

Developing User Assistance for Mobile Applications

Joe Welinske

2nd Edition

Developing User Assistance for Mobile Applications

Copyright 2014 © WinWriters, Inc.

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by an information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN 978-1-312-16946-3

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Publisher: Joe Welinske, WinWriters, Inc.

Cover Designer: Chris Barbee

Foreword: Tom Johnson

2nd Edition Editor: Joan Kalk

1st Edition Editors: Sue Heim and Karen Ziegler

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author nor the publisher shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

140919-C

To Shirley and Michael Sorenson —

*Cheerful and optimistic, they've always
made our family's little part of the world
a bit brighter.*

Contents

<i>Foreword</i>	<i>xi</i>
By Tom Johnson	
<i>About this Book</i>	<i>xiii</i>
The Author	xiii
About this Edition	xiii
What this Book is NOT About	xiv
Acknowledgements.....	xv
<i>GETTING STARTED</i>	<i>1</i>
<i>1 User Assistance in a Mobile World</i>	<i>3</i>
What Does It Mean to be “Mobile”?.....	5
There is a Role for UA in Mobile.....	7
The Mobile UA Skill Set	10
Take-away.....	15
<i>2 Understanding the Mobile Market</i>	<i>17</i>
The Smartphone Line-up.....	17
Native Apps Versus Web Apps	22
User Assistance Support for Mobile	23
Tablets	24
Using Simulators and Emulators	27
The Device Acquisition Conundrum.....	28
Take-away.....	30
<i>3 Key Issues for Mobile User Assistance</i>	<i>31</i>
Examples of Business Apps	32
Developing a Mobile Mindset.....	33
Discovery of Multi-touch Controls	34
Teaching Concepts and Contexts.....	36
The Four Cs.....	37
Limitations of Small Screens	40
Take-away.....	41

4	<i>Content Strategy for Mobile UA</i>	43
	Learning About Your Users.....	47
	Evaluating Your Options	52
	Planning for Implementation	57
	Take-away	63
	<i>DESIGN</i>	65
5	<i>UA for First-time Users</i>	67
	Feature Tours.....	70
	Overlays	74
	Take-away	79
6	<i>UA Embedded in the User Interface</i>	81
	Coach Marks	82
	Guided Help.....	87
	Hybrid Designs	92
	Embedded Help.....	94
	Take-away	96
7	<i>UA in a Dedicated Help Section</i>	97
	Help for Apple iOS Apps	98
	Help for Microsoft Windows Phone Apps	103
	Help for Google Apps.....	108
	Other Help	111
	Take-away	117
8	<i>UA in Other Forms</i>	119
	Stand-alone Help Apps.....	119
	Encapsulated Content.....	121
	User Interface Text	122
	Infographics	124
	Message Text.....	125
	Text Messages and Notifications	126
	Videos.....	127
	UA in Multiple Forms	128
	Take-away	128
9	<i>UA for Tablets</i>	129
	The Effect of Screen Size on Tablet UA	130
	Examples of Help in Tablets	131
	Other Tablet-specific Design Issues.....	136
	Working with the iPad.....	138
	Working with Android-based Tablets	140
	Working with Windows 8 Modern Apps.....	142

Take-away.....	145
----------------	-----

CONTENT CREATION..... 147

10 Writing for Small-screen Interfaces 149

Writing Procedures	150
Simplifying Language for Mobile UA	153
Improving Language in UI Elements	163
Take-away.....	168

11 The Language of Touch Gestures 169

An Overview of Touch Language Considerations ...	171
Applying Touch Language to UA	174
Writing About Physical Controls.....	179
Take-away.....	180

12 Images and Videos.....181

Representations of UI Elements.....	181
Movement and Process	182
Use of Fonts	183
UA Through Visual Representations	185
Screen Resolution.....	189
Video Tutorials.....	190
Take-away.....	197

PLATFORMS..... 199

13 Web Apps, Browser-based Help, and Web Views..... 201

Browser-based Help	202
Designing a Browser-based Help Template.....	204
The Web View.....	206
The Viewport <meta> Tag.....	208
Media Queries.....	210
A Sample Design Process	212
Context-Sensitive Help	215
Delivering Your Browser-based Help.....	218
Take-away.....	220

14 iPhone, iPad, iPod Touch, (iOS)..... 221

Hardware/Software	222
Knowledge	223
Xcode and Interface Builder	223
Learning About Objective-C.....	233
Learning About WebKit / Safari.....	234
Working with a Project in the Simulator	235

Importing a Provisioned App.....	236
Testing Accessibility with iOS Apps.....	237
Screen Captures	239
The Safari Debug Console on iOS	240
Localization	241
Miscellaneous Resources	242
Take-away	243
15 Android	245
Hardware/Software.....	245
Installing the Android SDK.....	249
Android Virtual Devices.....	252
User Interface Text	255
Screen Captures from an Android Device.....	256
Testing Accessibility with Android Apps.....	256
Android-based Tablets.....	256
Miscellaneous Resources	256
Take-away	257
16 Windows Phone	259
Hardware/Software.....	261
Installing the Windows Phone SDK	262
Working with UI Text	265
Screen Captures	272
Testing Accessibility with Windows Phone.....	272
Miscellaneous Resources	273
Take-away	273
PRODUCTION	275
17 Single-sourcing to Multiple Devices	277
Responsive Web Design.....	281
Supporting Interaction Verbs	285
Take-away	288
18 Customizing Content	289
Adaptive Content	293
Translation Issues.....	295
Generic vs. Custom Language.....	296
A Hybrid Approach to Responsive Design.....	297
Responsive Content Using MadCap Flare.....	300
Take-away	303
19 Development Tools	305
Help Authoring Suites	305

Editing and Prototyping Tools	315
Testing Tools	319
Take-away.....	324

EXTENDING YOUR REACH 325

20 Integrating with Other Types of User Assistance 327

Product Web Sites.....	329
Blogs	335
Simulations.....	336
Social.....	338
Web-based Forums and Knowledge Bases.....	340
Email Support.....	342
PDFs	344
Integrating with a Desktop Application	346
Take-away.....	349

21 Emerging Technologies and Platforms..... 351

Emerging Interaction Types.....	352
Voice Help	355
Automotive – Help Behind the Wheel.....	358
Wearables and the Quantifiable Self	364
Games.....	366
Home Entertainment	368
Desktop App Stores	369
Take-away.....	371

22 Translation Issues for UA..... 373

Tap	376
Double-tap.....	376
Tap and hold.....	377
Flick.....	377
Swipe.....	378
Drag	378
Press and hold	379
Rotate.....	379
Pinch to Zoom.....	380
Press the Home button.....	381
Go to.....	381

Index..... 383

Foreword

By Tom Johnson

It's somewhat surprising how ubiquitous smartphones are among tech writers (almost everyone has one), but how infrequently tech writers engage in user assistance for these same mobile devices. Why aren't technical writers saturating the exploding mobile market by playing key roles with documentation on mobile development teams?

One reason might be a question of scope and complexity. Companies who hire professional technical writers tend to have significantly complex content to document, requiring the full-time attention of technical writers to gather, create, and publish the information. Mobile apps, on the other hand, tend to be much simpler and easier for users to understand. There's only so much you can do tapping a small screen with your thumbs. The amount of help content is also substantially smaller and briefer.

