

4 Tips To Design Effective Mobile Business Apps

August 2013



OUTPERFORM THE FUTURE™

Contents

	Introduction	1
1	Re-engineer the desktop experience	3
2	Where will this be used?	5
3	Use the operating system to your advantage	6
4	Go Platform Independent	7
1.	Native Language Deployment	7
2.	Hybrid Approach.....	8
3.	Web Approach	8
5	Conclusion	9
	About Magic Software Enterprises	10



OUTPERFORM THE FUTURE™

| Introduction

Mobile technology is having a major impact on how we work: smartphones and tablets are taking over the enterprise IT world, and employees increasingly expect enterprise systems to be available and indeed more efficient than on a desktop. However the plethora of devices entering the enterprise, allied with the small screen real estate of mobile devices, can make designing effective screens for your users a challenging and complex experience.

In today's mobile world, it's all about apps. Whether it is Apple telling us, "there's an app for that", analysts blaming the poor market performance of Windows Phone and Blackberry 10 on a lack of apps, or Microsoft deriding a competitor's devices as just being, "a sea of apps", the focus is on apps. But what are apps, and why have they become so significant?

A well designed app is essentially a process, a workflow, wrapped up in an intuitive user interface and optimised to display on your device. In this way it is no different to desktop software, or indeed to a well-designed web page. The differences come from how we use desktop and mobile computing: our input devices, physical location, even our patience are very different when using a smartphone as opposed to a tablet, or laptop, or a desktop.

Although when we hear the word, "app", we typically picture a desktop-style icon which, when clicked, launches the process or workflow we want, this is not the only way that apps may exist in the future. Windows 8 presents one such paradigm, although it is not yet especially well implemented: the "Live Tiles" promise to show relevant information without having to launch a process, delivering information to the user when it is needed. Meanwhile the Moto X smartphone and Google Glass are always listening, ready to provide information when asked; and Google's Chrome OS sees a future of web apps, where the heavy lifting is handled by a server which can provide information as it is needed. The underlying link between all these views of the app is that its purpose is to provide the user with information when they need it, with as little input as possible.

This whitepaper is one of a two-part series prepared by Magic Software on designing mobile business apps. This whitepaper explores the principles underlying successful app functionality: what users are trying to do; where they need to work; and the importance of supporting multiple platforms. The companion piece, *6 Tips To Design Effective Mobile Screens For Business*, is a more in-depth look at how to create an effective user interface that enables users to easily learn, navigate and use the application.



1 Re-engineer the desktop experience

Often, business mobile apps seek to replicate a desktop application or website and make it more accessible to a mobile user, often by focusing on the core of the application and removing more complex features. It's a start, as at least the users have some access to the system, but it does perpetuate the idea that the mobile device is a "second-class citizen"; and in the increasingly mobile-first world we can do so much more by focusing on the workflow, not the applications.

The purpose of a mobile business app is to provide users with the ability to easily carry out the same tasks and processes on a mobile device as they could on a desktop. Therefore the first thing to consider is which processes and workflows your users are trying to complete on their mobile devices.

Typically, desktop processes involve doing several things at once, taking information from multiple sources in order to complete the process. This is workable when the user is sitting comfortably, with a large screen, mouse and keyboard, but when it comes to mobile we cannot make that assumption. Because of the ergonomic and size constraints, mobile processes work best when the user only has to focus on one thing, so the process must be figured out in advance and connect automatically to any information that may be needed.

When working within an ERP at a desk equipped with keyboard and mouse, we can move between different screens quickly to access data from different sources, using our mouse or keyboard to flip between screens and other systems such as email clients and CRM's. But on the mobile, all the information we need at our fingertips has to be digested into one or two screens at the most. This means it's essential to think about the processes involved in the background and look for ways to make these as efficient as possible, and try and 'mash up' all relevant data regardless of the source application into one simple screen.

