



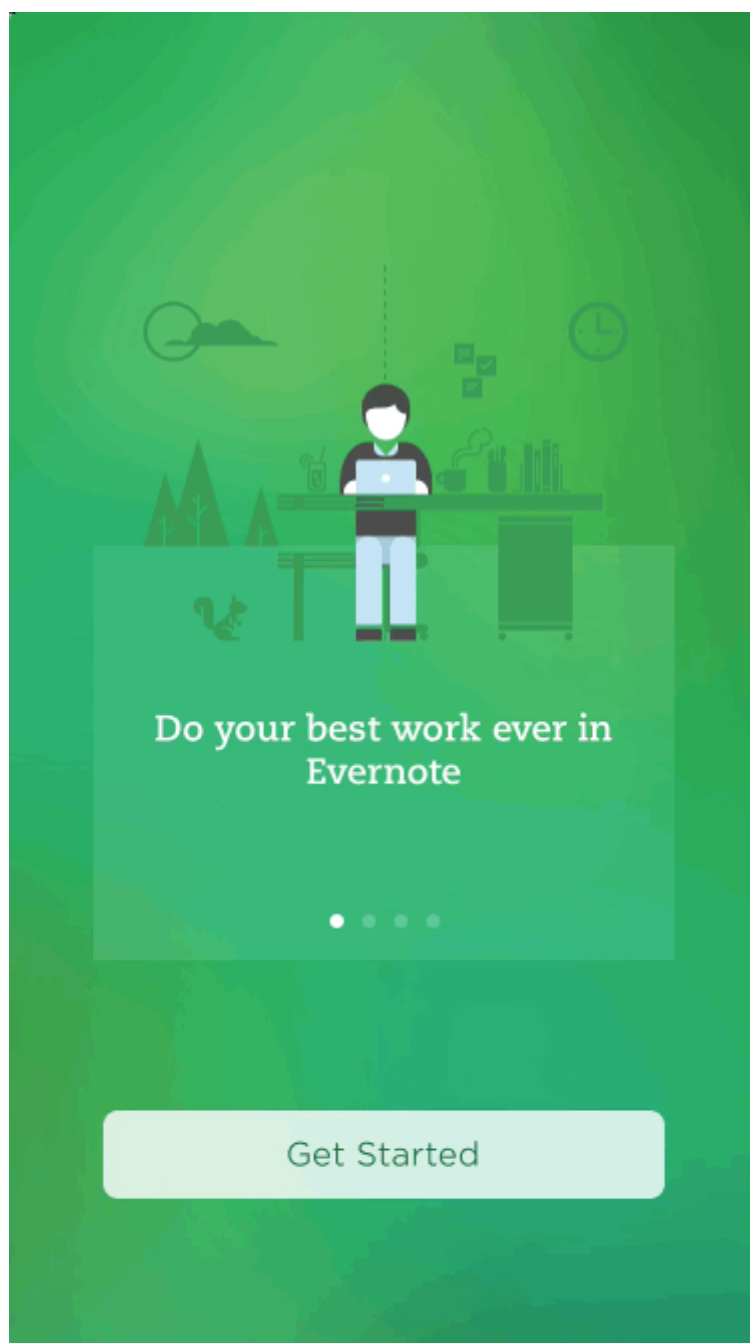
Nikant Vohra

[Follow](#)iOS Engineer at PayPal, Creator of Swift Programming Course <https://www.udemy.com/swift-de...>

May 29, 2015 · 6 min read

Things I wish I had known before starting iOS development—Part 1

My designer handed me a nifty looking onboarding flow to complete just days before the launching of our app. The flow resembled the Evernote iOS app.



