

1. Record Nr.	UNINA9910484528303321
Autore	Tsadok Avi
Titolo	Mastering Swift package manager : build and test modular apps using Xcode / / Avi Tsadok
Pubbl/distr/stampa	[Place of publication not identified] : , : Apress, , [2021] ©2021
ISBN	1-4842-7049-5
Descrizione fisica	1 online resource (214 pages)
Disciplina	005.133
Soggetti	Swift (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Table of Contents -- About the Author -- About the Technical Reviewer -- Chapter 1: Introduction -- Personal View -- The Future by Apple -- About This Book -- Chapter 2: Organized Mess -- Meet "Weathery" -- The Weathery Project -- Code Organization -- Separation of Concerns -- Presentation, Business, and Services -- It All Starts with Data Flow -- The Three-Layer Architecture -- Problems to Mini Problems -- Stability -- Teamwork -- Build Time -- Redesign Our App Architecture -- Take a Paper and Draw a UML -- Do This in Collaboration -- Relations Between Components -- What Side to Start? -- Opened and Closed Layers -- Define Design Patterns -- Define Your Entities -- Define Your Interfaces -- Summary -- Chapter 3: Swift Package to Rescue -- In the Meeting Room -- Let's Start with Terminology -- Module -- Library -- Framework -- Swift Package -- Code Location -- Dependencies -- Platform and System Version -- Versioning -- Create Our First Package -- Create a Package from the Command Line -- Xcode 11 and Swift Packages -- Package. swift -- Swift Tools and PackageDescription -- Products and Targets -- Naming -- What Are We Naming? -- Be Consistent -- Be Meaningful -- Be Unique -- Platforms -- Dependencies -- Working on Our Package -- Building -- Building from Xcode -- Creating a New Package from Xcode -- Summary -- Chapter 4: Design and Link Our First Package -- What Do We Do Next? -- Linking the Package Libraries -- Think As a Service Provider -- Design a Good API -- Think About

the Developer As a Naive One -- Access Control -- Private -- fileprivate -- Open/Public -- Parameters, Return Type, and Properties in Public Access Level -- Public Constructor -- Access Level in Testing -- Scenario-Driven Design -- Three Different Ways Produce Three Different Experiences -- Inout Parameter -- Naming -- Clarity -- Consistency -- Class/Protocol Names. Method Names -- Comments -- Rendered Documentation -- The Basics -- Discussion and Summary -- Parameters, Returns, and Throws -- Code Blocks -- More Special Fields -- Document Classes, Structs, and Enum -- Summary -- Chapter 5: All About Dependencies -- A Utils Package -- Code Coupling -- Why Code Coupling Matters? -- Types of Coupling -- Structural Dependencies -- Inheritance -- Aggregation -- Usage -- Fan-Out Similarity -- Fan-Out Similarity with Inheritance -- Logical Coupling -- Dependencies Between Packages -- A Version-Based Requirement -- Semantic Versioning Standard -- Control Our Dependencies' Update -- Branch-/Commit-Based Requirement -- A Local Dependency -- Project's Remote Dependencies -- Some Best Practices About Dependencies -- Reducing Coupling Level -- Dependency Injection -- Using Closures -- The Responsibility Is on the App -- Summary -- Chapter 6: And Sharing for All -- Sharing Options -- Sharing Using a Public Git Repo -- There Is No Support for Multipackage Repositories -- Versioning -- Remember Semantic Versioning Standard? -- Testing -- Remember, Public Is Public -- Distribute Your Package -- Readme.md File -- Readme.md File Location and Formatting -- Readme.md Markdown Formatting -- Readme.md Best Practices -- Adding Pictures and Animations -- Private Git Repository -- What Is the Problem Exactly? -- Personal Access Token -- Generating New Token in Git -- The Problem with Personal Access Token -- Using SSH -- What Is SSH Connection? -- Generating SSH Keys -- Editing SSH Config File -- Adding the Keys to the Agent -- Adding the Public Key to GitHub -- Going Back to the Package.swift File -- Other Machines -- The Monorepo Solution -- What Is a Monorepo? -- Dependencies' Management Can Be Simpler -- Cross-Project Changes Are Easier -- How to Move Forward with Only One Project? -- Monorepo Also Has Disadvantages. Solutions to Monorepo Issues -- Disassemble Your Monorepo -- Summary -- Chapter 7: Testing Is Caring -- Swift Packages and Testing -- The Importance of Testing -- The Package.swift File -- Our First XCTestCase -- @importable Import -- allTests Variable -- Adding More Test Methods -- Test Methods Start with "test" -- Arrange Act Assert -- Running Tests -- Straight from the Code Editor -- Running Tests from the Test Navigator -- Adding/Removing Tests from Our Scheme -- Running from Terminal -- Integration Tests -- Why Are Integration Tests Important? -- Setup Integration Tests with Swift Packages -- Link a Package to the Test Target -- Summary -- Chapter 8: Spicing Up with Resources -- Why Do We Need That? -- Swift Tools 5.3 -- So How Do I Add Resources? -- Bundles -- Does a Swift Package Have a Bundle? -- Access the Package Bundle from Code -- Access Package Resources from the Main App and Vice Versa -- Explicitly Declare Resources -- Adding a Whole Folder of Resources -- Process vs. Copy -- Exclude Resources -- Localization -- "defaultLocalization" Property -- "lproj" Folders -- Process -- Explicitly Declare Files Outside lproj Folders -- Provides Localization Texts with Dependency Injection -- Summary -- Chapter 9: XCFrameworks -- Swift Package Downsides -- Meet XCFrameworks -- XCFramework Creation -- Start a New Project -- Archive the Framework -- The xcodebuild Command -- The Output -- Create XCFramework -- The Swift Package Connection -- Creating a Binary Target -- Local Path -- Using URL -- Summary --

