be able to use their smartphone to access a company's services from everywhere. There are apps of all conceivable categories, ranging from games over social media to shopping.

Our work focuses on *business apps*. We understand them as mobile applications that support business tasks. This definition is broad by intention. It for example includes apps that employ sophisticated graphics (or even are games) but serve a marketing function. However, in most cases apps with form-based input and output are concerned. Apps are interesting for enterprises with respect to several usage scenarios and target audiences. Firstly, a company may want to provide its employees with apps to support its business processes. Secondly, it may offer apps to its customers for marketing, sales, or servicing purposes. Thirdly, a company with a focus on software development may develop apps for and by order of its clients according to their requirements. For companies that support at least parts of their business processes with IT, the integration of mobile devices into these processes appears as the next logical step.

Developing apps poses several challenges, especially to small and medium enterprises, due to several particularities. Up to now, the mobile market is heavily fragmented into incompatible mobile platforms like Android, iOS, or Blackberry. This leads to increased development and testing efforts. Furthermore, mobile operating systems are quite different from well-known desktop operating systems and limited in scope, requiring new concepts.

Small software development companies or businesses with small IT departments often lack the resources and expertise of larger enterprises, but are still eager to profit from the aforementioned benefits of apps. They often have no experience with app development and no department specialized in mobile topics. This was the starting point of our research into the state of mobile development. We contacted several regional companies that proved to be interested in the topic and set up the project *Business Apps*. As part of the project, status quo and best practices of app development were of a particular interest. The participants were selected from a pool of companies managed by the local chamber of commerce. All of them either have IT-intensive business processes (e.g. as financial service providers), or offer IT services and software. In other words: they share an interest to support their business with IT. Participation was of course voluntary; therefore, not all companies we contacted actually took part in the project.

Our department was responsible for managing the project and conducting the research. 11 regional firms of varying sizes and from different industries participated. We chose to first concentrate on a manageable number of regional companies in order to allow for in-depth conversations and insights. For a start, we regard this qualitative approach to be preferable compared to a quantitative survey, because a basic understanding of the current state of mobile development is still missing and is better elicited through personal interviews. With this paper, we intend to examine the current state and provide results which can later be used to plan larger surveys.

In order to investigate the status quo, gather typical requirements of business apps, and recognize best practices, we conducted semi-structured expert interviews mainly with managers but also with developers. Based on a questionnaire that gathered quantitative data of the respective company prior to the actual interview, we discussed several areas: after an introduction, we considered the company's current state of app development – how many apps, if any, had been developed already?; if none, why not?; which use cases did the apps support?; how (well) did the development process work from a business and from a technical point of view? This was followed by questions about the expectations and requirements on the development process of apps. Lastly, we asked our partners about their future plans.

Interviews were guided by an outline of questions, but gave plenty of room for discussion. We did not ask the questions one after another, but let our partners talk guided by intermittent inquiries from our side. This ensured a lively discussion, while still handling all important topics. The outline was developed based on our prior knowledge of the area and keeping in mind general criteria for developing software as described in the literature on software engineering and information systems development. Moreover, we followed the approach chosen for a project in a similar setting: qualitative research with a small number of participants and long, open inteviews (Majchrzak, 2010a; Majchrzak, 2010b). This combination made sure that we did not leave out important aspects. At the same time, the openness of the interviews made sure that notable issues that could *not* be anticipated would still be captured because they would likely be named by the interviewees. As a consequence, interviews lasted about 90 to 120 minutes on average.

After the interviewing phase, we compiled information on the status quo from transcripts of the interviews and analyzed it for typical statements and

often mentioned wishes and expectations. Section 3 describes the result. We also aggregated successful strategies mentioned during the interviews and analyzed them for general applicability, resulting in best practices (Section 4). Moreover, we found several topics of particular interest to smaller IT departments concerned with app development (Section 5), for which no satisfactory solution is available so far.

3 STATUS QUO

In this section, we describe the status quo broken down into the state of development, requirements, and future plans.

3.1 State of App Development

3.1.1 Apps Developed so Far

All participating companies had developed first apps. However, the number was small and typical projects were *proof-of-concepts*. To categorize the apps the level of heterogeneity was too high. In general, most apps were designed to be used internally. Some of them would normally target customers but were implemented for internal assessment.

The described apps share a characteristic: very limited functionality. Only some of them represent *new* ideas. Many apps transfer existing functionality to the mobile realm. Interviewees often told us that functionality available as a Web site would be provided by apps more comfortably. These apps usually did not yet incorporate the full functionality of the considered Web site but only its core services.

