

1

Introduction: Mobility is Different

A mobile phone is a Swiss Army knife. It is not a chef's knife or a buck knife. We keep wanting new features on the phone, like texting, voice memos, browsing, a camera, music, and television, because we would like these things in our pocket and the phone is already there.

And like a Swiss Army knife, the user experience of each of the features leaves quite a bit to be desired. A Swiss Army knife will not deliver the quality of cut a chef's knife will, nor will it fit in the hand quite as well as a good pocket knife.

Designing applications or web sites for mobile phones is in many ways the same as designing the best possible screwdriver or fishing rod for a Swiss Army knife. There is much that needs to be done before people will actually use the application – and people will not use the Swiss Army screwdriver in the same situations that they would use a full-sized screwdriver.

While the platform, user context, business context, device, and technologies involved in a particular mobile application may be different from similar desktop applications, the fundamental product design and development practices remain the same. The purpose of this book is to give product designers, software developers, marketers, project managers, usability professionals, graphic designers, and other product development professionals the tools they need to make the transition into the mobile arena.

This is not a book about technology or specific design recommendations. Instead, it focuses on the mobile users and their context.

It leans heavily on principles of human–computer interaction, usability, product development, business, and graphic design.

1.1 MOBILIZING APPLICATIONS

‘I don’t have a need for data services on my phone. Just give me a simple phone that has good reception and battery.’

I hear some variant of this from almost everybody to whom I talk about my work who is not actually in the mobile industry – although I grant that I do not talk to many teenagers about my work.

Focus groups show that real consumers are painfully aware that the web sites that they use not only would not work well on a mobile phone, but also would have little functionality or purpose. Most people are barely willing to read a long document or news story on a relatively comfortable full-sized monitor; it is difficult to know when or why a person would be willing to read the same story on a tiny screen. And willingness to pay for a service that provides text freely available elsewhere is even more rare.

This state of affairs, which is present in some degree in most of the world, is a result of some fundamental misunderstandings about what mobility means for customers and users. These misunderstandings cause the frequent failure of companies to create useful, relevant, enjoyable experiences.

Most mobile applications have been created as a miniaturized version of similar desktop applications. They have all the limitations of the desktop applications, all the limitations of the mobile devices, and typically some extra limitations due to the ‘sacrifices’ designers and developers make as they move applications from desktop to mobile device.

Some mobile applications have broken the ‘miniaturize’ trend and have enjoyed considerable success. While sound customization in the desktop environment is something done only by highly motivated users, phone ring tones have become a key component of the mobile user experience. FOX Network’s ‘American Idol’ television show allowed the audience to vote via text messaging, and text messaging even in the United States has become extremely profitable.

Text messaging is very popular (and profitable), especially in Europe, and most of Japan’s iMode traffic is actually similar short communications services. Sprint PCS did not have two-way text messaging in

its earlier offerings but developed a web-based similar product which fast became extremely profitable despite having never been advertised.

While there are several factors that these successful examples share, the most notable thing is something they do not share: they are not simply desktop applications ported to the mobile environment. A well-designed mobile application, to be successful, cannot simply be some subset of the corresponding personal computer (PC) application, but rather an application whose features partially overlap and complement the corresponding PC application's features.

1.2 WHAT IS 'MOBILE' ANYHOW?

The definition of 'mobile' is slippery. Visit the Consumer Electronics Show's 'Mobile' section and you will see a plethora of in-automobile media players, both audio and video. A laptop computer is certainly 'mobile' but is used more like a desktop computer.

Other attempts to apply a name to the field have used 'wireless', describing how the device communicates digitally. This again is problematic as more and more desktop computers are using wireless communications, as are automobiles, thermometers, and likely refrigerators in the future.

One of the earliest books on user-centered design in the mobile environment has used the term 'handheld', which wonderfully captures the essence of the size of the devices in question, but allows television remote controls into the definition.

Mobile phones epitomize mobile devices, but the category also includes personal data assistants like Palm, delivery driver data pads, iPods, other music players, personal game players like GameBoy, book readers, video players, and so forth. *Fundamentally, 'mobile' refers to the user, and not the device or the application.*

Further, this book is about the business and practice of mobile user experience management, not design for specific platforms. If you are designing a Palm application, go see a developer guide for PalmOS. If you are designing an iPod application, go see a developer guide for that platform. There are a number of mobile web and Java development guides available. These resources are invaluable.

To get entertainment and information services to the mobile user, some sort of communications device is necessary. Most target users of applications already have a mobile phone or other mobile communications device, which they carry with them most or all of the time.

1.3 THE CARRY PRINCIPLE

Of particular importance to mobile users are a special category of devices, namely *personal communications devices*, or PCDs. These are epitomized by mobile phones and text communications devices like the BlackBerry and Sidekick. The principles of design and management found in this volume apply to PCDs. In this book, the terms ‘mobile device’ and ‘personal communications device’ are used interchangeably. A PCD is:

- *Personal*. The device generally belongs to only one person, is personally identifiable, and has a messaging address and ongoing service.
- *Communicative*. The device can send and receive messages of various forms and connect with the network in various ways.
- *Handheld*. The device is portable. It can be operated with a single hand, even if two hands or a hand and a surface are more convenient.
- *Wakable*. The device can be awakened quickly by either the user or the network.

