IBM PRESS

Enterprise Class Mobile Application Development

A Complete Lifecycle Approach for Producing Mobile Apps

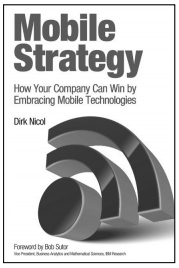
Leigh Williamson ▪ Roland Barcia ▪ Omkar Chandgadkar ▪ Ashish Mathur
Soma Ray ▪ Darrell Schrag ▪ Roger Snook ▪ Jianjun Zhang

developerWorks® series

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Related Books of Interest



Mobile Strategy

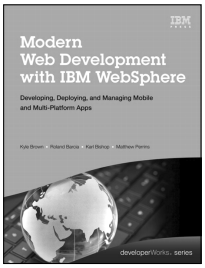
How Your Company Can Win by Embracing Mobile Technologies

By Dirk Nicol

ISBN-10: 0-13-309491-X

ISBN-13: 978-0-13-309491-6

Mobile Strategy gives IT leaders the ability to transform their business by offering all the guidance they need to navigate this complex landscape, leverage its opportunities, and protect their investments along the way. IBM's Dirk Nicol clearly explains key trends and issues across the entire mobile project lifecycle. He offers insights critical to evaluating mobile technologies, supporting BYOD, and integrating mobile, cloud, social, and big data. Throughout, you'll find proven best practices based on real-world case studies from his extensive experience with IBM's enterprise customers.



Modern Web Development with IBM WebSphere

Developing, Deploying, and Managing Mobile and Multi-Platform Apps

By Kyle Brown, Roland Barcia, Karl Bishop, Matthew Perrins

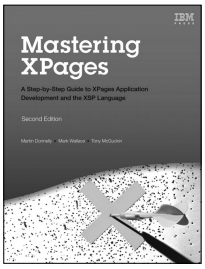
ISBN-10: 0-13-306703-3

ISBN-13: 978-0-13-306703-3

This guide presents a coherent strategy for building modern mobile/web applications that are fast, responsive, interactive, reusable, maintainable, extensible, and a pleasure to use.

Using well-crafted examples, the authors introduce best practices for MobileFirst development, helping you create apps that work superbly on mobile devices and add features on conventional browsers. Throughout, you'll learn better ways to deliver Web 2.0 apps with HTML/JavaScript front ends, RESTful Web Services, and persistent data. Proven by IBM and its customers, the approach covered in this book leads to more successful mobile/web applications—and more effective development teams.

Related Books of Interest



Mastering XPages

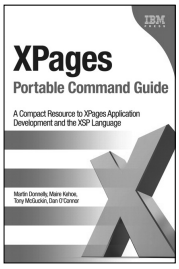
IBM's Best-Selling Guide to XPages Development—Now Updated and Expanded for Lotus Notes/Domino 9.0.1

By Martin Donnelly, Mark Wallace, Tony McGuckin

ISBN-10: 0-13-337337-1

ISBN-13: 978-0-13-337337-0

Three key members of the IBM XPages team have brought together comprehensive knowledge for delivering outstanding solutions. They have added several hundred pages of new content, including four new chapters. Drawing on their unsurpassed experience, they present new tips, samples, and best practices reflecting the platform's growing maturity. Writing for both XPages newcomers and experts, they cover the entire project lifecycle, including problem debugging, performance optimization, and application scalability.



XPages Portable Command Guide

A Practical Primer for XPages Application Development, Debugging, and Performance

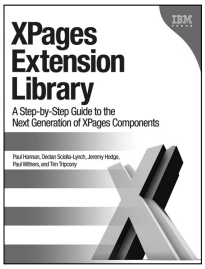
By Martin Donnelly, Maire Kehoe, Tony McGuckin, Dan O'Connor

ISBN-10: 0-13-294305-0

ISBN-13: 978-0-13-294305-5

A perfect portable XPages quick reference for every working developer. Straight from the experts at IBM®, *XPages Portable Command Guide* offers fast access to working code, tested solutions, expert tips, and example-driven best practices. Drawing on their unsurpassed experience as IBM XPages lead developers and customer consultants, the authors explore many lesser known facets of the XPages runtime, illuminating these capabilities with dozens of examples that solve specific XPages development problems. Using their easy-to-adapt code examples, you can develop XPages solutions with outstanding performance, scalability, flexibility, efficiency, reliability, and value.