Those apps not based on existing functionality are very simple. Oftentimes, only a single function is provided, typically supporting workflows or enabling easy access to information. A typical example is a calendar tool for trash collection days published by a local service provider. A majority of apps is concerned with providing access to information. Managing or even entering data still is negligible. We only learned of few app for that purpose. One was a kind of virtual clipboard used by mechanics to guide inspection activities and send data to a backend. It also made plausibility checks e.g. to make sure that all necessary inspection steps were followed. Another app was used by field staff equipped with tablets. It provided visualization and simple calculation of tariffs of financial service products.

Some apps did not even serve a purpose in terms of functionality. They were merely developed to assess functionality and especially development frameworks such as *PhoneGap* (PhoneGap, 2012) (now *Apache Cordova* (ApacheCordova, 2012)) or the *Sybase Unwired Platform* (Sybase Unwired Platform, 2012). The same applies for demonstration apps, e.g. a *management cockpit* to observe real-time data.

Besides realized apps, many projects had been driven to the design phase but not yet been implemented. This can be explained with the complexity of such apps. The participants described customer-centric apps; some of them were supposed to include cartographic services or make use of device-specific services such as the camera. Realizing these apps would be very valuable as novel services could be offered to customers. However, the companies found it very hard to estimate the actual effort required for doing so. Some companies also scrutinize the possibility of adopting or adapting apps by third-party vendors, specifically for Personal Information Management (PIM).

It has to be noted that the process for initiating apps was versatile. Instead of being launched by central management, several prototypic apps were built on employees' own discretion. These employees had acquired the necessary skills in their free-time and out of curiosity. Some companies realized the potential and provided support by assigning some time for free experimentation or by organizing workshops for experience sharing – the latter were partly held on weekends.

3.1.2 Platforms

Two approaches could be observed for platform choice. The first was to implement for one platform only: Android or iOS, sometimes Blackberry. Usually, this choice was made for apps that were designed for internal use or merely as prototypes. The second was to support Android *and* iOS. Quite notably, almost none of the projects targeting customers supported other platforms while companies stressed the importance of supporting these two. Not supporting the platform chosen by customers was perceived as a risk due to many peoples' quite emotional attitude towards "their" platform. One company noted that the choice of platforms should target to "win and safeguards the trust by customers".

Cross-platform development was only seen as important if customers were the target audience. The participants usually did not support a *bring your own device* (BYOD) policy fearing security problems due to employee-maintained smartphones and tablets. Particularly those companies that had provided staff with devices (usually Blackberry units) even prohibited BYOD. A common strategy for choosing multiple platforms was to take Android and iOS due to

their market shares and to develop an additional Web app to support all remaining platforms. Some companies argued that they still could develop an additional native version should another platform gain relevant shares among their customer base. Some companies refrained from multi-platform support despite its conceptual merits. Reasons were high effort, problems regarding usability and support of native appeal (e.g. due to differences in gestures such as swiping), and a highly increased administrative overhead.

We particularly wanted to learn whether compatibility to multiple platforms would be sought in the future. We found several, in parts strictly contrasting approaches: The *pragmatic* point of view is to support whatever platforms are demanded by customers. Several companies follow this paradigm. Prioritizing *usability* means to weigh up between Web apps and native apps while seeking to minimize effort. A *Web-based* strategy neglects native look & feel and waits for more device-support to be built into technologies such as HTML5 (HTML5, 2012) (which can be used to build fancy apps already (Zibula and Majchrzak, 2012)). The *dual way* proposes to support the *most important* platforms natively while using a Web app to serve all others. The desire to *prevent effort* may strive for supporting relevant platforms but perceives additional effort with skepticism. Following an *observant* strategy, one platform is picked for now but market development is followed closely. Finally, for internal usage a deliberate decision for one platform in accordance with hardware investments is made.

3.1.3 Decision Processes

The kind of developed apps and especially the selected platforms are strongly influenced by technical underpinnings. However, due to our business focus, we also wanted to understand which decision processes lead to the development of apps. Hence, we made this a topic in the interviews.

The simplest option is contract work for a customer. This is only relevant for some of the participating companies, though. Whether multiple platforms are supported solely depends on the customers' requirements – and budget.

In several cases, the companies developed apps proactively to reflect anticipated requirements. These apps were offered to key customers. Some of them also regularly conduct workshops with customers in which requirements are discussed. This in particular is popular if customers are shareholders at the same time (e.g. in cooperative legal entities).

Internally used apps are usually requested by the departments that want to use them to support business processes. Design is problem-driven in this case. The same is possible for external development, but not typical. App development projects initiated by management or even executives were not reported.

One large company had several app development projects initiated by their department for innovation management. The company provides IT services and tries to keep up with cutting-edge development to offer solutions that are mature and safe for business in alignment with rapid technological progress. Although this requires considerable effort among accepting the risk of following technologies that ultimately fail, this company had an edge in app development.

Regarding technological specifications derived from business decisions, no clear picture can be drawn. Most companies have not decided on a strategy, yet. Choice of technology is project-determined. Internal app development often is strongly influenced by the hardware strategy. If a company e.g. fixed to equip employees with Blackberry devices, app platform and development frameworks are mostly predetermined. Only some companies have decided on an intermediate strategy for the next few years; this usually stipulates the support of Android and iOS.