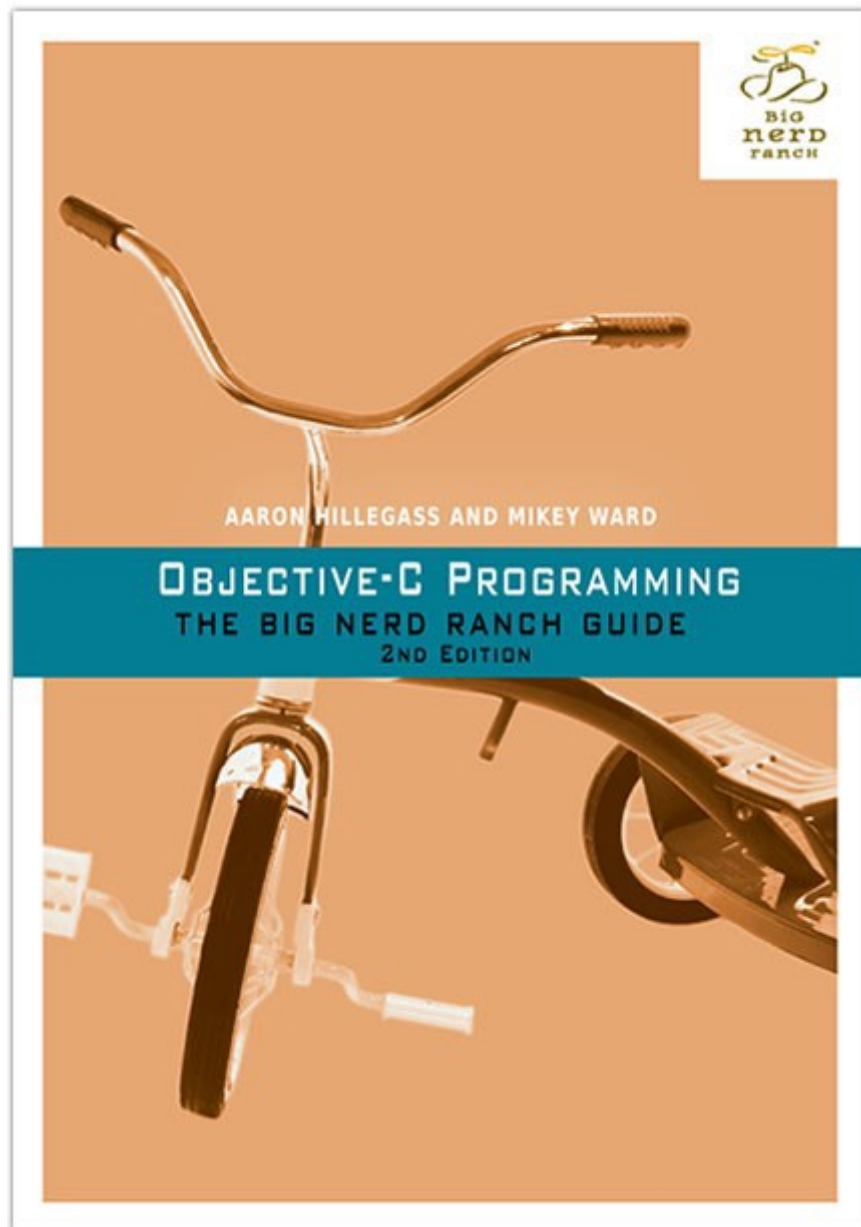
I jumped straight into coding the flow on Xcode using page view controllers and scroll view. I was able to complete the flow in two days taking help from Stack Overflow and Google. When I showed the flow to a friend who is also an iOS developer, he told me that I could have completed this in an hour if I would have used this [open source project](#).

This and many other experiences have certainly turned me into a better iOS engineer over the past one year. I would like to share my experience so that others do not have to repeat the same mistakes on their path to turn into an iOS ninja.

Focus on the fundamentals

When I started with iOS development I straight away took this free [awesome course](#) offered by Stanford University. Although I learned a lot from this course, it did not focus much on teaching me the fundamentals of the language used for iOS development which was primarily Objective-C at that time. As I started writing code for my own apps, I realised that I have missed a lot of fundamentals of the language which resulted in buggy code.

If you do not have enough programming experience in an Object Oriented Language, I recommend you to read a good book on the language before diving into iOS App development. My personal favorites are [Big Nerd Ranch Guide](#) for Objective-C and [The Apple's Guide](#) for Swift.



Sure you can learn a lot by reading the tutorials on [Ray Wenderlich](#) or watching great video lectures on [Team Treehouse](#) but if your fundamentals are flawed it will come to bite you back in the future.


Github is your best friend

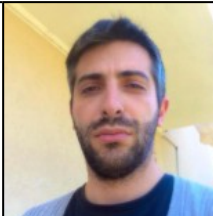
I absolutely love the iOS open source community. They have shown their awesomeness with tonnes of great projects like AFNetworking, Restkit, JSQMessages etc. You must learn to take advantage of the work that this community has done in the past.

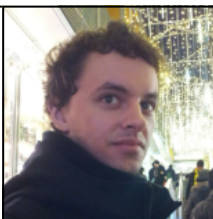
Before coming up with your own library or solution for a problem search on Github or Google for similar solutions. It is highly likely that a fellow developer has already open sourced a project that might fit your needs.

Feel free to reach out to the community via [Facebook Groups](#) or [Slack chat](#). They will be more than willing to answer your questions. Skim through the code of some good open source projects to learn how the experienced developers organize their code and start following similar design patterns for your own code.

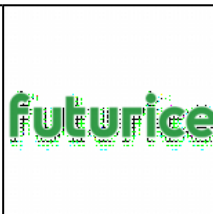
Some of the most amazing iOS resources on Github can be found [here](#).