Related Books of Interest



XPages Extension Library

A Step-by-Step Guide to the Next Generation of XPages Components

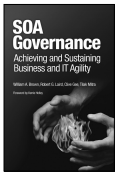
By Paul Hannan, Declan Sciolla-Lynch, Jeremy Hodge, Paul Withers, Tim Tripcony

ISBN-10: 0-13-290181-1

ISBN-13: 978-0-13-290181-9

XPages Extension Library is the first and only complete guide to Domino development with this library; it's the best manifestation yet of the underlying XPages Extensibility Framework. Complementing the popular *Mastering XPages*, it gives XPages developers complete information for taking full advantage of the new components from IBM.

Combining reference material and practical use cases, the authors offer step-by-step guidance for installing and configuring the XPages Extension Library and using its state-of-the-art applications infrastructure to quickly create rich web applications with outstanding user experiences.

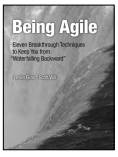


SOA Governance

Achieving and Sustaining Business and IT Agility

Brown, Laird, Gee, Mitra

ISBN: 978-0-13-714746-5

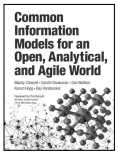


Being Agile

Eleven Breakthrough Techniques to Keep You from "Waterfalling Backward"

Ekas, Will

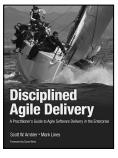
ISBN: 978-0-13-337562-6



Common Information Models for an Open, Analytical, and Agile World

Chessell, Sivakumar, Wolfson, Hogg, Harishankar

ISBN-10: 978-0-13-336615-0



Disciplined Agile Delivery

A Practitioner's Guide to Agile Software Delivery in the Enterprise

Ambler, Lines

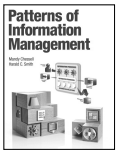
ISBN: 978-0-13-281013-5

Getting Started with Data Science

Making Sense of Data with Analytics

Haider

ISBN: 978-0-13-399102-4



Patterns of Information Management

Chessell, Smith

ISBN: 978-0-13-315550-1

Sign up for the monthly IBM Press newsletter at ibmpressbooks.com/newsletters

This page intentionally left blank

Enterprise Class Mobile Application Development

This page intentionally left blank

IBM PRESS

Enterprise Class Mobile Application Development

A Complete Lifecycle Approach for Producing Mobile Apps

Leigh Williamson Roland Barcia
Omkar Chandgadkar Ashish Mathur
Soma Ray Darrell Schrag
Roger Snook Jianjun Zhang

IBM Press Pearson plc
New York • Boston • Indianapolis • San Francisco
Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City
ibmpressbooks.com

IBM Press Program Managers: Steven M. Stansel, Ellice Uffer

Cover design: IBM Corporation

Editor-in-Chief: Dave Dusthimer

Marketing Manager: Stephane Nakib

Acquisitions Editor: Mary Beth Ray

Publicist: Brad Yale

Editorial Assistant: Vanessa Evans

Managing Editor: Kristy Hart

Designer: Alan Clements

Project Manager: Namita Gahtori, Cenveo® Publisher Services

Copy Editor: Cenveo Publisher Services

Indexer: Cenveo Publisher Services

Compositor: Cenveo Publisher Services

Proofreader: Cenveo Publisher Services

Manufacturing Buyer: Dan Uhrig

Published by Pearson plc

Publishing as IBM Press

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact international@pearsoned.com.

Library of Congress Control Number: 2015950670

ISBN-13: 978-0-13-347863-1
ISBN-10: 0-13-347863-7

Text printed in the United States on recycled paper at R.R. Donnelley in Crawfordsville, Indiana.

First printing: December 2015

I dedicate the book to Cheryl, my wife, who has always supported every project that I have undertaken.

–Leigh Williamson

I dedicate the book to my wife Blanca. I love you. Thank you for being my wife and best friend.

–Roland Barcia

I dedicate the book to my father, Devdatta Chandgadkar, for encouraging me in taking on new challenges and risks.

–Omkar Chandgadkar

I dedicate the book to the whole family who shares my excitement at being an author and is proud of me for having completed it. Thanks for all your support.

–Ashish K. Mathur

I dedicate the book to my two sisters Debbi and Pam, who are professional right-handed writers in their own right, for their "grammar lessons" in my youth that always inspired me to put my "write" foot forward.

–Roger Snook

I dedicate the book to my wife, Li Xu. She's the ultimate embodiment of faith, kindness, and patience. I also dedicate it to my kids, Grace, Daniel, and Timothy. The thought of them keeps me going when the going gets tough.

–Jianjun Zhang

Contents

This page intentionally left blank

Preface

About This Book

Every year sees dozens of new information technologies spring into general use. Many of these new types of hardware and software enjoy brief popularity in the IT community, but then fade from view, many times within a few months of their broader industry introduction. However, every decade or so, a new technology comes along with such fundamental impact that it forever changes the IT landscape. Every enterprise, across all industries, becomes motivated to adopt its use. Mobile apps represent one such transformational technology.