The interviews were used by several company representatives to utter criticism. For example, the closed nature of the iOS platform was seen as counterproductive while the importance of it prevented companies from not adhering to Apple's policy of acquiring a developer's license. Other representatives complained about the level of investment security. The (hostile) coexistence of a number of platforms reminds them of the *browser wars* of the 1990s. The usability to decide which platform is worth to acquire knowledge for is seen as a dilemma due to simultaneous customer demand. Finally, *security* was seen as a main concern with only Blackberry providing sufficient support (also see Section 5.1). It has to be mentioned, however, that all participants perceive profound chances in using the new technologies.

To conclude this phase of the interviews, we asked how satisfied the companies were with their current progress. Not all companies wanted to take a clear position but the general attitude was positive. This is remarkable since experiences are still very limited. At the same time, satisfaction reflects to some degree the potential seen in mobile devices.

It has to be noted as a final remark that some comments on technical issues were contrary to statements regarding decision processes. For example, iOS was praised for its ergonomics and the low effort in coaching employees in using it. This, however, at least partially relates to the restricted nature of the platform that enforces design standards.

3.1.4 Distribution

Apps *may* be distributed differently than typical for classical applications. Data storage devices such as optical discs are inapplicable. Rather, users download apps from Web sites or access them via *app stores*. The latter are organized similarly to Web shops are provide good overview and easy installation. For commercial purposes, software distribution tools are available. To our knowledge, these tools are not (yet) as powerful as their counterparts for distribution of software onto commercial workstations and servers.

In common, companies are in favor of app stores. It is very comfortable for their customers to find and install apps. The possibility to rate apps is seen as a positive enforcing to develop *good* apps. However, if apps can be commented there also is the risk of unjustified criticism. Some companies regard app stores as the only interface to customers; providing them for download on corporate Web site is seen as awkward style. App stores are also considered for internal deployment (so called *enterprise app stores*). There have been critical remarks concerning the deployment time of app stores. It would be "quite cumbersome" for Blackberry and require "up to two days" – for each deployment. The mandatory checks in Apple's app store could take up to a week.

Automatically keeping employee-owned devices up-to-date is seen as a problem. It is vital to e.g. install security patches without user interaction. *Mobile Device Management* might be a solution (also see Section 5.1). Particularly Blackberry (or rather its vendor Research In Motion) was described as advanced in this regard. Apple is seen as focused on consumers but shifting to recognize business customers.

In contrast with our expectation, *virtualization* was not mentioned in the interviews. While virtualization on workstations is used routinely and employees connect to the corporate network using terminal servers, virtualization on mobile devices does not seem to be considered. It might, however, be a future way to offer access to resources and to increase security. This specifically applies to allowing BYOD.

Finally, it has to be noted that enterprises currently do not plan to offer premium services or to sell apps to end-customers (i.e. consumers). Such services might become available in the future but most of the companies we interviewed see apps as an instrument for customer relationship management or to support processes.

3.1.5 Development

The next step was to have a look at the implementation of apps. No processes specifically tailored to apps could be observed. Development is conducted like for other application; due to the nature of apps particularity methods of graphical user interface design are employed. It was noted that app development should be rapid; therefore, agile-influenced processes are used even if normally not used by a company.

A particularity is the team size: it typically is one, i.e. one app is developed by one employee. If they become larger, development processes ought to be adjusted. Roles might be aligned with the structure of apps as determined by the platforms. Android adheres to a modified model-view-controller (MVC) design pattern which would allow for differentiated developer roles. The "one developer" philosophy also leads to a rather unstructured proceeding. At the same time, there can be significant effort for learning technologies. Mastering powerful libraries such as jQuery Mobile (jQuery mobile, 2012) is not trivial.

Regarding the development for multiple platforms, it was noted that there should not be multiple projects. Rather, a single project would ensure persistent quality on the different platforms. Moreover, it was noted that apps should strive for a native look & feel; Web apps would not be perceived as "real" apps by customers.

App testing poses novel challenges. In particular, coping with the different contexts an mobile application resides in is by no means easy (Schulte and Majchrzak, 2012). An example is the transition between different network connections. Moreover, *device fragmentation* is a problem since functions change with devices and there are great differences in available performance.

A final observation is that only few employees (and a small share of all developers) have experience in app development, yet. It can be expected that this will change soon.

3.2 Requirements for Developing Apps

Requirements engineering is an essential activity (Hull et al., 2010). If it is not done meticulously, the final application will be different from what was expected. Therefore, we discussed the role of requirements for app development. Mostly non-functional requirements (i.e., concerning an app's quality) were named, as we did not ask for requirements for one particular app but for expected general characteristics.