But there's probably another reason why we don't see more technical writers in the mobile space: tech writers tend to stick with information development, creating content in their own spheres and handing it off to dev teams to link to.

More than anything, Joe's book will suggest that for tech writers to thrive in the mobile space, their roles must go beyond documentation. The technical writer in the mobile space plays a UX role, integrating help content at key points of interaction in the user interface. Joe argues that the best help is not a separate page that opens up outside of an app, but rather appears in the UI at the moment the user needs it.

Only when tech writers start tearing down the artificial barriers between the help system and the application will they begin to create successful user experiences in mobile. Joe shows a wide variety of approaches for integrating help in mobile interfaces, grouping the approaches in different patterns and getting into the nitty gritty details of actual apps. He doesn't wander in theoretical abstractions. Instead, you're immersed in a visual panoply of app screenshots showing a wide variety of help interactions.

He builds his analysis from his immersion in apps and gives you details only an experienced professional, aware of all the nuances and intentions and effects of help techniques, can provide.

Beyond showing patterns of help integrated in mobile interfaces, Joe also provides a wealth of technical detail that will be immediately helpful to anyone getting his or her hands on code. Whether you're working with iOS, Android, or Windows Phone, he gives you step-by-step details for setting up emulators, working with web views, installing SDKs, configuring development environments, and more.

Giving so many technical details may seem a bit risky given the constantly changing nature of the domain, but those details might also be the most immediately helpful as you ramp up technically to move beyond just being the doc person to being an actual team member involved in UI development.

Without a doubt, Joe's book is the definitive work on user assistance for mobile. But as he points out, it's not just about mobile. Although the examples are grounded in the mobile space, it's not a stretch to envision similar techniques for desktop help, or to consider how desktop applications will evolve into the mobile space in more seamless ways — taking their help with them.

— *Tom Johnson is a technical writer in San Jose, California, mostly writing developer documentation, including documentation for APIs. He publishes the popular blog I'd Rather Be Writing. <http://idratherbewriting.com/>*

About this Book

The Author



Joe Welinske has been involved with software documentation development since 1984. He founded and produces the WritersUA Conference for Software User Assistance which has drawn thousands of attendees over its twenty-some years. Together with Scott Boggan and David Farkas, Joe authored the popular and pioneering book

Developing Online Help for Windows 95.

He has taught at the University of Washington in the Department of Technical Communication, the University of California at Santa Cruz, and Bellevue Community College. Joe served as President of the Puget Sound Chapter of the Society for Technical Communication and as Membership Director for the Puget Sound Chapter of the Usability Professionals Association. He was awarded the Myron L. White award as an International Leader in Technical Communication by the University of Washington. He is an Associate Fellow of the Society for Technical Communication.

Joe received a B.S. in Industrial Engineering from the University of Illinois in 1981. He received a Master of Education specializing in Adult Instructional Management from Loyola University in 1987. In his spare time, he likes to travel, kayak, and write.

About this Edition

An energetic attempt was made to have the latest screen shots and information in here. Frankly, it is a tough proposition to keep up with the rapid iteration of the various mobile platforms and apps. Some things are probably out-of-date as of the date this book is published. Hopefully, the examples can still stand on their own with respect to the issues that are discussed.

If you have a printed, black and white version of this book, you won't see the color and detail of the many screen captures. The mobile site listed below has the original digital versions of all the images.

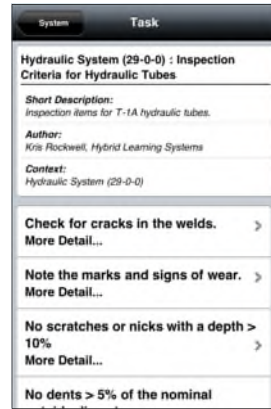
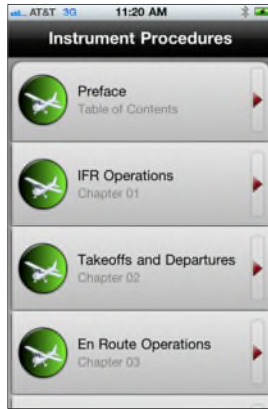
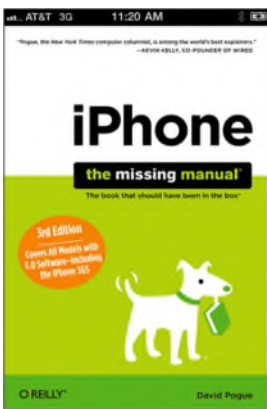
The printed book also limits your ability to benefit from the many resource links. Those are also available as live links in the mobile site.

Note → The web site for this book offers full-color versions of all the images as well as live links to all the resources.
Developing UA for Mobile Apps site:
<http://www.writersua.com/duama/>

What this Book is NOT About

Many organizations are employing smartphones and tablets as a viewer for reference information such as that shown in the figure below. For example, a field technician working on a piping system might have all of the latest service manuals available on an iPhone. Or a sales rep might have a list of equipment specifications in an app on a Samsung Galaxy. An HR manager might carry around all of the company policy and procedure manuals on a tablet. There are many great reasons to deliver general documentation through mobile devices. However, that is a different category of technical communication from software user assistance (UA). This book doesn't address that category specifically, but many of the techniques are relevant to it.

General reference materials on mobile devices



Acknowledgements

Many people have contributed to the development of this book. I've relied on thoughtful discussions with these colleagues: Bernard Aschwanden, Dave Farkas, Dave Gash, Gordon Meyer, John Alwein, John Daigle, John Fisher, Joan Kalk, Kelly Franznick, Kevin Siegel, Marta Rauch, Matt Sullivan, Matthew Ellison, Misty Weaver, Neil Perlin, Nicky Bleiel, Paul Mueller, Paul O'Rear, Peter Grainge, Scott Boggan, Scott DeLoach, Scott Prentice, Steve Stegelin, Tony Self, Whitney Quesenbery, and Willam van Weelden.

Several people took the time to read through the manuscript to give me suggestions: Rhonda Bracey, Andreea Chirita, Frank Ripp, John Collins, Rochel Weitz, and Thomas Risser. Linda Urban gave me a nice push to add more detail when I started to get lazy.

I really appreciate all of the great support!

Getting Started

Chapter 1 – User Assistance in a Mobile World

Chapter 2 – Understanding the Mobile Market

Chapter 3 – Key Issues for Mobile User Assistance

Chapter 4 – Content Strategy

1

User Assistance in a Mobile World

Preview

This chapter introduces the concept of user assistance for mobile apps, including a discussion of what it means to be mobile. It also highlights the skills involved in developing effective mobile user assistance.

The ubiquity of cell phones means that we all have experienced mobile voice calling and texting: instant communication with anyone, anytime. We are used to getting what we want extremely fast and in just the way we like. Most consumers have brought those same expectations with them as they move to smartphones (Figure 1.1), tablets, and mobile applications (apps). Ownership of smartphones passed the 50% threshold by mid-way through 2013, according to comScore and other organizations.