<p>vsouza/awesome-ios</p> <p>awesome-ios—A curated list of awesome iOS ecosystem, including Objective-C and Swift Projects</p> <p>github.com</p>	
---	---

<p>matteocrippa/awesome-swift</p> <p>awesome-swift—A collaborative list of awesome swift resources. Feel free to contribute!</p> <p>github.com</p>	
---	--

<p>cjwirth/awesome-ios-ui</p> <p>awesome-ios-ui—A curated list of awesome iOS UI/UX libraries</p> <p>github.com</p>	
--	---

If you are looking for the iOS best practices to follow in your own projects look at the following.

<p>futurice/ios-good-practices</p> <p>ios-good-practices—Good ideas for iOS development, by Futurice developers.</p> <p>github.com</p>	
---	---

Know your tools

Most iOS developers use Xcode as the primary tool for development. Xcode has a lot of powerful features like Storyboards, Auto Layout which if learned can increase your productivity to great extents. A lot

of developers avoid Storyboards due to some limitations, but I personally regard them as a great way to quickly layout your screens.

Learn navigating through XCode using shortcut keys. Though it might seem like a little time saved in the present, it consolidates into a lot of time saved in the future. These are some of the solutions I use consistently that have helped me to speed up my development process.

1. Use Cocoapods for dependency management. It will make the work of your team much easier.
2. Learn how to setup continuous integration early in your project so that you do not have to repeat redundant steps.
3. Use Testflight for beta distribution of your apps. With Apple buying Testflight, it has become very easy for anyone with a iTunes Connect account to distribute beta builds using Testflight.
4. Integrate Crashlytics in your app to get crash reports as they happen.
5. If you do not want to setup your own backend server go for the amazing service offered by Parse.

Read some great blogs and newsletters

I already told you about the awesomeness of the iOS open source community. A lot of great blogs have been started by some experience iOS developers which host great content every week. Some of my favorites are:

Cocoa with Love In this post, I'll look at best practice for using and managing text strings in your user interface. This is... cocoawithlove.com	
---	--

Arguably the best iOS Blog by Matt Gallagher. Matt's way of doing things is nothing short of masterful.

iOS Dev Weekly Subscribe to a hand picked round up of the best iOS	
--	---

development links every week. Curated by Dave...
iosdevweekly.com



Technically not a blog but insanely awesome updates every week by
Dave Verwer

NSHipster

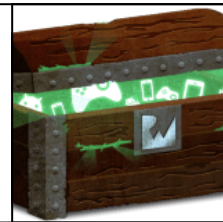
Playgrounds aren't a feature of the Swift language
per se -instead, they are a terrific showcase for all...
nshipster.com



NSHipster is a journal of the overlooked bits in Objective-C and
Cocoa. Updated weekly by Mattt Thompson

Ray Wenderlich

High quality programming tutorials: iOS, Android,
Mac, and more!
www.raywenderlich.com



Ray Wenderlich's Blog (Really good for beginners)

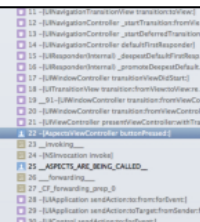
Custom Controls for iOS and OS X—Cocoa Controls

Edit description
cocoacontrols.com



Peter Steinberger

With Xcode 6, Apple added support for the
NS_DESIGNATED_INITIALIZER flag, and also added...
petersteinberger.com



Matt Gemmell

This is a brief status update regarding my writing
projects. It might be of interest to those who are...



mattgimmell.com**Natasha The Robot**

Just another WordPress site

natashatherobot.com

The insights you get by reading these blogs will certainly help you to turn into a better iOS developer.

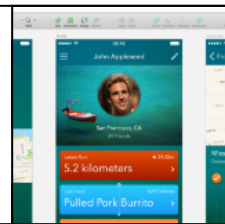
Design too can be easy

A lot of us developers tend to run away from the design aspects of iOS. We treat design to be out of our grasp and leave it all to designers. But with a little effort you can learn to design your own apps.

Nowadays the line between the designers and developers is blurring as a lot of indie iOS developers tend to design, develop and market their apps with success. I will talk about the marketing aspects in the next part but if you want to design your own iOS apps, learn the Sketch tool. It is super easy to learn and made specifically for designing apps and websites.

Bohemian Coding—Sketch 3

Sketch gives you the power, flexibility and speed you always wanted in a lightweight and easy-to-use...

bohemiancoding.com

You can find tonnes of Sketch resources and plugins online that will make your work even more fun and easier. Once you have your designs ready you can tie them together using this amazing tool.

**Free mobile & web prototyping for designers
—Marvel**

Turn sketches and designs into interactive web, iPhone, iPad, Android and Apple Watch prototypes...

marvelapp.com

In the next part, I will talk about the approach you must take while building your own apps as well as the marketing tricks for an iOS app.

. . .

**Want to learn more about iOS and Swift
enroll in this Swift Development Course
available on discount for a limited period.**

If you liked this post recommend it so that others can enjoy it as well.