It was argued by the companies that requirements should be gathered in a context-dependent way. Device-specific functionality like acquiring GPS coordinates or using *Near Field Communication* is often important. A particularly named requirement was

to develop apps that would enthuse users. However, we expect that this is just a temporary requirement and part of the hype. Some general requirements were named very often. Smartphones and tablets are easy to use; thus, apps should be easy to use. They should look good, provide sufficient performance and responsiveness, be adequately secure, and adhere to privacy standards. There was no clear line regarding a native look & feel. For several participants, this visual criterion was the most important non-functional requirement; for some, however, it was negligible. Energy efficiency was a secondary requirement only but we expect it to become much more important in the future.

Special requirements could be observed for connectivity. Originally, apps were supposed to either be *online* at all times or to work *offline*. However, for many scenarios, it is required to load information if a proper connection is available but to be able to work without a connection as well. This is not trivial to realize and might furthermore require sophisticated synchronization mechanisms.

In general, apps need to be robust regarding improper input data due to the way they are used. Touching the display is considered to be more prone to erroneous input than using an application with keyboard and mouse.

To ease the usage of apps, they should ideally provide a *single-sign-on* functionality. This might lead to problems, though. Typically, smartphones and tablets are used by a single user – but this is not always the case.

Compliance with typical requirements for commercial software might be impossible. For example, many companies have policies that enforce anti-virus software. Hardly any products are available for mobile devices, though. It is unclear what the reaction to this situation will be. Compromised customer data should be avoided at all cost. Moreover, there might be a need for specific privacy. An app that e.g. requests location-based services might seem harmless. If it, however, sends location data to a server regularly, an insecure connection or a data-leak in the server would be dramatic as it would allow for creating *movement profiles*.

Asked about practices for requirements engineering or the creation of requirements catalogs for apps, companies told us that is was too early to be specific.

3.3 Outlook and Conclusions

In the last part of the interviews, we asked companies about their future plans and scrutinized the overall importance they see in apps. For all participating companies at least the usage of IT but in many cases also the development of IT products has strategic relevance. Asked about the relevance of apps, some companies could directly take a position. Some companies see apps as part of their corporate strategy while others only consider them as operative tools. Some also emphasizes that it is not their perception that matters but their customers'.

The companies for which apps have strategic importance usually share a set of expectations. Examples are the

- estimation that PCs and notebooks usage will decline,
- assumption that most people will have a permanent Internet connection in the near future,
- believe that Web-based solutions have extremely high accessibility, and the
- prediction that customer contact will increasingly be bi-directional, involving them in product design.

Seeing apps as a mean to improve processes underlines the great deal of optimism a number of companies have.

About half of the participats plan to increase investments to put more emphasis on apps. The other half plans to assign more budget to app development but to cut down equally on other activities. The companies do not expect apps to fully replace other instruments e.g. for marketing, though. In general, prices for mobile device hardware and app development related infrastructure are expected to plummet.

When asking for a conclusion concerning the status quo of app development, the already reported satisfaction was acknowledged. However, this was mainly driven economically: *current investments were worth it.* Technologically not all companies were completely contented. Particularly the current compromises between quality and effort were seen as unsatisfactory. The economic satisfaction is driven by financial success: even some of the smallest app projects paid off; learning curves were told to be steep. Unsuccessful projects were not necessarily seen as failures since the gained experiences had value. Acceptance of apps among customers and employees beyond a hyped interest could not yet be fully assessed, though.

Finally, we asked the participants to name room for improvements and to provide us with their outlook. Progress was desired in several areas. Firstly, advanced language recognition would open up completely new possibilities. Secondly, security should be improved. Companies would particularly like to see more emphasize put on security on Android and

iOS. At the same time, not all of them considered the concept of Blackberry as perfect since it was seen as a kind of *security by obscurity*. Thirdly, multi-platform support should be possible with small budgets. Fourthly, data throughput during mobile usage should be increased. Fifthly, synchronization of app and backend should be supported by sophisticated tools. Sixthly, enterprise support is trailing and needs to be improved.

In general, technological innovation were sought that would enable companies to accept less compromises when developing apps. Moreover, companies have a desire to learn about customer expectations. In addition, they would like to see studies and evaluations, e.g. of development frameworks.

4 BEST PRACTICES

In the following, best practices as derived from the experiences of the participating companies are described. Due to the early state of app development, they are mainly meant to be thought-provoking impulses. Best practices for cross-platform app development are not given as the level of knowledge about this is low in companies; work on cross-platform best practices has for example been described in (Heitkötter et al., 2012).

The first best practice is to use apps to *motivate developers*. Several companies reported employees would acquire knowledge on app development on their own. Obviously, the topic is seen as *fun* rather than as work. While this cannot be universally assumed for any kind of app development, it might be utilized. If possible, app development projects should be seen as an incentive for employees willing to work on them. Ideally, developers should feel that they can contribute to the success of their company in adopting a new technology. However, incentives should be used with care. Employees that do not share the enthusiasm for apps should not feel like the work they do is considered subordinate.