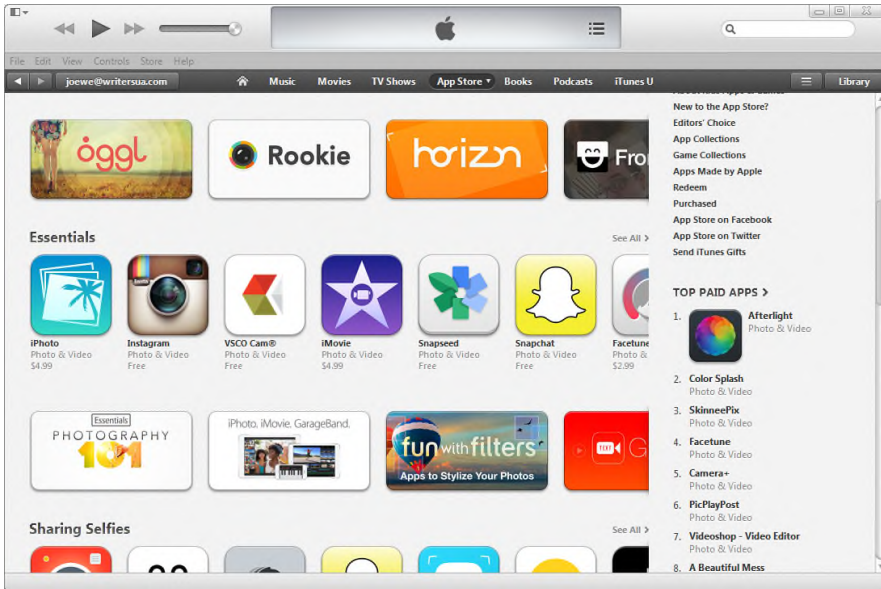
Figure 1.1 Today's smartphones



There is no question that mobile devices have sparked a huge, new software segment – the mobile app. The associated apps are all about speedy access to discrete actions and specific information. This is a very different environment from the fixed workstation, large-screen desktop environment of traditional enterprise applications.

The Apple iTunes App Store (Figure 1.2) hosts over a million apps that have been downloaded tens of billions of times and generated billions of dollars in revenue. Google’s Android operating system is now in the hands of more consumers than any other smartphone and it also has over a million apps in its inventory. Windows Phone has had less success, but continues to grow. There are not many computer/software/IT organizations that don’t have a mobile strategy for their products and services.

Figure 1.2 iTunes App Store



The independent Pew Research Center wrote in a 2008 research paper,

“The mobile device will be the primary connection tool to the internet for most people in the world in 2020.”

That prediction appears to be spot on. If anything, the world may already have reached that tipping point. More importantly for us, the last several years have seen a very rapid conversion in the IT industry in the way software is designed, implemented, and consumed. This creates an important pair of questions for you as a user assistance professional: What is your role in mobile and how can you prepare to take that on?

This book is based on practical experience working with a variety of mobile apps and their associated development frameworks. It is a distillation of challenges and solutions encountered with respect to design, technology, and implementation. The work has included iPhone, iPad, Android, Windows Phone, and web apps. Hopefully this book will prove to be a useful guide for the user assistance community.

What Does It Mean to be “Mobile”?

The term “mobile” is widely used among software developers and the mainstream consumer when talking about today’s computing devices. But what does it really mean to be mobile? The answer isn’t as simple as you think. The 2014 WritersUA Skills & Technologies Survey (Figure 1.3) asked UA professionals what platforms they currently support. Two out of five respondents (39%) indicated “Mobile” as one of those platforms. That number has been steadily increasing over the past five years of this survey.

Figure 1.3 Platform Support for UA Professionals - WritersUA Survey

System	Response
Windows (1) (v.8, 7, Vista)	83%
Windows (2) (XP, Server 2003/2008)	64%
World Wide Web	64%
intranets/extranets	44%
Mobile	39%
Linux	31%
Mac OS X	26%
Java	22%
Windows (3) (2000, NT, and earlier)	21%
UNIX	20%
IBM mainframe	4%
OPEN VMS	3%

- ▶ **2014 WritersUA Skills & Technologies Survey**
<http://tinyurl.com/mobileua-survey1>

But what is it that makes a device fit the mobile label. Laptops have been around for two decades now and their main characteristic to distinguish them from other computers is that you can easily carry them anywhere. Despite that, most of us don't necessarily think of laptops as fitting the mobile label. Rather, it is phones and tablets that come to mind as mobile. Why is that? What is different about this category of devices? There are three elements that distinguish what most of us mean when we say a device is mobile:

- They are always connected to our data and services through the air - through cellular coverage or a wifi connection.
- Phones and the more popular mini-tablets are pocket and purse-sized. We take them with us without thinking twice about it.
- Software services are provided through a seemingly limitless parade of compact, easy-to-download, inexpensive or free apps.



Understanding how to effectively design mobile devices means understanding how to do the most within the constraints of those three elements. Where things get interesting is when you realize that the popularity of mobile is driving the development of traditional and emerging devices. We see products in very different market categories all rushing to make themselves more like “mobile”, as shown in Figure 1.4.

- Microsoft's **Surface** is a compact, SIM-based, touch device that runs Windows Store apps.
- **Roku** uses popular apps like Netflix as channels to deliver streaming video through a device the size of a hockey puck.
- Ford **Sync** is an in-dash automotive entertainment system that is always-connected and hosts popular mobile apps.
- Samsung **Galaxy Gear** brings messaging and apps to your wrist.

The bottom line is that mobile design is increasingly important to all of us – regardless of whether or not the software we support can be found in iTunes or the Google Play Store.

Figure 1.4 Surface, ROKU, Sync, Galaxy Gear (clockwise from top left)

There is a Role for UA in Mobile

There is no question that there is a role for user assistance in the mobile world. Our skills and experience are definitely applicable. But the nature of many of our future deliverables is likely to be very different from what we have done in the past.

The single most important lesson of this book might be that bringing over Help designs from desktop applications is most likely going to result in a less than optimal result. The most useful and interesting user assistance approaches are the ones that try to match the environment of the apps they support.

This book showcases several dozen examples of mobile user assistance. The following online resources provide several hundred screen shots of UA on Android, iOS, and Windows Phone. A Mobile User Assistance Screen Shot Gallery is available on the WritersUA site. Gordon Meyer of Apple has put together a nice collection of “Help” for iPhone apps. The inspireUX site highlights some useful UA designs. One of the few unifying characteristics of the many examples is that they don’t use designs brought over from desktop applications.

- ▶ **Mobile UA Screen Shot Gallery**
<http://tinyurl.com/mobileua-gallery1>

▶ **Gallery of iPhone Help**

http://www.g2meyer.com/gallery/main.php?g2_itemId=5026

▶ **inspireUX Help Design Patterns**

<http://tinyurl.com/mobileua-inspireux>

Your first important step toward your own design should be to evaluate existing designs to determine which approaches you think work and which don't. Develop your own pattern library and refer to that for inspiration as you begin new projects.

The iPhone, Android, and Windows Phone are successful because the designers recognized that the mobile user environment is very different from the desktop. Mobile app developers tend to be very sensitive to the differences between mobile and desktop software. If you try to bring your desktop Help designs into a team meeting with experienced app developers you may find a lot of them rolling their eyes.

As user assistance professionals, we need to break away from the traditional online Help designs and consider what it means to be mobile. The best mobile UA is going to be where the UA content is customized, as closely as possible, to the user experience of the individual apps. This may not always be possible, but it should definitely be the ideal.

