Why should businesses use Swift 5 for iOS app development?

PATRICK R | JULY 14, 2020

Partner Post - Intuz Consulting Beyond Technology

Swift is a powerful programming language that is easy to learn and implement by developers. The language is available for macOS, watchOS, tvOS and iOS and more such. The code that Swift has is very easy to learn, implement, and interactive as the syntax is concise and expressive. It contains modern features that developers find interesting and love to use them in fun manner. The code that Swift offers is safe by its design and developed software that runs lightning-fast. Finding the right iOS app development services and process will benefit your business in getting the expected result in the end.

What's new in Swift 5 and 5.1 update

Apple has released the [brand new Xcode 11](#) that features strong support for the new iOS 14. Apple has release the new version of IDE that contains support for the new version of its developing language i.e. Swift 5.1.

The new version Swift 5.1 has come up by integrating new language features and enhanced module stability. It enables the binary framework creation for sharing I with others benefiting the languages that has added support for getting module stability.

If we talk about module stability then it defines a text-based module interface file that describes the API of any binary framework and allows it to get compiled with essence of the code utilizing diverse versions of the compiler.

Here are some of the brands that are using Swift for iOS mobile application development

- Slack

- Uber

- Lyft

- Robinhood

- Delivery Hero

- ViaVarejo

- Bepro Company

- Ruangguru and more such…

The major language updates of Swift 5.1

Swift 5 and 5.1 Major Language Updates

- Opaque return types
- Implicit returns
- Ordered collection diffing
- Creation of uninitialised arrays
- Matched optional enums against non-optionals
- Warnings for ambiguous none cases
- Static and class subscripts
- Universal self

Binary compatibility and a stable ABI

ABI was stable for the new Swift 5 app development need and this is the reason for which Swift libraries incorporated iOS, macOS, watchOS, and tvOS that were easier a difficult element for developing and having very less size and hence were not included in any of the libraries.

Standard library updates

- It offers support to handle and update diff on the collection of appropriate types

- Enhanced flexibility and initializing an array

- Extra APIs for making it simple for working with Strings that covers the creation and management of contiguous strings and help to work with Unicode text and generic initializers for String.Index and Range

- Variation of incremental API enhancements to work with SIMD types that covers support to extend vectors and reductions and vector swizzles

- Identifying protocol to support entities that will need unique identifiers

Additional compiler

Swift development allows easy access to the memory for debugging and the need of he releasing builds. It provides huge supports for dynamically callable types that assists in enhancing the interoperability with dynamic languages such as Python, JavaScript and Ruby. It helps in implementing the language protocols and handling future enum cases. Also, it introduces user-defined dynamic "callable" types and literal initialization using coercion. It provides full support for "less than" operator in the compilation conditions and helps in identifying key path.

Package manager updates

The Swift Package Manager consists of series of new brand features that are included in the Swift 5 application development version and covers dependency mirroring, targeted-specific build settings, customized deployment targets, and hold full capability for generating the code average data. In addition, the Swift opts for running the command that covers an ability to import libraries in the REPL with building an executable. Swift 5 implements the Package Manager proposals from the whole Swift evolution process that includes dependency mirroring, target specific build settings and platform deployment settings.

Module Stability

Swift 5.1 helps in creating binary frameworks that can shared with others that are benefiting language's added support for the module stability. It will make it superiorly easy and speedy for developers to create iOS-based applications. Swift 5.1 defines a new brand text based module interface file that will describe the binary framework API and will enable it to get compiled with the codes and with the full assistance of diverse compiler versions.

Language Server Protocol

Swift 5.1 OSS toolchain packages for Ubuntu and macOS covers binaries for SourceKit-LSP, implementation of the LSP (Language Server Protocol) for Swift and C-baed languages.

SwiftSyntax Updates

It has been re-architectured having a strong focus on improving its performance by using the parser from the Swift compiler. The performance of the syntax tree visitation and operations that are related to this has been improved by re-architecting the whole internal data structures for making them more efficient.