Secondly, focussing on *component-orientation* and *service-oriented architectures* (SOA) is recommended. While many apps were build as single entities, they are suited to be constructed from standardized components and to use SOA.

Many commercial apps display information and are used to enter data. The functionality typically found is quite similar. For example, several types of diagrams will often be used for visualization and many apps will need to process master file data of customers. Therefore, such apps should adhere to a common architecture and be built as far as possible from standard components. The initial effort to create such components will be higher than to develop a single app but break even usually is reached with the second app already. As a positive side effect, the app landscape will become easier to overview. However, which components to provide as standardized artifacts and which to design individually has to be chosen with much care.

The usage of a service-oriented architecture makes sense if the company is using SOA anyway and apps rely on backend systems. At a first look, using services seems to be adverse for enabling apps to work without a reliable Internet connection. However, having orchestrated services can be beneficial. Instead of relying on a great number of single requests like in a typical client/server-architecture, data is gathered by the backend and exchanged with the app client in few service requests. This could also save computational effort on the device and thus increase battery life. Relying on data processing in backend systems might also increase security. It is considered to be simpler to secure the connection between app and backend then to design secure apps. A final critical remark has to be given, though: implementing SOA solely for apps is not considered to be a good strategy.

Thirdly, we found that enterprise apps might serve to increase *information availability*. Mobile devices are particularly suited for providing information. The first step should be to determine which information can be made available, and whether it makes sense to have it available mobile. *Mobile* in this case is an elastic word: it starts with an employee sitting at his desk and using a smartphone since data access is faster or more convenient. It can mean that employees check their appointments for the next day after returning home. It goes as far as working completely remote.

The second step is to consider the level of criticality of information. The more critical they are, the more effort needs to be put into security for mobile access. The third step is to provide access by means of an app. It is advisable to make apps as customizable as possible. Employees will then be able to use them more effectively. While many companies describe this kind of usage as very rewarding, it is hard to assess its monetary value. An expected increase in employees' working precision and speed is hard to measure and even harder to attribute to such small changes as introducing apps on mobile devices.

Particularly due to having several financial service providers among the participating companies, a fourth best practice could be derived: the *support of field staff and sales*. Reports from usage of mobile de-

vices in this area were very convincing. There are two possible usage scenarios: visualization and data entry. The first step usually is to provide visual access to information. Tablet computer are well suited for this task due to their larger screen, their convenient size, and the less person-attributed feel in comparison to a smartphone.

The vision of using tablets for visualization is to replace pen & paper. They are essential for field staff, e.g. in an insurance sales scenario. In the course of appointments with customers tariffs have to be shown, calculated, and adjusted. Calculating and visually demonstrating individual options requires either immense amounts of preprinted papers or using a laptop. However, the latter is a kind of barrier: while adjusting numbers, the customer only sees the back of it. Moreover, it will be perceived as the sales person's laptop and most likely not used without reluctance. In contrast, papers can be handed over and written onto. They do not allow for ad-hoc adjustments, though. Tablets provide the ease-of-use and low barrier of paper while enabling individual adjustments and recalculations. They are easy to use and even accepted by people with low technical-affinity (Majchrzak et al., 2011). To keep calculated data, an additional app can be offered to customers that will synchronize with what has been worked on with the sales person.

Data entry has similar characteristics: paper is iteractive and personal but forms usually have to be corrected once returned to the company; laptops are comfortable and powerful, and enable checking data in real-time. Again, using tablet computers can be a convenient solution. However, in this case more sophistication is required than for mere visualization since data entry by touchscreens usually is inferior to using keyboards – at least, if much text has to be entered.

Linking apps to a backend not only provides real-time verification but also allows most activities to be (pre-)computed on servers. This simplifies app implementation but again security concerns have to be taken into account. Moreover, apps needs to be designed to provide some kind of "user mode". The sales person might want to access data from all his customers using the app. However, when handing the tablet to a customer to show him some data, it should not be possible to easily browse to the file of another customer.

Despite all praises, investing into mobile technology for sales staff should be done carefully. A laptop computer provides a large set of possibilities and probably even terminal-server access to most of the company's resources. A tablet in contrast is bound to the functionality realized in an app. Therefore, sales processes have to be tailored to allow seamless work. However, this can be used for optimizations at the same time. It is advisable to keep laptops for a while after introducing tablets.