Over the course of 35 years in the computing industry, I've participated in four of these transformational IT waves, with mobile computing being the most recent. There are lots of patterns of adoption that are repeated during each wave.

The important adoption pattern for this book is when established enterprises begin the process of integrating the new technology with their existing systems. These companies have made significant investments in previous generations of IT and the new generation has to be knitted with the old instead of starting over "from scratch."

That is why we titled this book *"Enterprise" Mobile App Development*. It has been written specifically to cover issues and topics that arise when mobile app development meets corporate enterprise IT systems. The coauthors have decades of enterprise software-development expertise as well as extensive depth of knowledge in the mobile aspects of their chapter topic. We cover the entire lifecycle involved in enterprise mobile app development, not just one activity such as coding or testing. The chapters are designed to be useful by themselves, but they all fit into a progression that roughly follows the flow of mobile-development activities in a project.

The IBM® developerWorks® Series

This book continues the line of IBM Press publications that comprise the IBM developerWorks series.

We've seen that as new technologies such as mobile, cloud, and social computing technologies have developed, there is a need for books aimed at the enterprise IT professional level that offer practical, hands-on coverage. Thus, we intend to meet that need with topics in the revamped IBM developerWorks series—this book is the second in the series.

Just as IBM developerWorks has always provided the most up-to-the-minute information on topics of interest to developers, we want the books in this series to provide the best combination of in-depth instruction and links to new and updated material on the web so that the books will both inform our readers on the subjects that interest them and help readers follow along with exercises and examples even when the underlying technologies and products change.

So one of the key aspects of the books in this new series is that we not only provide links to information on developerWorks that are relevant to the topics in the text, but we also provide a "landing pad" about each book on developerWorks that links to constantly updated instructions for installing the tools, working through the examples, and helping developers understand what they need to do to be effective with the IBM products that the books are about.

You can find the landing page for this book at: www.ibm.com/developerworks/dwbooks/enterprise-mobile/index.html.

We hope you enjoy reading this book as much as we've enjoyed writing it.

Leigh Williamson, August 2015

How This Book Is Organized

- **Chapter 1, "Mobile: The New Generation of Information Technology,"** introduces the reasons why mobile applications are more than just a compelling new technology for enterprises. They motivate business innovation and transformation. Chapter 1 provides some examples of this, as well as an introductory discussion of some of the challenges and considerations that mobile software brings into the enterprise IT space.
- **Chapter 2, "Mobile Development Lifecycle Overview,"** provides a full discussion of the lifecycle for developing enterprise mobile apps. While there are many development tasks that are same between mobile software and other kinds of software, there are also some parts of the mobile lifecycle that are unique. Chapter 2 starts with a quick refresher about software lifecycle concepts in general and introduces DevOps concepts that are crucial for success in a modern development project. In addition to describing the idealized mobile development lifecycle, we also make a nod to pragmatism and cover techniques for migrating from a not-so-perfect lifecycle toward one more aligned with mobile-development considerations.
- **Chapter 3, "Design Quality Is Crucial, Make the Investment Up-Front,"** is an important new topic in the coverage of modern software development. It seems like an

automatic assumption that good design is essential for any mobile app. It's widely said that without design, a mobile app will fail to be an effective system of engagement. But what do we really mean by "design" in the context of mobile app development? And how can a development team apply techniques that will produce "good" design outcomes? Chapter 3 has been written by practitioners of new design thinking, specifically in the context of enterprise mobile software. You'll learn specific exercises and techniques that will benefit the entire development team and result in outstanding user-centered mobile apps that will delight the people who ultimately interact with them. Design practices are important for all kinds of products, but unfortunately, this is an aspect of enterprise software development that has frequently lacked investment. The mobile era is changing all that. Do not overlook this chapter and miss the opportunity to learn "battle-tested" techniques for putting design into your mobile project.