Only after you have analyzed personas, use cases, feature priorities, and first-time scenarios, can you make a reasoned decision on whether or not you need to consider some form of user assistance. Assuming the answer is to do so, you now need to consider all the many UA design choices that may or may not be appropriate to meet your objectives. Your careful implementation of the UA solution is what ultimately determines whether you have helped your users be successful in working with your app. User assistance in mobile apps is common and the designs are as different as the apps themselves.

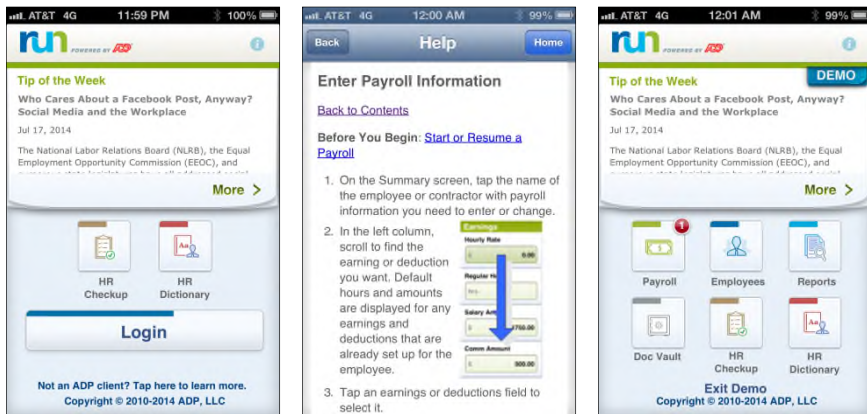
While most mobile apps are easy to use, some have more features than others. For example, popular apps like **Snapseed** and **Pro Camera** provide iPhone users with a strong set of tools with which to manipulate images. Assuming customers are interested in those tools, you now need to make a more nuanced decision about the value of adding some form of user assistance. Snapseed uses a walkthrough that is opt-in and available throughout the app. Pro Camera embeds Help text with the various advanced settings.

We, as designers, also have to realize our personal experiences are mostly from within a tech-savvy bubble. A large percentage of smartphone users have likely skipped working with laptops/desktops completely. The whole

idea of interacting with data may be new to them. For example, take an older user who has a new tablet but has never used a computer before. In this case, an iPad app like the Chicago Tribune has to not only explain the controls of the app but also introduce the customer to the concept of online versus print consumption of news. The Tribune uses what is called an “overlay” to point out the key elements of the digital page.

Any discussion surrounding whether or not UA is needed in apps should not ignore the growing number of enterprise-specific apps. You can find apps for most of the Fortune 100 in the iTunes store through Search. Most of these apps are for employees only and it isn't too much to ask them to learn how to use the apps effectively. **Run ADP** (Figure 1.5) is an Android/iPhone app that gives payroll administrators many of the same capabilities that they have with the desktop application. The app provides introductory videos, a demo mode, and traditional Help. With million-dollar payrolls at stake, it makes sense to provide layers of user support.

Figure 1.5 Run ADP app



The bottom line should be a mobile user experience that meets the needs of a variety of user types and use case scenarios. Thoughtful, appropriate, and well-designed user assistance is one of the elements in achieving that goal.

A Contrary View

The opinion of some app designers is that the use of user assistance indicates a less than optimal effort in designing the app – that the need for Help information signifies a failure. The following arguments against UA in apps are often brought up:

- Apps, by definition of being apps, should be limited in features. You don't need help if the app is constrained in the first place.

- Too much information impedes the use of the app. Limited screen real estate means there is no room for Help text.
- Help information unnecessarily adds to the footprint of an app, taking up precious device storage and memory space.

There may be some anecdotal evidence for these positions. However, it is very difficult to make valid pronouncements on good vs. bad when you try to apply them to thousands of apps and millions of users. The individual nature of apps and their associated audience personas and use cases are likely going to vary quite a bit. It is easy to talk about designing for “mainstream” users, but do we really know what that means in a general sense? We can certainly decide what mainstream means with respect to our own user base and define and test our designs against that definition.

To assume that apps need to be simple is just simplistic. While many successful apps use that approach, there are equally successful apps that are more robust. With faster processors, more memory, larger screen real estate, and higher resolutions, apps have a foundation that is fully capable of handling more complex activities. Many tablet apps are more robust than their phone-based cousins due to the increased screen real estate.

The Mobile UA Skill Set

If you have been involved with user assistance for traditional software applications, you are in a good position to move into support for mobile UA. If you are new to the discipline, you will find it useful to learn the fundamentals of technical communication. While mobile apps represent a new way of thinking about software development, most of the skills involved with desktop application development are still very relevant.

Designing the UA

Many of the general categories of user assistance for desktop applications apply to mobile. These include tutorials or guided assistance for first-time users, embedded helpful information in the user interface, and the creation of a section of more detailed procedure and reference information. An app might use one or more of these approaches. The difference from traditional UA lies in how those items are executed or accessed.

The *most* important design skill might just be thinking outside the box. Apps represent a significant break from decades of software design. The best apps creatively envision new ways for people to work on the go. Our user assistance needs to match that creativity by moving away from the legacy designs of fixed workstation computing. One of the least useful entities is the traditional “Help system” with its navigation hierarchy and repository of topic pages.

Creating content

There are countless quotes by famous people along the lines of Mies van der Rohe’s “less is more.” This concept definitely applies to mobile UA. Effective UA for apps is more about crafting words and phrases and less about generating volumes of content. Large-scale documentation probably doesn’t have a place in the mobile world. That being said, our foundation skills of writing, editing, task analysis, and subject matter expert (SME) interviewing are still extremely important.

In mobile, effective writing is about spending more time coming up with precisely the right words. During the editing process, the emphasis must be on strictly limiting the volume of text while maintaining quality and usefulness. Knowing the customer and the task at hand are still critical to developing the right UA approach. Procedures, reference information, wizards, embedded Help, and user interface text are still key elements. They just have a very different form factor in mobile.

Knowledge of information design and content management is going to help your overall mobile UA effort. However, many of the processes used to support traditional software development may be too unwieldy for mobile purposes. For example, mobile UA emphasizes embedded, brief content. It would be difficult to manage this with most of today’s content management systems. In the coming months and years, keep an eye out for evolving best practices on this subject.

The use of well-crafted images and video is definitely part of today’s mobile UA. This is an area on which to spend more time honing your skills.

Indexing and search techniques are of less use in mobile. If a user needs to search for Help content, then the user assistance material is probably far too extensive.

Generally, a mobile help system with topics, indexes and table of contents is difficult for a user to process. For most apps, the priority should be on context-sensitive, embedded content. When more in-depth information is

needed, it can be layered and extended to websites, forums, and wikis, where search, indexing, and a table of contents is more easily integrated.

Building digital deliverables

For much of the past two decades, an important aspect of user assistance has been converting our designs and content into a digital format for consumption by our users. This includes creating compiled HTML Help, support pages on a web site, or embedded text in the user interface. While the look and feel of these deliverables has changed with mobile apps, ultimately we are still working with software.