Consider for example if the user needs to process a purchase order request: making an informed decision may require checking an ERP system to ensure there is sufficient budget; maybe checking the current sales figures or estimated pipeline for the quarter. This is quite a simple process, but it already requires four information flows (Figure 1). While workable on the desktop, chances are that it is simply too time-consuming and fiddly for mobile; especially if our user is in a hurry between meetings.

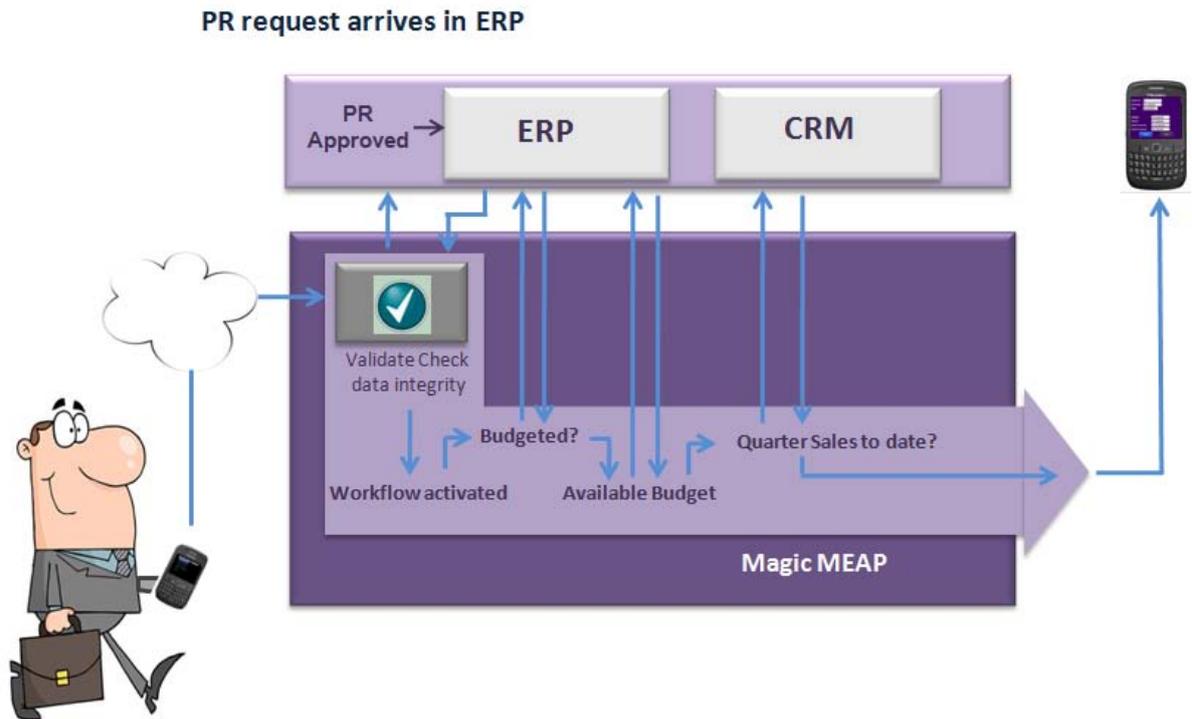


Figure 1: An example of the data flows needed to process a purchase order request.

This example shows the importance of starting to design a screen from the perspective of considering what the user wants to achieve. By offering the user a “purchase order approval” app which displays all the relevant information in one easy to read screen, we have re-engineered the enterprise by focusing on the process, not applications.

2 Where will this be used?

When designing screens for a desktop or laptop, it was fairly easy to predict where the application would be used: the device's form factor all but ruled out any truly mobile uses, for example while walking. It was clear that the device would be positioned on a table, shelf, or lap. With mobile screens, we have to assume that the device and hence the app will be used around the clock, in any location imaginable and often while carrying out other tasks.

What this means is that the user needs to be able to use the screen in situations where they cannot read a lot of text, or locate and press small buttons, and will not have the patience to switch between apps to find what they need. Unlike controlled office spaces, the mobile world often involves unpredictable or poor lighting, and both reading screens and entering data by touch can create ergonomic challenges.