- **Chapter 4, "Mobile Application Development,"** covers coding and building the mobile app, which are the tasks that are at the core of the development activities. The techniques, languages, and architecture used for mobile development are rapidly changing, with important best practices for mobile software emerging at a furious pace. Chapter 4 covers a discussion about the factors to be considered when selecting an approach for coding and building your enterprise mobile app. Regardless of future evolution in mobile software and technology, these factors will remain relevant to any enterprise mobile project. We recommend rereading Chapter 4 before settling on this aspect of each new mobile undertaking. Chapter 4 also touches on the subject of Cloud software, which fits with modern mobile app architecture like hand and glove. You'll see a comprehensive architecture for mobile/cloud software systems that includes all of the considerations important for an enterprise class solution. The chapter covers mobile app deployment concerns as well, since those are a higher priority for enterprise class apps where exiting systems need to be involved.
- **Chapter 5, "Mobile Enterprise—Beyond the Mobile End Point,"** picks up from where Chapter 4 leaves the mobile backend topic and goes deeply into multiple enterprise mobile backend systems, covering how the mobile "front-end" can connect to and integrate with them. Typical protocols and application programming interfaces (APIs) are discussed. Security is covered in depth, since it is a topic with which every enterprise needs to be concerned. Management of mobile devices and software is part of the topics in this chapter too.
- **Chapter 6, "A Comprehensive Approach to Testing of Mobile Applications,"** deals exclusively with mobile software quality considerations. It doesn't take very long to think of many of the ways that mobile apps present challenges for testing and verification. We cover the range of techniques available to address these testing challenges, along with a comparison to help understand when one approach is better than the others. For mobile apps, testing is a continuous activity that goes on even after the app has

been placed into production and is in use on real end user's devices. Given the damage to enterprise reputation and monetary results possible from a "bad app," mobile quality assurance is a job that's never "done." Chapter 6 discusses how to apply the technology and products in a process that flows in a constant cycle.

- **Chapter 7, "Best Practices of Mobile DevOps,"** "closes the loop" of the development cycle with an in-depth discussion of DevOps best practices. Picking up from the initial DevOps coverage in Chapter 2, this chapter goes into more detail about putting DevOps into practice in a mobile software project. Follow the guidelines laid out in Chapter 7 and you will be able to crank up the velocity of your development team and accelerate delivery of the mobile app; something that every mobile project aspires to achieve.
- **Chapter 8, "Conclusions and Further Readings,"** provides review and reference material for further research on each of the topics.

Acknowledgments

Thank you to Kyle Brown and Steven Stansel as key advisors and supporters of this book project. Thank you both for your patience, perseverance, encouragement, and very good guidance. Also, many thanks to Mary Beth Ray who, as Executive Editor, stuck with us through the winding pathway that marked the development of this book. And, of course, so much credit is due to the fantastic group of coauthors who brought decades of experience and boundless enthusiasm to project.

–Leigh Williamson

I wish to convey that all glory and honor goes to God the Father and Lord Jesus Christ. Thank you to my wife, Blanca Barcia. My children (Alyssa, Savannah, Amadeus, and Joseph), my parents, mother-in-law, and other family members. Thank you to my spiritual brothers and sisters at Fairview Gospel Church. Thank you to all folks who contributed to this book by providing content or reviewing: John Ponzo, Greg Truty, Todd Kaplinger, Gal Schalor, Chris Mitchell, Gang Chen, Heather Kreger (and others who worked on the Mobile Reference Architecture for Mobile), and the whole IBM MobileFirst® team.

–Roland Barcia

I would like to thank my colleague Leigh Williamson, for his guidance and mentorship, and Soma Ray, for her continued support and honest feedback.

–Omkar Chandgadkar

I would like to acknowledge my mentor, guide, and coauthor Leigh Williamson for providing me an opportunity to be part of this endeavor as well as critical support in developing the material. I would like to also acknowledge the Rational® Test Workbench team for reviewing the content.

–Ashish Mathur

I would like to acknowledge my two coauthors Leigh Williamson and Omkar Chandgadkar for being mentors, inspirations, and a great support system during the creative process.

–Soma Ray

This page intentionally left blank

About the Authors

Leigh Williamson is an IBM Distinguished Engineer who has been working in the Austin, Texas lab since 1989, contributing to IBM's major software projects including OS/2, DB2®, AIX®, Java™, WebSphere® Application Server and associated family of products, the Rational brand of software offerings, the MobileFirst line of solutions, and the IBM Cloud products and services. He is currently a member of the IBM Cloud Strategy team, influencing the direction for the IBM Cloud portfolio. You can follow Leigh on twitter @leighawillia. He holds a B.S. degree in Computer Science from Nova University and an M.S. degree in Computer Engineering from University of Texas at Austin.

Roland Barcia is an IBM Distinguished Engineer and CTO for the Mobile IBM Cloud Support and Lab Services. Roland is responsible for technical thought leadership and strategy, practice technical vitality, and technical enablement. He works with many enterprise clients on mobile strategy and implementations. He is the coauthor of four books and has published more than 50 articles and papers on topics such as mobile technologies, Bluemix™, IBM MobileFirst, Java, Ajax, REST, and messaging technologies. He frequently presents at conferences and to customers on various technologies. Roland has spent the past 16 years implementing mobile, API, middleware systems on various platforms, including Sockets, CORBA, Java EE, SOA, REST, web, and mobile platforms. He has a master's degree in computer science from the New Jersey Institute of Technology.