5 DISCUSSION

Our findings regarding status quo and best practices are useful in several areas. Having knowledge of the current state of mobile development in enterprises is important for research on new technologies. The design of mobile enterprise frameworks has to follow the needs of businesses. It also has to be based on the current situation typically found in software development departments to be capable of being integrated into their environment. Designing a product for mobile development without regard for the needs and capabilities of businesses will often prove unsuccessful. The following subsection highlight some of the needs most often mentioned during the interviews. They are the basis for the open research questions that we identify in the second subsection. The best practices that we compiled might serve as recommendations for companies with similar problems. They can also inform further research in related areas, because they highlight what has been successful for solving particular problems and what might be transferable to other areas.

5.1 Topics of Particular Interest

We explicitly asked our interview partners for topics of particular concern. When analyzing the transcripts, we additionally looked for problems that were considered not successfully solved up to now. As a result, we compiled the most often found topics of particular interest for which so far no universal solution exists, at least for small- or medium-sized software departments: security on mobile devices, mobile device management (MDM) in general, cross-platform app development, and testing mobile applications. They are discussed in the following.

Mobile security is a highly relevant topic. It subsumes the security of data and applications on mobile devices used in an enterprise context and is hence of utmost importance due to the critical information present on devices. Security has to be considered both in the administration of employees' devices and when developing apps for internal or external use. On the one hand, standard security considerations also known from PCs or laptops, for example regarding wireless communication and application permissions,

are more prevalent on mobile devices. On the other hand, they require particular security measures due to their mobility, e.g., data protection in case of theft or loss. This unique combination of security risks along with a reduced awareness of users due to the playful nature of mobile devices poses severe challenges. Furthermore, users often blend private and business use to the point of using their private devices for work (BYOD). First solutions to some of the problems are available as part of MDM (see below). MDM solutions often allow to specify security policies and provide means to react to device loss. Platform providers such as Google and Apple prepare guidelines for developing secure apps (And, 2012; Apple Inc., 2012). Nevertheless, a holistic security approach still requires considerable effort and knowledge.

Commensurate to the scope and versatility of mobile devices, *mobile device management* increases in importance. Due the proliferation of apps, smartphones and tablets need to be administered similar to desktop PCs and laptops. This includes the configuration of (new) devices, installation of apps and updates, distribution of relevant enterprise data, and maintenance. Additionally, data should be synchronized with other devices and data sources. Most functions should be provided over-the-air, i.e., remotely without the need to hand in the device to administration. The heterogeneity of the mobile device market further complicates matters and necessitates a cross-platform MDM, which is hindered by the closed nature of mobile platforms with respect to administrative functionality. The MDM market changes rapidly, which makes orientation especially for smaller companies difficult. Abstract, strategic overviews of the market like (Redman et al., 2012) do not offer enough insight for deciding on a MDM solution. In general, one can distinguish between *on-premise* and *cloud-based* MDM. Especially for smaller companies, cloud-based solutions might be viable, because they do not need to be installed and maintained by the company itself.

The mobile market is fragmented into largely incompatible mobile platforms, most importantly iOS, Android, Blackberry, and Windows Phone. In an enterprise context, at least the first two, but often more, need to be considered when developing apps. Implementing an app separately for each platform as a native app requires a lot of resources. *Cross-platform approaches* to app development promise to serve several platforms from a single code base. Several categories can be distinguished (Heitkötter et al., 2012), but all of them have some limitations. Depending on the particular app, a mobile Web app – essentially a Web site optimized for mobile use – might be sufficient, or a so-called hybrid app that has access to device-specific features. However, these Web-based approaches do not produce truly native apps with a native look & feel. No mature cross-platform solution with a native look & feel exists so far, although first steps in this direction emerge such as Titanium Appcelerator (Appcelerator, 2012). A novel approach is to use model-driven software development (MDSD) to build apps. MD^2 for example can be used to describe apps as a model using a domain-specific language and to automatically generate native code from this model (Heitkötter et al., 2013a; Heitkötter et al., 2013b). Similar to the MDM market, cross-platform approaches are an ongoing topic. Smaller development companies need help navigating the confusing market to pick a suitable approach.

During app development, *testing* is a particular concern to ensure a defect free and smooth user experience on all supported devices. Again, heterogeneity causes particular problems. Devices, even with the same mobile operating system, differ significantly with respect to screen sizes, input methods, and other device functionality. These features have a direct impact on the functionality and user experience of apps. Hence, testing an app only on a single device type will not be sufficient. User interface tests, often manually executed, become more important compared to traditional applications. Due to the inherent mobility, apps have to be tested with respect to their reaction to context changes (Schulte and Majchrzak, 2012), e.g. regarding device orientation, a loss of network connection, or relocation. The emulators available for most mobile operating systems are not sufficient for thorough testing. Testers need access to several physical devices with different specifications. Provisioning such a set of current devices is not feasible for smaller app development teams. However, there are some cloud-based solutions that offer manual or automated remote control of mobile devices, e.g. (DeviceAnywhere, 2012). Up to now, companies are on their own when developing a testing strategy that considers the particularities outlined above.

