Leigh Williamson, John Ponzo,
Patrick Bohrer, Ricardo Olivieri,
Karl Weinmeister, Samuel Kallner

Swift@IBM

IBM Press

Swift™ in the Cloud

WILEY

Swift™ in the Cloud

Swift™ in the Cloud

Leigh Williamson

John Ponzo

Patrick Bohrer

Ricardo Olivieri

Karl Weinmeister

Samuel Kallner

WILEY

Swift™ in the Cloud

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-31937-5
ISBN: 978-1-119-36853-3 (ebk)
ISBN: 978-1-119-36847-2 (ebk)

Manufactured in the United States of America
10 9 8 7 6 5 4 3 2 1

Library of Congress Control Number: 2017946220

Karl would like to dedicate this book to his supportive wife, Samantha, and their crazy kiddos, who still gave him enough time to write.

Leigh dedicates this book to his wife, Cheryl, always his compass through life, and his daughter, Claire, the light of his life.

About the Authors

Leigh Williamson is an IBM Distinguished Engineer who has been working in the company's Austin, Texas lab since 1989, contributing to major software projects for IBM, including OS/2, DB2, AIX, Java, WebSphere, Rational, and MobileFirst products. He is currently leading a cross-disciplinary team of IBM Cloud Consultants who assist clients with cloud computing strategy and execution. Leigh is an active mentor and leader in the IBM technical community, working with international mentees; working in the Advanced Technical Eminence program; leading multiple patent brainstorming teams; publishing technical blogs and articles; conducting external broadcast webinars; and speaking at IBM and non-IBM conferences.

This is the fourth IBM Press book on which Leigh has collaborated over the past 15 years, with works covering Java standards, WebSphere Application Server, and enterprise-class mobile application development. He holds a B.S. in Computer Science from Nova University and an M.S. in Computer Engineering from The University of Texas at Austin.

You can follow Leigh on Twitter at @leighawillia. His LinkedIn address is `https://www.linkedin.com/in/leigh-williamson-9048654`.

John Ponzo is an IBM Fellow and Chief Technology Officer for Mobile who has shaped the future of IBM business in mobile computing and delivered innovative products and services to web browser and server-based standards. He is a pioneer in technology that promotes end-user interaction through mobility, web programming, web application middleware, and software tools. John led the development of key software technologies including HTML/JavaScript runtime libraries, XML middleware, web services runtime libraries, Eclipse Web Integration, and Web 2.0 enterprise collaboration services. He also authored the "Enterprise Mobile" and the "Mobile First" theses that are the cornerstone of mobile strategy and execution at IBM. John also serves as the IBM technology ambassador to Kenya.

John is the primary technical collaborator between Apple and IBM in the effort to define and enhance the Swift programming language for both mobile client development and cloud services development.

Patrick Bohrer is a Distinguished Engineer in the IBM Cloud division. His responsibilities include serving as the technical lead for the company's global efforts around Swift@IBM (`developer.ibm.com/swift`). He formerly served as the technical lead of the Mobile Innovation Lab in Austin, Texas. Patrick also helped lead IBM Research's Mobile Research agenda after co-leading the 2012 Global Technology Outlook topic entitled "Mobile First," which helped set the technical direction for current mobile and cloud efforts at IBM. Patrick received a B.A in Computer Science from The University of Texas at Austin.

Karl Weinmeister is the Program Director for Swift@IBM Engineering, based in Austin, Texas. He is passionate about improving people's lives and experiences with technology. In his role, he has helped to enable Swift to extend from its mobile roots to become a full-stack language ecosystem.

Previously, Karl led engineering for the IBM Mobile Innovation Lab. He is a diehard Duke basketball fan and enjoys spending time with his family.

Ricardo Olivieri is a Senior Software Engineer at Swift@IBM Engineering. His areas of expertise include gathering and analyzing business requirements, architecting, designing, and developing software applications. Ricardo has extensive experience in the complete software development cycle and related processes, especially in Agile methodologies.

Ricardo has several years of experience in Java development as well as in Groovy, Perl, and Python development, with a strong background in back-end (server-side, business logic, SQL, and NoSQL databases) and front-end development. Several years ago, Ricardo added Business Process Manager (BPM) design and development to his skill set, which allowed him to assume the role of BPM consultant and developer using IBM Business Process Manager. While working at the Mobile Innovation Lab, Ricardo gained valuable skills and knowledge in the iOS and Android ecosystems.

Ricardo is now mainly focused on the adoption of the Swift language on the server and the IBM cloud, Bluemix. He has a B.S. in Computer Engineering from the University of Puerto Rico, Mayagüez Campus.

Samuel (Shmuel) Kallner is a Senior Technical Staff Member in the Smart Client Platforms group at the IBM Research Lab in Haifa, Israel. He is currently the Technical Lead of the Kitura project. Shmuel has over thirty years of experience at IBM working on a wide variety of projects including mobile apps, web-based end-user application development environments, mobile app developer tools, both sides of client-server–based applications, and more.