Omkar Chandgadkar is an experience designer with a background in computer engineering and Human Computer Interaction. At IBM, he is involved in conducting strategic research for developer tools and designing for the complex challenges of enterprise customers. Through his work, Omkar strives to design experiences that solve user problems and generate business value.

Ashish Mathur is an IBM Senior Technical Staff Member and Lead Software Architect for IBM Rational functional testing tools and has the mission to build the next-generation mobile and

desktop web-testing software. He has been working on automated testing software since 1993 contributing to major IBM and Rational testing software, including Rational Test Workbench, RFT, RQM, RPT, and Rational Test manager. He has been in multiple roles in automated testing including that of a tester, consultant, subject matter expert, and a developer of the tools. He works out of the IBM India Software Labs in Bangalore, India.

Soma Ray is a UX strategist with research and design backgrounds. With educational background in Electronics, Business Administration, and Human Computer Interaction from University of Pune, India and University of Michigan, Ann Arbor, Soma has always strived to make technology more accessible and empathetic for its users. She has worked in the enterprise technology industry and currently works for the IBM Design Studio in Austin Texas.

Darrell Schrag is a 27-year software professional having spent significant time in the DoD/Aerospace and Financial Services industries. Darrell joined Rational Software in 1993 to bring software-development practices and tools to successful customers. Darrell continues to contribute to customer success with IBM after its acquisition of Rational Software. Darrell has spent time at IBM as a Rational services consultant as well as a worldwide mobile and DevOps specialist. Darrell is currently a Cloud Advisor in the IBM Cloud business unit, helping customers find their best path forward with IBM cloud solutions.

Roger Snook brings 25 years of software product innovation and consultative engagements across several industries focused on developer and project productivity to drive good business results—good design is good business! Roger is an IBM Certified Expert IT Specialist, Open Group Master IT Specialist, and an OMG Certified UML Professional in the Washington DC/West Virginia area, and holds a B.S. degree in Computer Science from Rensselaer Polytechnic Institute, Troy, New York. You can find Roger on several social networks or volunteering in his local community youth soccer or faith-based activities.

Jianjun Zhang is a Senior Technical Staff Member in IBM's Systems group, Middleware division. In the past number of years, he worked on exciting projects including helping to integrate Worklight into IBM's MobileFirst portfolio to become the foundation of the mobile strategy, leading a SaaS product development that helps business developers create mobile and web applications for departmental use without having to master coding skills. Lately, he is building cloud services to help business individuals and organizations alike to develop and manage cloud applications, Internet of Things devices, and business insights. He has a Bachelor's degree from Fudan University in China and a Master's degree from Northern Illinois University in the United States.

CHAPTER 1

Mobile: The New Generation of Information Technology

In this chapter we discuss the reasons that businesses want to develop mobile apps, and the impact that mobile apps have on how businesses operate. We also cover the challenges posed in the development of mobile apps, as an underlying motivation for the rest of the book.

Why Businesses Are Adopting Mobile Applications

Across the globe, more people are using mobile devices, which are increasingly user-friendly and intuitive, as their primary means of obtaining information and requesting services over the Internet. In addition, most enterprises realize that the users of their business applications have shifted from traditional personal computers (desktops and laptops) to mobile devices (smart phones and tablets) as a means to access web-based information. This applies whether the intended user for the application is a direct customer, employee, or business partner.

This crucial shift in end user behavior has motivated enterprises to develop mobile channels for their existing business applications, and to plan for new kinds of applications that can exploit the unique characteristics of mobile devices. While there certainly is value in producing a mobile app user interface for an existing business application, the users of mobile applications have come to expect more from their mobile experience.

Driving Business Process Innovation

Compelling and successful mobile apps create an experience that fully engages the end user. So-called "systems of interaction," these apps anticipate the desires of the user and take full advantage of the rich collection of data that the new mobile devices offer. Plus, they motivate changes in the business processes used to support the applications. Systems of interaction encompass both the engagement with the end user and also the context that existing systems-of-record enable. In short, they open up huge new avenues for innovation in business and propel new ways for businesses to interact with their various stakeholders.

The avenue for innovative user engagement is not limited to mobile phones and tablet devices. The term "omni-channel" has been coined to refer to applications that offer end user engagement across a spectrum of devices, from phones to tablets to PCs to kiosks to automobiles, and many more forms of human-technology interaction. Each of these different application end points ("channels") offers unique characteristics that enable valuable interaction with the user under circumstances best suited to that specific channel.