Topics not discussed due to space restrictions are mobile requirements engineering, motivating and training developers, energy efficiency, and coping with offline situations.

5.2 Open Research Questions

While our project has led to insights regarding the status quo of using applications for mobile devices, it also has shown that there is a large number of open questions. This is a limitation of our work: we have

worked qualitatively and with a small sample; findings cannot be statistically significant or may not necessarily be true on a global scale. Therefore, open questions can be derived but answers need to be discovered in future research.

Some of the open questions have a practical nature and stem directly from the findings and special topics discussed above. It is beyond the scope of this paper to provide solutions to the questions that we can rise. In fact, there is much work to be done in a variety of directions and with a number of disciplines to contribute. Therefore, our approach is to rise questions and – if applicable – to give first ideas for paths to solutions.

Firstly, practical problems have to be solved to support companies. Several topics for research can directly be taken from the status quo drawn in Section 3:

- How can companies be supported in deciding whether app development should be pursued by them? How can the business value of using apps be determined?
- For which areas does app development apply? What are particular requirements that apply to apps, and how can they be engineered?
- How should decision processes be adapted to allow more efficient development?
- Which ways of distribution are reasonable? How can companies safely use app stores?
- What are the challenges of app development and what novel techniques might improve development processes?
- Which strategical underpinning might app development have for companies?

For many of the above questions, solution sketches already have been presented. It is our belief that there is no purely scientific answer to most of these questions. In fact, we deem a close observation of corporate practices along with scientific research to be the most adequate path to feasible solutions. This includes refraining from a hyped thinking about apps and seeing them as *one* instrument to support internal processes and customer-related activities.

Secondly, theory-driven work is required. This particularly applies to the purely economical and the purely technical topics. Regarding economical topics, the value-creation by apps has to be scrutinized. It is easy to estimate development effort and relate it to sales figures if apps are offered for a premium. Measuring the effect of process improvements by app usage or even of increased customer support is close to impossible. Therefore, ways have to be found to support the processes that lead to investment decisions. We consider this to be a theoretic question at first: explanatory models for the value of apps in different situations and for varying scenarios have to be found in a first step. The second step is then to evaluate them qualitatively and eventually quantitatively in cooperation with enterprises.

Regarding technology, there are a number of questions that require profound theoretical work before assessing their application to corporate problems. There are four particular areas that we deem to be especially important:

Cross-platform development is supported by various approaches ranging from using Web apps over Web technology-based solutions to sophisticated frameworks based on cross-compiling or model-driven software development technology. Which framework to use in what situation is target of ongoing research (Heitkötter et al., 2012). The same applies to the development of improved frameworks with a focus on business problems.

There is no *sound background* for app development. Companies require guidelines, best practices, and a knowledge base similar to what they can draw from for classical software engineering.

Projects such as (Zibula and Majchrzak, 2012) have started with *scenario-based evaluation of technologies* applied to app development. Such work has to be extended, put into a greater structure, and eventually unified.

There is no *theory of app testing*. At the same time, testing apps is greatly different to testing of desktop and server applications. Of course, many techniques can be transferred and there is no need to come up with completely new theories. After all, apps are computer programs and underlie the same ramifications as any other program. However, the context they run in and the changes in user interaction patters and runtime model ask for amendments to existing theories. Based on this, tools and techniques need to be developed and eventually evaluated on practical examples.

Thirdly, a theoretical base for app development has to be worked on. This is the most open challenge, yet the hardest to work on. While no strong differences to classical applications development could be observed, *some* inherent particularities became obvious. The need to deal with context changes falls into this category. Moreover, the way that mobile technology is used poses some profound differences in terms of value to businesses and perception by humans. This even related to the ongoing discussion of ubiquitous computing (cf. e.g. (Bell and Dourish, 2007)). Finding suitable “small packets” for research and pursuing theoretization in an interdisciplinary way will be a challenge for the upcoming years.

6 RELATED WORK

Due to the nature of our paper and the novelty of the topic, relating to other work is no straightforward task. In a narrow sense, no directly related work that explores mobile strategies from a business point of view and surveys the status quo exists. In a wide sense, almost all papers considering app development are related due to their involvement in the early advances in this area. Work that related a bit more closely to particular aspects of our paper has been cited at the appropriate locations. Therefore, we will only highlight a few approaches that relate in a closer sense.

WASSERMAN (Wasserman, 2010) gives an overview of software engineering issues prevalent when developing mobile applications. He draws on a survey among mobile developers and highlights what makes mobile development different. The main part consists of a research agenda for software engineering, not a description of current practices, and hence is a worthwhile addition to our work. In contrast to our work, he focuses on the immediate environment of the developer himself, but not on the point of view of businesses that decide on mobile strategies. The work of DEHLINGER and DIXON (Dehlinger and Dixon, 2011) takes a similar direction. HOLZER and ONDRUS (Holzer and Ondrus, 2009) study the mobile market from a developer's perspective as well. They intend to give advise to developers on how to engage in mobile app development. Similar to (Wasserman, 2010), their work has a narrower scope than our project, but provides additional insight.