About the Technical Editor

Matthew Perrins is a Senior Technical Staff Member working on IBM Cloud Developer Services in Austin, Texas. Matt is one of the architects for the company's production-ready public Cloud Developer Experience on Bluemix—the IBM cloud platform. He is focused on making it very easy to develop and deploy mobile, web, and digital channel applications with the IBM cloud, and to integrate them with world-class leading runtimes for Swift, Node.js, and Java. Matt has been leading the IBM cloud teams in the evolution to true DevOps continuous delivery with cloud-native architectures, and driving that integration into the IBM Bluemix user experience.

Matt has spent a significant part of his career building systems-of-record and systems-of-engagement solutions using IBM technologies with IBM clients. He understands developers, user experiences, transactions, and cognitive solutions and how to deliver them at scale on the IBM cloud.

Credits

Executive Editor
Jim Minatel

Project Editor
Adaobi Obi Tulton

Technical Editor
Matthew Perrins

Production Editor
Dassi Zeidel

Copy Editor
Marylouise Wiack

Production Manager
Katie Wisor

Manager of Content Development and Assembly
Mary Beth Wakefield

Marketing Manager
Christie Hilbrich

Professional Technology & Strategy Director
Barry Pruett

Business Manager
Amy Knies

Project Coordinator, Cover
Brent Savage

Proofreader
Nancy Carrasco

Indexer
Johnna VanHoose Dinse

Cover Designer
Wiley

Cover Image
© fishbones/Getty Images

Acknowledgments

Many thanks to everyone at Wiley Publishing for their outstanding work on this project: to Jim Minatel for encouraging us to take this book concept forward and for yet again supporting the realization of a book that engages in deeper learning; to Adaobi Obi Tulton, the project editor, for driving this project to completion in the most efficient way possible—it was a real pleasure to work with such an accomplished and adept editor; to Marylouise Wiack, the copy editor, for translating this book into readable prose; and to Dassi Zeidel, the production editor, for bringing everything together to create a final, polished product.

Sincere thanks go to Matthew Perrins, the technical editor, for the incredible amount of work and personal time he selflessly put into ensuring that the content in this book can be utilized seamlessly by readers. Also, thanks to Steven Stansel, who was instrumental in formulating the original concept and proposal for the book. Brian White Eagle provided invaluable assistance with peering over the horizon in Chapter 9. And Shereen Ghobrial contributed the bulk of the mainframe-related content.

The biggest thank-you must, of course, go to our own families. This book was written over six months, predominantly at night and over weekends and holidays. In addition to sharing us with extremely demanding full-time jobs, our families made further sacrifices to enable us to spend time on this project.

Contents at a Glance

Contents

Introduction

Since Apple introduced the Swift programming language in 2014, it has become one of the most rapidly adopted computer programming languages in history. Programmers love the modern syntax used by Swift and the way it's fun to develop code, similar to how they felt about Java a generation ago. Programming skills and experience in Swift are in high demand in the industry, with the promise of high salaries for those who invest the time to learn and practice the language.

Apple originally introduced Swift as an alternative language to Objective-C for developing iPhone, iPad, and macOS applications. The company has now expanded Swift into the realm of solutions for the Internet of Things, with support for tvOS (Apple TV) and watchOS (Apple Watch wearable devices). As illustrated in Figure 1, Swift is now one of the most popular open source projects and Swift frameworks such as Kitura are gaining ground quickly.

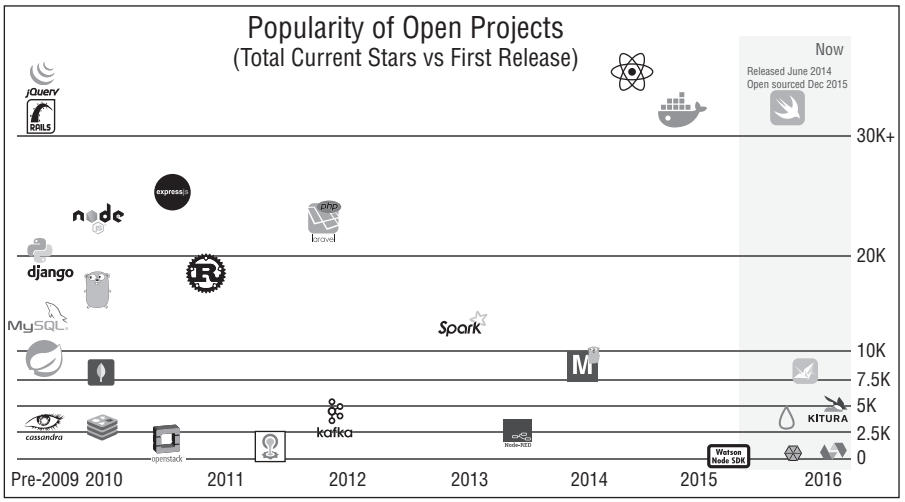

Figure 1: Popularity of the Swift programming language

At about the same time that Swift was first introduced, Apple and IBM formed a strategic partnership to produce innovative, industry-specific mobile applications for the Apple iOS ecosystem. IBM proceeded to embrace the Swift programming language and implemented over 100 mobile apps as part of this partnership. IBM engineers saw the value of Swift firsthand in these applications. The open source release of the Swift language in late 2014 began another chapter in the partnership, as IBM chose to invest in and support the cause of bringing the Swift language to the cloud as a result of the focus on server-side Swift environments.