For example, consider the potential experience of an airline customer who checks in with the airline website two days before their flight and confirms their seat assignment and also requests an upgrade. Then less than one day before the flight, this customer uses the airline mobile app on their iPad Mini to check on their seat upgrade and "check-in" virtually for their flight (providing greater assurance to the airline that the customer will actually take the flight). Then, within a few hours of the flight, this same customer uses the airline app on their smart phone from the airport to check the gate assignment and double check the departure time for the flight (and check again the status of their upgrade request). The customer might also check if there is an airline lounge at the departure airport, and also check on their current frequent flyer mileage totals. The relationship established with this customer can continue.

During the flight, this customer may use one of their devices to buy WiFi access to the Internet. He might check the new estimated arrival time for the flight and email a colleague to confirm their plans once they arrive.

The interaction does not stop there. Once this customer arrives at the destination airport, he can use the airline system of interaction to track his luggage, check on ground transportation, and deal with navigation to a meeting. The connection with the customer can go on and on for days. He might need to confirm (or change) their return flight. He might want to confirm his frequent traveler mileage status or he might want to plan completely new trips. The customer might be using any variety of end user devices to connect to the airline IT systems and services at any time, day or night.

This is a system of interaction. It is ongoing, not a single transaction. It requires context to be effective. It is intelligent and learns from feedback from the user. It is an example of the "killer app" of the 21st century.

A Formula for Designing Engaging Systems

IBM has devised a formula for creating compelling new systems of interaction that drive business innovation. Here is the equation:

Mobile apps [smartphones, desktops, vehicles, devices, . . .] + interaction characteristics [. . .] = drives process innovation [detect, enrich, perceive, act]

The formula starts with mobile first. The mobile app is what interacts with the end user and can be deployed to more than smartphones. Modern "mobile apps" have deployment target devices as diverse as cars, TVs, desktops, body monitors, and many more.

Interaction characteristics are applied to the mobile app. We define the compelling interaction characteristics as:

- **Omni-channel**: Available across many different classes of end point device
- **Context and social aware**: To automatically tune the interaction to the place and relationships involved
- **Connected to systems of record**: Such as existing business services and data sources, as well as cloud-hosted third party services
- **Experience driven**: Able to learn and adapt to specific end user responses
- **Highly instrumented**: So that data for analysis can be obtained
- **Rapidly revised**: So that app developers can continuously improve interaction

These characteristics are applied by the system of interaction to drive business process innovation. The methodology that drives business process innovation must:

- **Detect** opportunities to engage customers and employees
- **Enrich** interaction context with historical data and trends
- **Perceive** via "in-the-now" dynamic interaction context from location, time, social media, and other events, and . . .
- **Act** on the insights gained to enable positive business outcomes.

Hence, the design characteristics for using mobile technologies to drive business innovation are simply: Detect, Enrich, Perceive, and Act.

Let us consider a quick example. We will use an app for hailing taxis to illustrate our formula for systems of interaction.

As a consumer, you download the app and set up a personal profile including preferences and payment information. When you need a cab, you press "Hail Cab" within the app. Your current location is captured, and in a few seconds you are informed that a driver named Mike is heading in your direction. This cab can be available if you confirm within the next minute. Perhaps some social factors can help influence your action here. Have your friends had a good experience using Mike for taxi service? When you confirm, you see Mike's actual location as he approaches along with an estimated time of arrival (ETA). And Mike's cab location is updated even if Mike takes a wrong turn along the way. When he arrives at your location, you get in the taxi cab and payment is established to Mike upon conclusion of the ride.

Let us see how our methodology might apply to this example scenario.

- **Detect**: Customer loads the app on their smart phone
- **Enrich**: Get customer profile, favorite cab, and favorite destination
- **Perceive**: Location of the customer, location of the closest cab, other nearby friends heading to the same destination
- **Act**: Connect the cab driver and the customer through notification and establish secure channel to pay through the app

You can see that this example app meets all of the criteria for a good system of interaction, providing all of the dimensions to deliver compelling value to the user.

Unique Challenges for Development of Engaging Applications

The creation of systems of interaction involves some unique requirements and challenges.

Most of the rest of this book provides an in-depth view of IBM's recommendations for planning, developing, testing, and deploying mobile applications. Some of those recommendations are echoed in the following paragraphs, and you can refer to the later chapters for more details.

Form Factors and User Input Technology

The first and most obvious aspect of mobile applications is that the form factor for display and user interaction is significantly different from what is used by other forms of software. A smaller form factor means that the amount of data displayed to the end user, and layout of that data, needs to be tuned to the "real estate" available on the device. Significantly less data may be displayed on some devices and therefore it must be exactly the "right" data (most relevant to what the user needs at that point in the application). This variety in form factors motivates the "responsive design" approach for application presentation, where the same application takes advantage of the display resources available on the device where it is running.