The solution is to ensure that screens are highly legible and unambiguous, with clear indication of data entry fields and buttons. Contrast is extremely important, because in bright sunlight we tend to lose detail, and colours become harder to distinguish. Likewise, when the user is moving around and using their finger to touch buttons, they are going to struggle to accurately touch areas that are too small. While there is no "perfect" button size, it is best to ensure that there is enough area for someone with large fingers to press, without risking touching a different button.

Because the screens are going to be used to inform decisions, and represent a process that the user needs to follow, it is important to present the data in a logical way; it must follow the thought process of the user in order to minimise the cognitive burden. As far as possible, avoid making your users think! In the example of the purchase order request, the mental process would be as follows:

- Was this expense budgeted for?
- How much budget do I have remaining?
- What sales are forecast for this quarter?
- Based on the above, should I approve this request?

This tip naturally follows the first: by considering where they are likely to be when trying to carry out their process, we can anticipate the challenges of that environment and design the screen accordingly.

3 Use the operating system to your advantage

There is a wide range of mobile operating systems that your users could be using, and each of these has its own way of doing things. While mobile OS desktops may look similar, a closer look shows that iOS apps, Android widgets, Windows Phone live tiles and so on all present different ways of achieving the same results. This may appear daunting, but in fact these operating systems are pointing you toward ways in which their users are expecting to work.

The better you understand how the various mobile operating systems work, the easier it will be to appreciate how users expect to receive messages, have notifications reported, switch between screens or apps, and other simple elements of user experience that can make the difference between the screens being intuitive or making their use a chore.

Remember that your screen may end up being used on multiple devices with varying screen sizes and orientations. Consider what the use of different devices may say about the user's location and intentions: for example, when holding a smartphone with a 4" display in portrait mode, the user is far more likely to be busily moving between meetings and hence preoccupied than if they are accessing the screen on a 10" tablet in landscape mode, especially if there is also a keyboard networked to the tablet.

These differences in usage highlight the importance of taking a multi-channel approach as the same familiar look and feel of the screen should be kept across multiple devices, but the ways in which information should be displayed and indeed which information is needed will change. This is easily enabled by using a multi-platform application development tool, which allows you to choose how screens should display and act on different device sizes and operating systems.

4 Go Platform Independent

The final challenge in designing a mobile business screen is choosing which platform to build it for. After all, unlike the desktop ecosystem where for many years the screens just had to support Windows and Internet Explorer, secure in the knowledge that the next upgrade would be several years away and announced well in advance; in mobile there is a plethora of competing platforms, with new releases almost at random and BYOD schemes mean that a user could bring an obscure device to the enterprise and expect to be productive on it. While you want to support the wide variety of devices, you don't want to spend a lot of time and effort developing for a platform that becomes obsolete or rarely used in the near future.

However, enterprises don't necessarily have to make these 'all-or-nothing' choices. New platform agnostic development platforms overcome the need to try and anticipate where the mobile future lies. These engines use precompiled and preconfigured business logic that contains coding functionality and services. This allows developers to bypass the intensive code-writing stage and means that developers can work independently of the underlying platform. It's an approach which can also automate and synchronise a mobile application with other diverse applications, be it ERP or CRM, without the need for manual line-by-line scripting.

To create an application to be deployed on a mobile device there are four options available:

- Native language
- Hybrid apps (Native code with HTML and JavaScript)
- Web approach (HTML 5 and JavaScript)
- Mobile middleware platforms

1. Native Language Deployment

The advantage of deploying in a native language is that it provides a more suitable look and feel to the use, and therefore will be more readily adopted. However to support multiple device types, knowledge and support for at least 4 client languages are required:

- iOS for iPhones and iPads: Objective C (proprietary scripting mixed with C)
- Blackberry devices: J2ME (incomplete and inconsistent implementations of MIDP 2.0)

- Android devices: J2SE (incompatible with iOS and Blackberry)
- Windows based devices: Windows 8 (JavaScript, HTML5)

To further complicate the matter different versions of these languages exist and are continually released for example there are considerable differences between Windows 6.5, Windows 7 and Windows 8, not to mention the 'flavours' of Android.