TARNACHA and MAITLAND (Tarnacha and Maitland, 2006) take an entrepreneurial perspective on the opportunities of mobile commerce and analyze the dependencies between technology and business strategy. Hence, they focus on businesses for which mobility is a key factor, as their products entirely depend on it. Our work, instead, looks at enterprises with a modest relationship to software that want to expand towards mobile applications supporting their regular business. Published in 2006, the statements of (Tarnacha and Maitland, 2006) mostly apply to the pre-smartphone area and are in detail not applicable to today's situation.

7 CONCLUSIONS

We presented work on app development by enterprises. Our paper summarized the status quo and best practices learned from regional companies before discussing the findings and formulating various topics for ongoing research.

Besides providing some insights, we hope to stimulate research with our work. Research has to keep pace with the extremely high speed the market for mobile apps evolves with. Moreover, there are many unsolved challenges for which solutions need to be provided.

Our own work is not finished. We are currently working on a project for cross-platform app development, one of the challenges presented earlier. Our idea is to provide a technically sound solution that still keeps a domain-specific focus (on business apps) in mind. Moreover, we keep contact to some of the enterprises we worked with. Thereby, we will tackle on some of the questions raised in this paper.

ACKNOWLEDGEMENTS

We would like to thank the companies that filled out the questionnaire and particularly those that were available for interviews. Their participation made the underlying project of this paper possible.

REFERENCES

(2012). Android developers. designing for security.

ApacheCordova (2012). Apache Cordova.

Appcelerator (2012). Appcelerator.

Apple Inc. (2012). *iOS Security*. http://images.apple.com/ipad/business/docs/iOS_Security_May12.pdf.

Bell, G. and Dourish, P. (2007). Yesterday's tomorrows: notes on ubiquitous computing's dominant vision. *Personal Ubiquitous Comput.*, 11(2):133–143.

Dehlinger, J. and Dixon, J. (2011). Mobile application software engineering: Challenges and research directions. In *Workshop on Mobile Software Engineering*.

DeviceAnywhere (2012). Deviceanywhere.

Heitkötter, H., Hanschke, S., and Majchrzak, T. A. (2012). Comparing cross-platform development approaches for mobile applications. In *Proc. 8th Int. Conf. on Web Information Systems and Technologies (WEBIST)*.

Heitkötter, H., Majchrzak, T. A., and Kuchen, H. (2013a). Cross-platform model-driven development of mobile applications with MD2. In *Proc. of the 2013 ACM Symp. on Applied Computing (SAC)*. ACM.

Heitkötter, H., Majchrzak, T. A., and Kuchen, H. (2013b). MD2-DSL – eine domänenspezifische sprache zur beschreibung und generierung mobiler anwendungen. In *Proc. der 6. Arbeitstagung Programmiersprachen (ATPS 2013)*, number 215 in LNI. GI.

Holzer, A. and Ondrus, J. (2009). Trends in mobile application development. In *Mobile Wireless Middleware, Operating Systems, and Applications-Workshops*, pages 55–64. Springer.

HTML5 (2012). HTML5.

Hull, E., Jackson, K., and Dick, J. (2010). *Requirements Engineering*. Springer, 3rd edition.

jQuery mobile (2012). jquery mobile.

Majchrzak, T. A. (2010a). Best Practices for the Organizational Implementation of Software Testing. In *Proc. of the 43th Annual HICSS*. IEEE CS.

Majchrzak, T. A. (2010b). Status Quo of Software Testing – Regional Findings and Global Inductions. *Journal of Information Science and Technology (JIST)*, 7(2).

Majchrzak, T. A., Jakubiec, A., Lablans, M., and Ückert, F. (2011). Towards better social integration through mobile web 2.0 ambient assisted living devices. In *Proc. 2011 ACM Symposium on Applied Computing*, pages 821–822, New York, NY, USA. ACM.

PhoneGap (2012). Phonegap.

Redman, P., Girard, J., and Basso, M. (2012). Magic quadrant for mobile device management software. Technical report, Gartner.

Schulte, M. and Majchrzak, T. A. (2012). Context-dependent testing of apps. *Testing Experience*, (19).

Sybase Unwired Platform (2012). Sybase unwired platform.

Tarnacha, A. and Maitland, C. (2006). Entrepreneurship in mobile application development. In *Proc. 8th int. conf. on Electronic commerce*, pages 589–593. ACM.

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

Zibula, A. and Majchrzak, T. A. (2012). Developing a cross-platform mobile smart meter application using HTML5, jQuery Mobile and PhoneGap. In *Proc. 8th Int. Conf. on Web Information Systems and Technologies (WEBIST)*.