There are a couple of key process differences from desktop application development. One is the rapid iteration/evolution of operating systems. The other is the high-level of control imposed by platform gatekeepers on how apps are compiled. What this boils down to is a constantly changing environment for how mobile apps are designed and built.

As with much of mobile development, it works best when designers and developers work in concert. You don't want coding to dictate the design. On the other hand, there are always governing conditions and resource tradeoffs to consider. For most of the techniques described in this book, you are going to have the best results when working hand-in-hand with your developers.

Experience with HTML, Cascading Style Sheets (CSS), and JavaScript continue to be important. A focus on HTML5 is critical to working with the mobile platforms. The iPhone/iPad eco-system is already highly dependent on HTML5. Using CSS helps to minimize the size of our code that is stored in an app or delivered over the air. It can also facilitate a design that adjusts for different screen dimensions. Also, CSS is the clear path forward for animation and simulations. Adobe's Flash has been shut out of the Apple eco-system. The Flash player was also removed from Android in 2012.

Understanding how apps are built can really make the difference in how effective you are in your UA efforts. The technical limitations and opportunities of mobile apps are very different from desktop computing. Programmers are really good at sniffing out whether we have any clue as to the challenges involved in building software. It is to your benefit to learn as much as you can. Fortunately, the development tools for mobile are very accessible and relatively easy to use. Several chapters in this book get into the specifics of turning your content into digital deliverables.

It is always a very good idea to learn a programming language. Once you've learned one it is easier to get at least a working understanding of others. Understanding how programming works makes it easier to communicate with developers. It also gives us the chance to dig into code on our own when we need to find something. Objective-C and C are the languages of choice for iPhone/iPad. Java is the bulwark of Android. Windows Phone is based on C# with graphical apps relying on Direct3D.

An important skill for mobile UA is usability testing. Even the smallest elements of an app can have a disproportionate effect on whether a customer enjoys the app or not. The very fact that mobile devices are mobile makes it easy to present apps to people for testing. The simulators available for the various mobile platforms provide a great way to make quick user interface iterations – largely without the assistance of a programmer. There is a huge body of information available about what usability testing is and how to do it well.

- ▶ **User Experience Professionals Association**
<https://uxpa.org/>

Other areas of interest

One of the fun things about user assistance is that there are so many peripheral disciplines that align with our core competencies. The following areas of interest are discussed in this book to various degrees. Some of these areas map to mobile UA more than others. You will want to cast a wide net to capture all the relevant developments in these fields as they relate to mobile.

Localization/Translation: The mobile app platforms have a variety of language strategies but they might not easily fit in with existing desktop production schemes. Text expansion due to different languages is an important consideration. Also, translating the language of touch interactions is very new.

Instructional Design and eLearning: The use of tutorials and videos definitely fits into the mobile arena. The difference is scale. How do you embed training of content into a mobile environment? How do you adjust length and detail for an on-the-go experience?

Agile Development: The creation of mobile apps is fast-paced and highly iterative – almost by definition. If anything, a team needs to be even more tightly integrated when putting together a mobile app.

Structured Authoring / DITA: Organizing our valuable content is always a good thing. Separating content from presentation also has a lot

of merit. The question is whether a process like structured authoring or a framework like DITA is useful in a setting where large-scale documentation is ill-advised. A related issue is if, and how, traditional content management systems can be adapted to support mobile development.

Help Authoring Tools: The very different nature of UA for mobile means that we need to look to very different types of tools for creating that UA. The Help designs for desktop applications are generally not suitable for mobile UA. Tools that only support those kinds of designs are not likely to be very useful.

Social Media: There are many ways that user assistance can employ the collaboration and community building of Twitter, YouTube, Facebook, and similar entities. Mobile apps are generally always “on-line” so it makes sense to integrate social media wherever appropriate.

Simplified English: A system of best practices originally developed for the aircraft industry, Simplified English offers a method to reduce complexity in technical language. Many of the principles regarding the choice of words and the length of phrases can be applied to writing for the small-screen of mobile devices.

Take-away

User assistance has an important role in the development of mobile apps. However, it may be difficult to identify opportunities right now. It is important to look beyond mainstream apps and search out apps tailored to specific business contexts. Keep a look out for emerging mobile development in your own organization.

As you begin to work with mobile apps, try to disengage from legacy design and implementation techniques from the desktop world. Look at this as a completely new medium and tailor your UA accordingly. There are many unique challenges including screen real estate and multi-touch gestures.

Your past experience with UA will be extremely valuable. Fundamentals of communication still apply to mobile. Usability testing and user research may be the most important skills to acquire and hone.

Best Practices

- ❖ Be constantly aware of the speed with which mobile apps are beginning to dominate the software industry.
- ❖ Consider what it means for an app to be “mobile” and adjust your UA design appropriately.
- ❖ Understand that mobile UA applies many of the same skills as traditional UA, but with many new techniques.

2 Understanding the Mobile Market

Preview

This chapter discusses the current landscape of smartphones, tablets, and other mobile devices. It offers recommendations for acquiring and upgrading devices for your own development needs.

Since the first edition of this book was published in 2011, the percentage of mainstream consumers using smartphones has increased significantly. An August 2013 report by Gartner reported that global smartphone sales exceeded feature phone sales for the first time. “Smartphones accounted for 51.8 percent of mobile phone sales in the second quarter of 2013, resulting in smartphone sales surpassing feature phone sales for the first time,” said Anshul Gupta, principal research analyst at Gartner.

Most of us in the tech industry now own a smartphone and/or a tablet. If you don’t have one you should be considering getting one. It will be very difficult to get serious about working with mobile software unless you have some personal experience with the devices, apps, and ecosystems. Developing Help for a mobile app without having a smartphone would be like creating Help for a desktop app if you could only view the app on a TV. You can probably get it done but it is not likely to result in great user assistance.

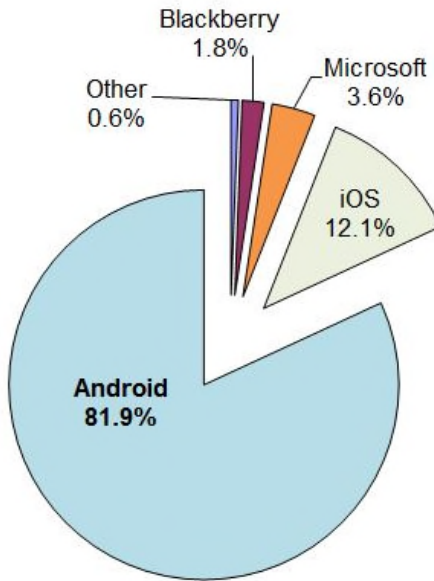
The Smartphone Line-up

If you are buying a smartphone mainly for your own personal use then the world is your oyster. There are dozens of new choices appearing every month. However, if you’re buying one with an eye on learning about

mobile UA, you need to give it a more focused consideration. You'll need to balance the high cost of the devices and associated service contracts against the value it can provide for your professional use. Hopefully the organizations that employ us will start buying devices for us to work with, just as they provide us with desktop workstations. This is already the case in companies like Amazon and Microsoft. They have in-house inventories of all the popular devices which designers and developers can “check out” for test purposes.

Figure 2.1 shows a snapshot of global market share for smartphones as of Q3 2013.