Therefore this approach will require of multiple skill sets and the challenge of maintaining them.

2. Hybrid Approach

A hybrid approach has the benefit that an HTML application can be wrapped as a means of delivering to end users and app stores. But the disadvantage of skills requirement as above and the security issues associated with the current status of HTML5.

3. Web Approach

HTML5 is widely expected to become the technology of choice for future mobile application development, as mobile platforms become increasingly fragmented HTML5 is the latest iteration of the HTML standard, used for building and deploying web content. While previous versions of HTML were designed primarily for marking up text-based content, HTML5 is much more interactive, allowing web developers to take advantage of new capabilities such as 3D graphics rendering and gesture control without the need for plug-ins.

Most smartphones and tablets now come with built-in browsers, allowing developers to create sophisticated HTML5 applications that can be deployed on multiple platforms, rather than having to build separate versions for iOS, Android, Windows Phone, BlackBerry and so on. HTML5 can also be used to build hybrid applications, which be used to build hybrid applications, which behave like native apps but have HTML5 inside. Developing for multiple mobile platforms means learning the native language of different vendors, each of whom has a completely different stack, different testing, different development paradigms and processes; so the cost of developing multiple platforms natively is substantial. However, this does not mean the technology is ready for every implementation and, in particular, issues around security, synchronicity and the fact that it is an evolving standard can make it an unsuitable option for ISVs.

The real issue with HTML5 is that it is still HTML. This means that it is open to many of the same security vulnerabilities as previous iterations of the standard, including SQL injection, which is the number one risk to web applications. This technology was not built to take data from an enterprise and try to use that on the go. It was designed around content display, more to do with the viewing of pages, an organisational portal etc. It is not really designed well for a transaction-based area. Furthermore, it needs a large amount of bandwidth in order to synchronise the downloading and refreshing of different objects. If bandwidth is constrained – as is often the case when using 3G networks during peak hours or when signal is weak – objects can become misaligned. Take for example, a Facebook tag of someone might appear on the wrong photo. The implications of this can be particularly worrying in a business context. For example, if a business manager using a purchase order approval app on a mobile device receives the request to approve or reject a PO before the cost breakdown comes through, he could end up approving it without full knowledge of the facts.

5 Conclusion

This whitepaper has presented key principles of designing mobile business screens, from the perspective of the unique and varied challenges facing mobile users, and why these require the desktop experience to be re-engineered. Along with its companion piece, *6 Tips To Design Effective Mobile Screens For Business*, this whitepaper discusses the key challenges facing mobile users and designers, and offers advice on how to overcome them; however while this is a good start there is no substitute for experience and user testing. The most important considerations remain what users are trying to achieve, where and what devices they are using, alongside what user interface will best allow them to accomplish these objectives.

| About Magic Software Enterprises

Magic Software Enterprises ([NASDAQ: MGIC](https://www.nasdaq.com/markets/stocks/quotes/MGIC)) empowers customers and partners around the globe with smarter technology that provides a multi-channel user experience of enterprise logic and data.

We draw on 30 years of experience, millions of installations worldwide, and strategic alliances with global IT leaders, including IBM, Microsoft, Oracle, Salesforce.com, and SAP, to enable our customers to seamlessly adopt new technologies and maximize business opportunities.

For more information, visit www.magicsoftware.com.



Magic is a registered trademark of Magic Software Enterprises Ltd. All other product and company names mentioned herein are for identification purposes only and are the property of, and might be trademarks of, their respective owners. Magic Software Enterprises has made every effort to ensure that the information contained in this document is accurate; however, there are no representations or warranties regarding this information, including warranties of merchantability or fitness for a particular purpose. Magic Software Enterprises assumes no responsibility for errors or omissions that may occur in this document. The information in this document is subject to change without prior notice and does not represent a commitment by Magic Software Enterprises or its representatives.