Development of these applications highlights the critical, expanding role of server-side logic in powering these new experiences that users now take for granted. From syncing data across devices; to connecting people with friends and co-workers; to monitoring news and alerting us about new events based on our interests and activities; to providing cognitive insights into our applications; server-side logic is critical to creating truly brilliant apps. Now the ability to develop, debug, and deploy this logic in the same language used to create mobile experiences is a game changer for the development community.

This book, written by members of the development team at IBM who helped bring Swift to the cloud, covers everything you need to know about how to develop Swift programs that run in cloud environments. It combines technical information with the concepts that originally led to the development of the technology. This book provides plenty of examples of Swift language code, as well as a living website where a community consisting of the authors and other passionate Swift experts continues to discuss Swift and its future directions.

IBM, Apple, and Swift

On July 15, 2014, IBM and Apple announced what was at the time a very surprising partnership, with the goal of transforming business applications by building mobile enterprise and industry-specific solutions for the Apple platform. This partnership was unexpected by most industry watchers and radically altered the enterprise mobile computing landscape.

The IBM offering that resulted from the Apple partnership is called MobileFirst for iOS. It focuses on enterprise and industry transformation by providing users with the latest features of the Apple platform and user design, coupled with the back-end data center integration required to reinvent the next generation of enterprise applications.

Key features of the IBM offering include large-enterprise–class robustness and scalability; back-end enterprise data center integration; big data and analytics integration; and a highly polished mobile front-end user interface experience.

The user interface experience was codesigned by Apple and IBM to significantly upgrade the standards for usability, elegance, and user satisfaction beyond typical enterprise software. Since 2012, IBM has been making massive investments in IBM Design Thinking philosophy and techniques, building up several large design studios in an effort to apply good design to business software. This software design emphasis by IBM was one of the natural collaboration areas of the partnership, with the strong Apple design culture being applied to all of their own products.

As IBM MobileFirst for iOS was being developed, a choice was made to use the latest iOS platform APIs for iPhone and iPad business applications. The scope was later extended to include Apple Watch. IBM leveraged many of the extended development kits provided by Apple, such as HomeKit, CloudKit, and the connected car capabilities in various new and innovative business solutions. As of December 2015, IBM had created over 100 industry-specific mobile apps for the IBM MobileFirst for iOS collection.

While Apple has pioneered the transformation of the consumer mobile app experience, IBM and Apple consider business mobile app transformation to be an underserved market and a really great design opportunity that they can uniquely address together.

Introduction of Swift

Apple introduced the Swift language at the Apple World Wide Developers Conference (WWDC) in 2014, and IBM decided to begin using the language to build the first wave of IBM MobileFirst for iOS mobile applications. IBM assembled a team of developers with expertise in building mobile solutions. Their previous programming language skills included Java, JavaScript, and Objective-C. This team of IBM programmers quickly learned Swift and began using it to implement the mobile apps in the IBM MobileFirst for iOS collection.

Working with Apple, we at IBM learned a lot about what it takes to build amazing mobile business applications. IBM also discovered the value of Swift.

Swift was designed by Apple to be a safe, interactive, and high-performance systems language. It also blends the ease of scripting language syntax with the performance of a systems language. The IBM team found that in comparison to mobile apps developed in Java for the Android mobile platform and Objective-C for iOS, Swift apps required less code.

The IBM team appreciated everything about Swift, from its type safety—which empowered them to be agile and evolve the applications quickly while knowing that the compiler would catch any errors—to its performance and memory advantages, which are critical for application responsiveness. The concise syntax of the language also led to great developer productivity.

The IBM teams also learned to understand what is necessary to develop application-specific web services to power these business mobile applications. It significantly enhanced the overall productivity of the team to not have to switch languages away from what was used for the mobile front end (Swift) to work on the back-end services.

The developers found the Swift code easier to read, share, and evolve. What the Swift language did for legibility of the application code represented a large increase in productivity for the development team. Other languages used for mobile apps were generally more verbose. It also significantly increased code quality with Swift-based type checking. The importance of a strongly typed language in the productivity of the development team can hardly be overstated. Most of the more than 100 applications developed by IBM for various business solutions could be produced by a handful of programmers working in small teams.

Figure 2 shows a comparison of Swift with other programming languages and illustrates how Swift enables inherent application performance and developer productivity benefits through its attributes such as:

- Modern programming language constructs
- Error detection at compile time, not runtime
- Code reengineering