Another obvious physical difference for mobile applications is that the mechanisms for user input are different. Mobile devices have pioneered the use of nonkeyboard "gestures" (e.g., touch, swipe, and pinch) as an effective and popular method of user input. Gestures must be planned for and supported for a satisfying mobile application user experience. In addition to tactile user input, mobile devices are a natural target for voice based user input. In fact, the traditional keyboard typing form of user input is probably the least effective and least popular mechanism for input to the new systems of interaction.

Besides gathering input directly from the end user, new systems of interaction have the capability to receive input from other sources such as geo-location from the GPS component of the device and image information from the camera typically built into the device. These forms of input make mobile apps more powerful and useful than applications with a more limited array of input possibilities, and they must be considered during mobile application design and development.

Usability and User Interaction Design

There are several reasons why usability and user interaction need greater attention in the design of mobile applications. The difference in form factors and user input methods is one. It is much more difficult and time consuming to plan how to display only the data that is precisely necessary than it is to simply display all possible data and let the user visually sift through it for what they want. The mobile app designer, by contrast, has to consider the screen real estate. When an application needs to present a broader scope of data with multiple layers of detail, it is usually better to use a progressive discovery approach that allows the user to "drill down" into incrementally greater levels of detail focused on fewer specific items.

The rich variety of input methods available on mobile devices is another reason that early design work must identify and leverage more efficient ways for input data to be delivered than the simple "just type it in a form" design, which is a default for traditional web and PC applications.

Designers must avoid extensive keyboard typing for mobile apps in order to reduce end user frustration (with drastically smaller touch keyboards and lack of traditional typing feedback). Yet, identifying nonkeyboard ways in which information can be gathered and delivered to the mobile app is a significant design challenge.

In addition, there is still a more subtle reason for paying extra attention to the mobile app design effort. The way in which end users interact with mobile devices and the applications running on them is different from how they interact with stationary PCs (and even laptops). Mobile device users typically hold the device in their hand while also interacting with the immediate circumstances of their physical situation. Mobile users typically cannot concentrate on the mobile app for very long before switching attention to their physical surroundings. The interaction model for users of mobile apps is short, interrupted, and "bursty" (meaning that they need to complete the application task very quickly before switching attention).

All of these factors drive the need for applying user-centered design very early in the mobile app development project. Ideally these usability and design considerations should be codified in the requirements for the mobile application, and then linked to the later stage development deliverables, along with the tests that validate that the user interaction and "consume-ability" of the app is as satisfying as possible.

Choice of Implementation Technology

There is a spectrum of implementation choices for mobile applications on the market, and no one answer is perfect for all situations—each choice has its advantages and disadvantages. So the challenge for mobile development teams is to understand the trade-offs between the technologies, and make a choice based on the specific application requirements.

The choice of implementation technology for a mobile project impacts other decisions related to the application's development, including:

- Limiting choices for development tools
- Team roles and structure
- How the application is tested and verified
- How the app is distributed and delivered to the end user

So, the choice of implementation approach for a mobile application is crucial, and this early stage decision needs to be made very carefully.

Native Application Implementation

A "native" implementation means you are writing the application using the programming language and programmatic interfaces exposed by the mobile operating system of a specific type of device. For instance, a native implementation for an iPhone will be written using the Objective-C language (or more recently the Swift programming language) and the iOS operating system APIs that Apple supplies and supports.

Native application implementation has the advantage of offering the highest fidelity with the mobile device. Because the APIs used are at a low level and are specific to the device for

which the application is dedicated, the application can take full advantage of every feature and service exposed by that device.

However, native mobile app implementations are completely nonportable to any other mobile operating system—for example, a native Apple iOS app must be totally rewritten if it is to run on an Android device. That makes this native implementation a very costly way of producing a mobile business application.

Web Applications

Newer smart phones and tablets come with advanced web browsers preinstalled, and it is relatively easy to implement a standard web application with special style sheets to accommodate the mobile form factor and approximate the mobile device "look and feel." Mobile applications implemented using this approach support the widest variety of mobile devices, since web browser support for JavaScript and HTML5 is fairly consistent. There are several commercial and open source libraries of Web 2.0 widgets that help with this approach. In addition, the web programming model for mobile application implementation has an advantage for enterprises that already have developers trained in the languages and techniques for web application development.

The disadvantage of pure web application implementation is that such apps have no access to functions/features that run directly on the mobile device, such as the camera, contact list, and so forth.