Figure 2.1 World smartphone market share (Gartner, Q3 2013)



► Worldwide Smartphone Sales to End Users

<http://www.gartner.com/newsroom/id/2623415>

The Apple family of devices – iPhone, iPad, iPod Touch – has seen its share shrink to a third of what it was in 2011. Collectively called “iOS”, these devices are highly profitable and the overall growth in mobile has kept Apple at the top of the pecking order in terms of revenue.

However, the biggest story of the past two years has been the stunning growth of Android. Emerging from Google as an open-source project, Android has come to command a huge market share in smartphones.

Breaking things down by country you get a somewhat different view, as shown in Figure 2.2. Outside of the U.S. we find Android maintaining

high numbers. Windows Phone is much more popular in France, Great Britain, and Italy than in the U.S. According to a Kantar World Panel report in early 2014, Windows Phone has grown its market share in Europe to 9.7%. It is in double-digits in Great Britain and France and has overtaken iOS in Italy. Figure 2.2 is drawn from the figures posted on that site in early 2014.

Figure 2.2 World smartphone market share by country (percent)

	Android	Blackberry	iOS	Windows	Other
Australia	57.7	0.3	35.1	5.1	1.7
China	80.9	0.2	17.4	0.7	0.8
France	65.2	1.8	22.2	9.6	1.2
Germany	75.1	0.8	16.2	6.8	1.1
Great Britain	55.2	2.4	30.7	11.3	0.3
Italy	67.3	2.1	12.6	17.0	1.0
Japan	30.5	0.0	68.7	0.0	0.8
Spain	86.6	0.0	7.2	5.3	1.0
USA	54.8	0.6	38.9	5.0	0.7

► **Smartphone OS Market Share - Kantar Worldpanel**

<http://tinyurl.com/mobileua-kantar>

The key statistic for many software vendors, however, is app revenue. While Android and iOS offer roughly the same number of apps, the iOS App Store has brought developers more than twice as much revenue as the Google Play Store, according to an October, 2013 report by Distimo, an analysis site for the app market. Collectively, app store downloads accounted for over \$26 billion in sales in 2013.

Apple's iOS

The mobile ecosystem from Apple is called iOS. This encompasses the iPhone, iPad, and the App Store. iTunes has been a key piece of technology for many years. While iTunes is still widely used, the iOS devices can download and sync data independently. Because the big money in apps is happening in iOS, that is where most developers are putting their first efforts. Unless your organization is specifically ignoring iOS, it is the best place to start learning about mobile UA.



Apple has a pattern of maintaining a relatively short list of devices which makes it pretty easy to keep up with mobile developments in iOS. At the time of this writing, the main devices in the line-up are the iPhone 5s, 5c, and 4s. Earlier versions are still available from phone companies at

discounted rates. Apple aggressively pushes their customers forward to update on a regular basis.

The 5s and 5c use the same chassis and share much of the other hardware. The 5s includes newer features like finger-print security and a faster chip. The 5c has been marketed as a lower cost alternative. The iPad and iPad mini have evolved with the major changes being in weight, thickness, speed, and display resolution. The lack of radical changes in the iPad design might be a contributor to flat sales in 2013.

Google's Android

The growth of Android over the past few years has been nothing short of remarkable. In 2011, Android had just passed Apple in terms of the overall installed base of users for smartphones.

In the ensuing time period, Android has absolutely dominated worldwide sales with four out of every five devices sold. Android experienced a 236% year-over-year increase in app installs for 2013-14.

Android also has 62 % of the global tablet market share.



However, that number is fragmented amongst hundreds of devices and a half-dozen versions of the operating system. For developers, the tools and the marketplace are not quite as attractive as iOS. Android has been a less likely place to look for apps that connect to traditional enterprise applications – but that is changing. It is safe to say that most enterprise efforts in mobile development are going to consider Android just as important as iOS.

Early versions of the Android development tools were very tricky to work with compared to iOS. The upgrades in 2013 brought Android a set of really good tools. If you want to explore mobile apps in a relatively unrestricted, but fast-moving, ecosystem this is the place to be. There are also loads of inexpensive, unlocked, used Android phones available for sale.

Microsoft's Windows Phone

Windows Phone has not had anywhere near the level of success of Android and iOS. The relatively small market share for Windows Phone is mainly a legacy of being way too late in responding to the capacitive touch screens and app ecosystems of iOS and Android.

The user interface has received mostly favorable reviews. It definitely presents an innovative point of view about how to interact with a smartphone.



Most of the success of Windows Phone is found in the line of phones from Nokia. The figures in the April 2014 AdDuplex report show how dominant the line of Lumia phones is.

- ▶ **AdDuplex report on Windows Phone**
<http://tinyurl.com/mobileua-addduplex>

The purchase by Microsoft of the phone division of Nokia should continue to boost the momentum of the platform. Nokia had already discontinued its development efforts with the mobile Symbian OS. That platform will be maintained for legacy purposes.

It is likely that Windows Phone will continue to cement its position in third place after Android and iOS. If your company currently develops Windows applications, you may want to start your efforts with Windows Phone. The Windows Phone developer tools from Microsoft are easy to use and free.

BlackBerry

2013 signaled the end of BlackBerry as we knew it. The long awaited new operating system was met with dismal sales. While many people still get a BlackBerry from their employer, it is a platform with a very limited future. BlackBerry's secure messaging and email system continues to be valued by enterprise. However, that is likely to morph into a service that won't be dependent on BlackBerry hardware. Third-party app development for BlackBerry is more or less dead in the water. Getting a BlackBerry isn't going to help you learn much about mobile UA in the near future unless your company is designing apps for it.

Native Apps Versus Web Apps

One of the important considerations is whether you will be supporting what are called “native” apps as opposed to web apps. The decision on the type of app will be made in the early planning for the app. For many UA professionals, this decision will often be made before they even get involved in the project.

Native apps are platform-specific applications. They use proprietary frameworks and programming in a compiled wrapper. Apps for iOS, Android, and Windows Phone are all considered native. They follow a regimented development formula. These kinds of apps do not work cross-platform. In the case of iOS and Windows Phone, these apps are also vetted for quality and content. All of the statistics in the previous section were related to native apps.

Web apps are simply web sites that are optimized for use on mobile devices. These apps use the same open source technologies that have been around for many years, including HTML, CSS, JavaScript, etc. Web apps can run on any device that has a web browser. There is no authority controlling how these apps are created beyond the web standards themselves. Web apps do not have the high profile that native apps have and so it is hard to find statistics on their overall usage.

The most popular and visible apps are in the native category. Generally, native apps enjoy better performance than web apps since many of the functions are drawing upon the local, internal resources of the device operating systems. Web apps are making constant queries to a remote server for all their operations.

Native apps benefit from established app stores for distribution. Apple’s App Store, Google Play, and Windows Store provide an easy way for users to find, purchase, and install apps on their mobile devices. Web apps require their own distribution mechanism. This can be difficult for mainstream apps. However, corporate web apps have an easier time of it because you have established channels to reach the customers or employees for which the app is designed.

For user assistance considerations,

- If you are supporting a web app, you can leverage any and all of your prior knowledge about working with web technologies. If you have already built web sites or web-based Help, you have many of the skills needed. The main difference will be in applying those skills with techniques appropriate for smartphones and tablets.