For example, a mobile phone will receive a text message even when in its ‘sleep’, or standby state. Note that most computers, if they are asleep, can not communicate with the network.

This combination of features makes the service indispensable and the PCD an ever present part of the user’s life. The service represents safety and social connection. Because the service is indispensable, users tend to carry the device with them all the time. This fact forms the core of understanding the mobile user experience.

The fundamental distinction between mobile-targeted design and design targeted for other platforms is *The Carry Principle*: the user typically carries the device, all the time. The Carry Principle has several implications on the device:

- *Form*. Devices are small, battery-powered, have some type of wireless connectivity, and have small keyboards and screens (if present).
- *Features*. Any information or entertainment features that might be desirable to have away from a computer or television, including television itself, will eventually get wedged onto the PCD. Devices evolve towards the Swiss Army knife model.
- *Capabilities*. The wireless connection, small size, and power constraints have made devices have slower connection speeds, slower processors, and significantly less memory than desktop computers.

- *User interface.* The small screen drives the device to a single-window user interface, so sharing information between applications is problematic.
- *Proliferation.* A personal, always-present device needs to match a user's needs, desires, and personality reasonably well. One form, one feature set, one user interface will not fit all.

The Carry Principle also has implications for the PCD users:

- *User availability.* The mobile user is more available for communications and application interaction than a computer user simply because the device is always present.
- *Sustained focus.* Because the user is doing so many things, there may not be sustainable time available for the device.
- *Social behavior.* Always-available connections has made attending meetings and dinner with friends a modified experience. Coordination across space allows both more and less social behavior.

Each of the above has implications for application design.

1.4 COMPONENTS OF A MOBILE APPLICATION

Any serious consideration of the design of software starts with a consideration of where the software will be used. Designers of web sites or applications intended for use on desktop or laptop computers tend to ask 'which operating system shall we target?', as computers are so standardized.

In reality, the *desktop environment* comprises a number of agreed-upon characteristics. All have a largish color computer screen of at least 800×600 pixels, a full keyboard, a mouse, speakers, and applications residing in windows. Connectivity may be slow (30 Kb/s) or fast (500 Mb/s or more), but it is generally there. In the US, landline network access is generally unlimited.

Further, the user of a desktop application is sitting at a desk or at least with a computer in the lap. There is a working surface, and both hands and attention are focused on the computer. Interaction with other people takes place only through the computer, not generally in person around the computer.

Devices in the *mobile environment* do not play by the same rules. This is not due to the lack of standards, but due to the highly varying

needs of mobile users. The differing capabilities of low-end mobile phones, high-end smart phones, and alternative devices lead to a variable environment. Expect this situation to continue for a long time.

A mobile application consists of:

- a *PCD*, with its own use metaphor, browser, application environment, and capabilities
- a *user*, using any of a set of mobile devices, who could be riding a train, sitting in a meeting, sitting in a restaurant, walking down the street, focused on other tasks, or engrossed in the device and application
- one or more *application platforms*, which can include web browsers, application environments (such as BREW, Palm, Windows Mobile, Symbian, or Java 2 Micro Edition), messaging technologies (including email, SMS, MMS, and instant messaging), media environments (types of music and video players), and so forth, with new capabilities becoming available regularly
- one or more *output interfaces* with the world outside the mobile device, including screen, speaker, infrared, Bluetooth, local wireless (Wi-Fi), cellular wireless, unique terminal identification
- one or more *input interfaces* with the world outside the mobile device, including (limited) keypad, touchscreen, microphone, camera, RFID chip reader, global position, infrared, Bluetooth, local wireless (Wi-Fi), cellular wireless
- optionally a *server infrastructure* that complements the mobile application and adds information or functionality to the above
- *interfaces* between the application's servers and other information sources
- a *network* and the corresponding *wireless carrier* (operator), who enables some of the above technologies, connects the user to the Internet and other users, sells applications and other services, may specify permitted devices, and frequently defines what may and may not be accomplished on the network

In contrast, an application delivered to a personal computer operates in a more predictable environment. Operating systems are limited to approximately three, rather than dozens. There is one browser markup language, and though there are rendering differences between browsers, they are trivial and readily handled compared with mobile browsing. Influence of any sort of the end user's ISP is unheard of. There are

definitely complexities associated with developing for the personal computer, but mobile is more complex in almost every dimension.

1.5 ABOUT THIS BOOK

This book is intended to help product design and development professionals make the transition from desktop to mobile with sophistication and understanding. It covers the obvious – devices, technologies, and users in the mobile environment – but goes further. *Chapter 2* discusses the characteristics of mobile users and how they differ from desktop users. *Chapter 3* presents a framework for understanding the range of mobile devices and how they fit into users' lives, then discusses the anatomy of the personal communications device. In *Chapter 4*, learn about various application presentation technologies and how to choose the best one for a project. *Chapter 5* covers general mobile design principles and sources of more specific design recommendations. Find sample mobile user interface design patterns in *Chapter 6*. Media generation for mobile is covered in *Chapter 7*. *Chapter 8* covers the various players in the mobile value chain, and their history, different goals, and typical decisions. *Chapter 9* discusses modifications of a user-centered design process for mobile applications, including modifications of user research techniques. *Chapter 10* discusses an example application, from concept to design and project management.