Hybrid Mobile Application Implementation

Hybrid mobile application implementation is a compromise between pure native implementation and pure web implementation. You write the mobile apps using industry standard web programming languages and techniques such as HTML5 and JavaScript, but you package the app into a natively installable format that is distributed via the app store mechanism.

Hybrid apps are linked with additional native libraries that allow the app to have access to native device features from the single application code base. Because the bulk of a hybrid application is implemented using device-agnostic technology, most of the code for the application is portable and reusable across many different mobile operating systems. However, small segments of native code can also be integrated with the hybrid app, which means that the developer can decide how much of the app implementation shares a common code base (using the web technology) and how much is device-specific customization (written in native code).

Mobile Application Build and Delivery

Because businesses want to deliver mobile applications into the market quickly, mobile development projects typically have extremely aggressive time lines. Inception-to-delivery time frames of a few months are common. The pressure to deliver mobile apps quickly results in the adoption of agile development methods for most mobile projects.

An important element in agile development practices is continuous integration and builds. Application changes that are delivered by developers need to be processed immediately for all of the mobile operating systems on which the application is required to execute. If the mobile

application is a hybrid or native implementation, several different builds of the application need to be triggered each time a change set for the application is delivered by a developer. The build setup and configuration for each supported mobile environment will be different from the others, and it is most likely that a small "farm" of build servers will need to be provisioned and available to handle these builds of the mobile application for multiple operating systems.

Testing

Testing poses another huge challenge for mobile application development, because it represents a step-jump in complexity and cost over more traditional applications. Unlike traditional PC and Web applications, the range of potentially supported mobile devices and release levels is staggering. Test matrices for mobile projects commonly contain hundreds and even thousands of permutations of device, mobile OS level, network carrier, locale, and device orientation combinations.

There are more variables for mobile testing that are not relevant for other kinds of software. The same device model may function in a subtly different way when connected to a different carrier network. And the quality of the network connection can have profound impact on the behavior of a mobile application. Even the movement of the mobile device itself may be an important factor in the behavior of the application (some applications specifically exploit device movement).

The majority of mobile apps are based on multi-tier architecture, with the code running on the device itself serving as the "frontend" client to data, and the services supplied by more traditional middle-tier and data center representing the "backend." Effective and comprehensive testing of mobile apps requires that all tiers of the application be addressed, not only the code on the mobile device. The setup and availability of test versions of the middle tier and backend services can present very large cost and complexity challenges for the testing of mobile applications.

Many mobile projects start by using manual testing approaches, which is the quickest way to begin testing. But you have to buy all of the various mobile devices that you plan to support with the app, and pay someone (likely a team of people) to tediously go through a written script of instructions describing the tests on each of those devices for every build of the application. While manual testing serves an important purpose in providing crucial usability feedback for the app, it is extremely expensive and inefficient.

As an alternative, there are mobile app testing solutions that rely on running an agent program on the device for interaction within an automated execution. This approach has the flexibility of using either real physical devices or emulators for testing, with the added efficiency of automation. However, the test team bears the costs of setting up the devices to be tested and installing the test agent on them.

Enterprise Mobile Development

The brief coverage in the previous section of a few of the challenges faced when developing mobile apps should begin to raise your awareness that there are plenty of new and different concerns to address for mobile, compared to more established kinds of enterprise software. All by

itself, the development of code that is to be installed and run on mobile devices poses substantial obstacles. But when that mobile app needs to be a part of a mission critical enterprise IT system, or at least integrate with existing enterprise data and resources, that is an order of magnitude greater level of complexity and headaches.

This book has been written specifically to cover issues and topics that arise when mobile app development meets corporate enterprise IT systems. The coauthors have decades of enterprise software development expertise as well as extensive depth of knowledge in the mobile aspects of their chapter topic. We cover the entire lifecycle involved in enterprise mobile app development, not just one activity such as coding or testing. The chapters are designed to be useful by themselves, but they all fit into a progression that roughly follows the flow of mobile development activities in a project.

Summary

Our aim in writing the book is to combine the concerns that we have learned over years of corporate software development with the fresh and rapid aspects of new mobile app production. Since the mobile software domain is still fast moving and evolving, we have worked hard to emphasize techniques and considerations in this book that will remain constant regardless of the latest technology of the day.

One of the guaranteed elements in the software industry is change. We learn to deal with it by identifying the concepts and concerns that endure at a level above the detailed technical aspects. Mobile apps represent technological disruption in a very big way, especially for enterprise software and those of us devoted to producing it. The authors hope that you finish this book with a new list of enduring ideas for how to make your enterprise mobile app successful and delightful!

Index

Note: Page numbers followed by ‘f’ and ‘t’ denote figures and tables, respectively.

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X