- If you are supporting a native app, you need to have a solid understanding of the platforms the apps are based on. This book references many issues specific to iOS, Android, and Windows Phone. You will also benefit from having experience with web technologies as those are often applied in native apps.

User Assistance Support for Mobile

The references and examples presented so far in this chapter are based on sales figures for large-scale mainstream consumer demand for particular types of devices. That picture changes a bit if we look at the current state of development in the user assistance community. Figure 2.3 shows the breakdown of mobile platform support as reported in the 2014 WritersUA Skills & Technologies Survey. The top vote-getter is **Web apps** (44%) – which is an open-source, cross-platform method for delivering apps to the mobile web. That is a category that is rarely reported in conjunction with the branded platforms. Nonetheless, it is a major, valid platform that claimed support from close to half of the survey respondents.

Close behind, in a dead heat for second, are **iOS** (42%) and **Android** (39%). The similar numbers probably reflect a trend for enterprise to support both platforms when rolling out apps. For corporate purposes, the economics of the two platforms are not as important as the market share. Essentially, corporations want to get their enterprise apps in the hands of their employees and customers as easily as possible.

Windows Phone trails considerably with 16%. However, that is a higher figure than what we see with consumer U.S. adoption. That likely has some correlation with the large investment IT departments already have in Windows. It will be interesting to see where that number goes as Microsoft attempts to consolidate the Windows and Windows Phone platforms.

While **Blackberry** is listed at 10% it is hard to see this continue given the current state of that platform. The **Window Store** category (6%) is the app-style format that is emerging in Windows 8 and most prominent in the Surface family of devices. **Symbian** (3%) has already been relegated to legacy status by Nokia.

Figure 2.3 Support for Mobile Platforms – WritersUA Survey

System	Response
Web apps (HTML/CSS)	44%
iOS	42%
Android	39%
Windows Phone (v.7/8)	16%
Blackberry	10%
Windows Store (Metro-style, Modern, RT)	6%
Symbian	3%

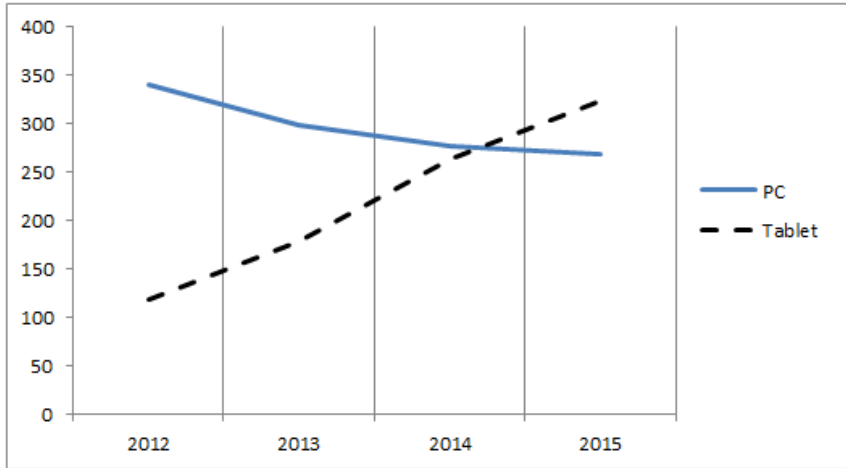
- ▶ **2014 WritersUA Skills & Technologies Survey**
<http://www.welinske.com/skills-and-technologies-survey-platforms-14/>

Tablets

The popularity of the iPad has been amazing. While tablet computers have been available for many years, it was Apple that finally captured the hearts of consumers. From the first release in the summer of 2010, the iPad was enormously popular and became a product behemoth in just over a year. The important distinction with the iPad was to marry the device size with an “always on,” speedy OS, reliance on gestures for input, and an app store ecosystem. This device category offers a simplicity and comfort that attracts customers.

The original iPad dramatically and successfully expanded the definition of mobile to include something bigger than a smartphone and smaller than a laptop. But also, different than a netbook. It could be argued that the iPad ended the sales of netbooks and has been one of the biggest drags on the PC market.

Figure 2.4 shows a comparison of recent PC and tablet sales. The numbers in the Gartner report suggest that tablet sales will exceed that of PCs sometime in late 2014. However, some news sources in early 2014 suggested that Apple’s tablet sales were flat. It could be that the current form factor has filled demand – for now.

Figure 2.4**PCs vs. Tablets Worldwide Sales (Millions of units)**▶ **Gartner report**

<http://www.gartner.com/newsroom/id/2645115>

Figure 2.5 Tablets

The biggest competitor to the iPad has been the growth of smaller tablets – mostly running Android. The seven-inch size tablet became so popular that Apple finally decided it needed to add one to its line in 2012. Currently, smaller tablets have the largest market share. Microsoft is hoping to have a piece of this success with their hybrid tablet/laptop Surface Pro. Figure 2.5 shows the variety of sizes with the iPad Air (A), Galaxy Tab 7 (B), and Surface Pro 3 (C).

If your apps will be deployed on tablets in addition to phones, you'll need to invest in some tablet hardware. Just as with the phones, it is difficult, if not impossible, to adequately design and test without having devices in hand.

Getting beyond the media hype, the popularity of a product is what drives development of apps for the product. The iPad has already achieved enough traction to make it a slam-dunk choice for most developers.

The underlying codebase of the iPad more or less ensures that most developers start with a smartphone design and then make it bigger. The iOS eco-system of tools, templates, and existing code is based on the iPhone.

On Android, the tablet OS is just another step in the evolution of the overall platform. Since version 3.0 Honeycomb, all versions of Android support both phones and tablets. The same development tools used for Android phones are being used for Android tablets. As described in the Android chapter, Android doesn't really differentiate between device types. Developers can choose from a fairly unlimited array of specs and dimensions. However, the seven-inch size tablet has definitely become one of the most popular.

The Kindle Fire from Amazon has also become a major player in the tablet market. The Fire is based on a forked version of Android. What that means is that Amazon has highly customized Android for its own use. Their own apps may have functions that you wouldn't find in other Android apps. For third-party apps, (ones not coming from Amazon) developers simply use the same techniques as for Android apps.

Microsoft's Surface has been a somewhat odd entrant in the tablet race. Running Windows 8, it is trying to bridge the gap between the more limited computing power of the iPad and more powerful laptop/desktops. So far it has had mixed results. The Surface RT failed to catch fire with customers in its first incarnation in 2013. With the exception of Microsoft's Office apps, the Surface only runs apps from the Windows Store. This makes it a non-starter for most enterprise use. The second edition of Surface still has the drawback. It makes it extremely confusing for non-techies who expect to be able to run their traditional Windows apps.

The Surface Pro has been more successful – but only relative to the Surface. Surface Pro is capable of running any application that has been developed for Windows. The Windows 8.1 reboot has been made more attractive to enterprise use by making it possible to boot into the traditional desktop. The real test of the new system though, will be

whether enterprise developers choose to make apps that run in the Windows Store interface. Previously called “metro-style”, the Windows Store apps are the ones that Microsoft expects to compete with what we currently find in iOS and Android. The creation of corporate apps for iOS and Android is documented throughout this book. It remains to be seen if IT departments decide to invest the effort to port those apps to Windows Store style.

What this means for user assistance professionals is that working with tablets is going to be more like working with smartphones than working with desktop applications. For iOS and Android, almost all of the information presented in this book regarding phones is going to be relevant to tablet design and implementation.

The Microsoft strategy is less clear. At the time of this writing, the development of Windows Phone apps is still very different from Windows 8 and Windows Store. In fact, there are currently separate “stores” for each version. Rumblings from Redmond are that Microsoft is headed toward a more unified approach.

Specific details on the design of UA for tablets is provided in Chapter 1 – *UA for Tablets*.

Using Simulators and Emulators

The major smartphone and tablet platforms provide simulators/emulators (see Figure 2.6) so developers can test project changes quickly without loading apps onto the associated devices. Simulators are a boon to user assistance professionals. They are fun to work with and allow you to experiment with designs without the help of programmers. Simulators are discussed extensively in this book. However, there are a few important limitations.

- There really isn’t a substitute for working with the actual devices. You can’t get a feel for ease of use, performance, and suitability to task through a simulator. It is also hard to understand why a person likes an iPhone over a Galaxy over a Lumia until you play with them.
- Simulators are most useful if you have access to the project code for an app. A simulator can’t display an app without the associated code. There are many scenarios where it might not be possible for you to get the code for the project you are working on. However,

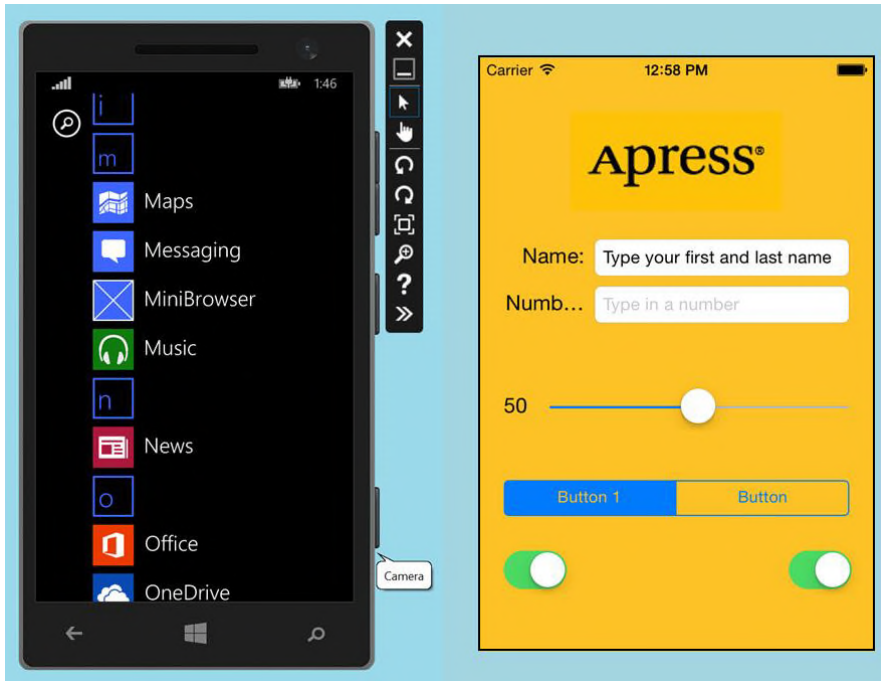
you can display browser-based Help in a simulator without project code.

- To use a simulator you need to install most, if not all, of the tools associated with a particular platform. You won't find any stand-alone simulators to use. You will need to learn the basics of the development tools in order to work with the simulators.

Figure 2.6

Windows Phone emulator

iOS simulator



The Device Acquisition Conundrum

Keeping up with phones, tablets, and apps is easier said than done. We're faced with an ever-growing list of devices and their associated operating systems. There are distinct differences on the surface of these environments and under the hood. Windows Phone has a completely different form factor from iOS, while Android is different from both of those. A person who buys an iPhone over Android is doing it for reasons beyond the cost of the device and the service contract. You can't effectively develop user assistance for a mobile app without understanding the nuances of the user interface and the expectations of the customer.

One of the challenges in keeping up with the device explosion is the cost of the devices. It isn't practical to have access to all possible devices. It might be difficult to justify having just one, let alone three or four of them. The average cost for a new smartphone is in the neighborhood of \$500-600. In the United States, this cost is often subsidized by the telecommunication companies in exchange for the customer taking on a two-year service contract. Those contracts can easily cost \$2,000-\$3,000. Add to that the obnoxious practice whereby the networks hold back on software updates in order to boost the sales of new devices and extended contracts.

Most of us only need one contract – for our primary phone. It isn't practical to have multiple contracts. Most of us also cannot afford the retail, non-subsidized price for several new phones. What is a UA professional to do?

One solution is to buy used, “unlocked” phones that are out of contract. The fast growth of the smartphone market means there is a burgeoning market for used, unlocked phones. You can find such devices on eBay for half of what they cost new. If this is a supplementary device for you, you may be able to get along using the wi-fi capability alone. Another option is to buy a pre-paid SIM card, which can usually be purchased for as little as \$25. Pay-as-you-go internet access can work just fine for UA testing. Unlocking a phone on your own can be a complicated and involved process. You are best off buying a phone that is already unlocked.

Beyond cost, there is the issue of updates. Apple has a pretty straightforward policy of pushing everyone forward at the same time on devices and software. The app development process follows Apple's release schedule very closely. The rigid control Apple has over their platform also means that developers don't have to deal with more than a couple of major variants in technology at any one time.

Version control with Windows Phone has been a mixed bag for developers and consumers. Windows Phone version 8 completely shut out the devices built for version 7. This created a lot of frustration for owners of Phone 7 devices. In order to get the benefits of the new OS it was necessary to buy a new phone.

The rate of progress in Windows Phone is an area of concern for developers. Version 8.1 was revealed to developers in April 2014. This was a full year and a half after the release of version 8.0. While this means there is only one version target for developers, it also reflects a very slow pace of improvements for the platform.

The growth of the Android platform is not limited to devices and apps. The number of versions of the operating system is somewhat amazing. From 2008 to early 2013, there have been nine major updates with “sweet” names – from Cupcake to KitKat. If you are buying an Android device, you shouldn’t consider anything with less than the Ice Cream Sandwich release. A full list of releases is shown in the Android chapter.

A final option is one that is employed by organizations with a high-stake in cross-platform applications. Many companies, like Amazon, have a department that purchases and inventories new devices as they become available. Organizations with more limited resources might consider purchasing a representative model from iOS, Android, and Windows Phone each year. Designers and developers can then check out a device to work with.

Take-away

If you are serious about mobile UA, getting one or more mobile devices is extremely important. You need to understand how mobile apps differ from the desktop. You also should gain an appreciation for the nuances of different platforms.

The cost of devices and connectivity contracts can be expensive. You’ll need to do some careful research to find the solution that best fits your situation and budget. This might include purchasing used, unlocked devices.

Best Practices

- ❖ Apple’s iOS and Google’s Android dominate market share and should be the main focus with native apps for phones and tablets.
- ❖ Consider working with platform-specific simulators and emulators for native UA design.
- ❖ It is very important to have hands-on experience with the types of devices you plan to support.