Benefits of using Swift 5 for iOS app development
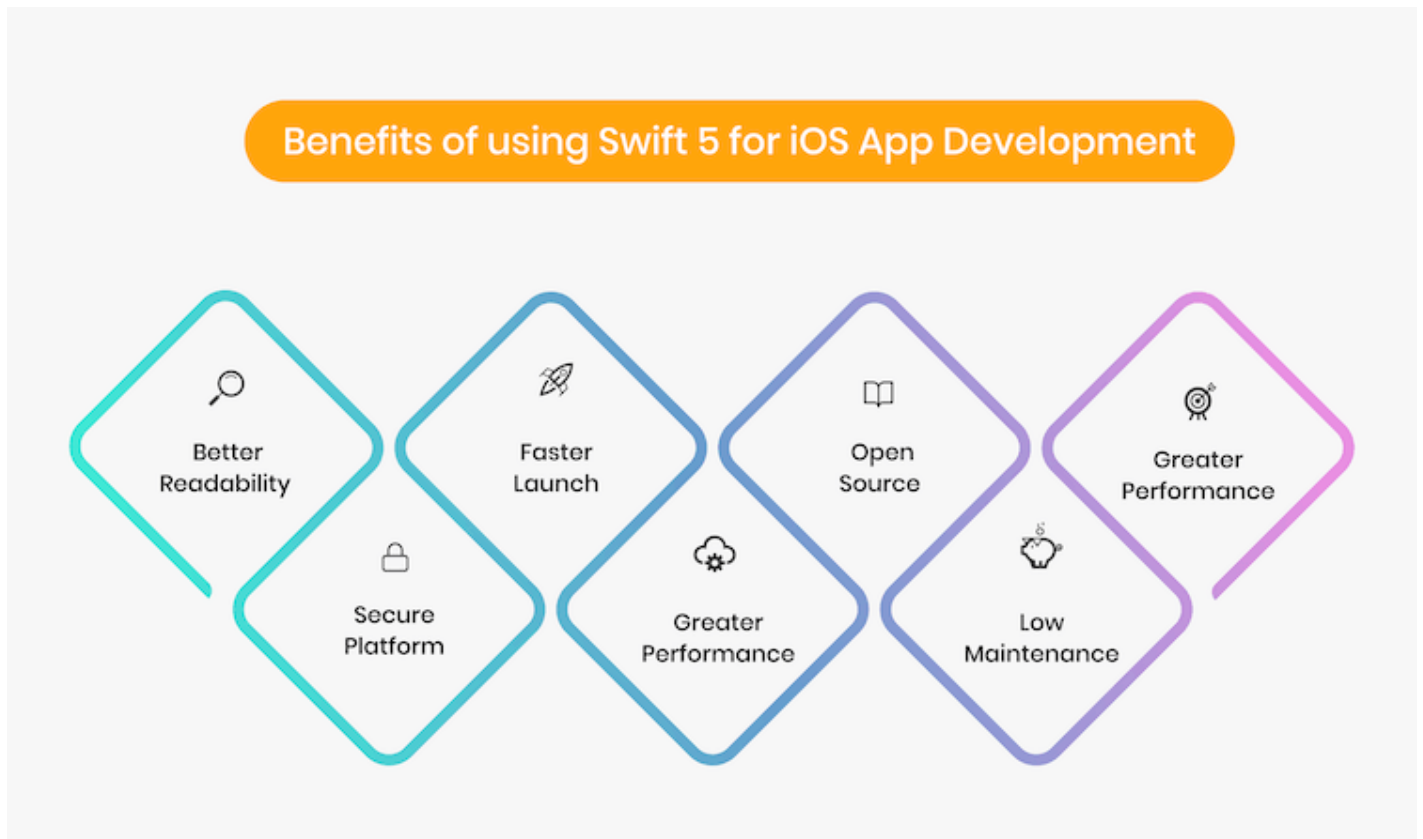
Smaller App Size

Swift 5 makes the language compatible with binary that means the end result of Swift mobile development will benefit the smaller for users for running their new operating system i.e. macOS, iOS 12.2, watchOS 5.2 and tvOS 12.2.

Faster Launch

The another benefit of Swift 5 is that it has faster launch time as every dynamic library is utilized by an app that will be cached in the memory and will be shared among the apps.

Greater Performance

Users don't want to work in a broken app and with an app freeze that glitches and this is the major reason behind the app uninstallation rate and it is important for a business to avoid it.

The API stability of Swift 5 assists apps to behave and offer improved performance.

Better Readability

One of the most important reason of choosing Swift 5 is that it because of its readability. It is simple to modify, write and read and it has a clean syntax that makes it unique. It requires less number of code lines s compared to Objective-C in comparison to iOS apps. Startups and enterprises picks Swift for the need of iOS mobile application development and lesser down the cost of developing app and the overall development time.

Low Maintenance

As soon as the app is published, it requires an ongoing maintenance and that to time to time. For an instance, the app will reach your users and you will need to add new content and track the performance for integrating new features into it for behaving the whole functionality.

Open Source

Swift is an open-source technology and it is announced in 2015. It opens new ways for employing swift over diverse platforms. Open source means more and more improvements can be made in the design and structure of it. The future mobile apps can be created with enhanced functionality and improved performance.

Secure platform

Swift is a secure mobile app development platform that everybody needs in today's fastest-growing world of mobile apps. It has a dialect and structured development that avoids the errors that are possible to come in Objective-C.

How the Swift 5.1 update will affect iOS app development process?
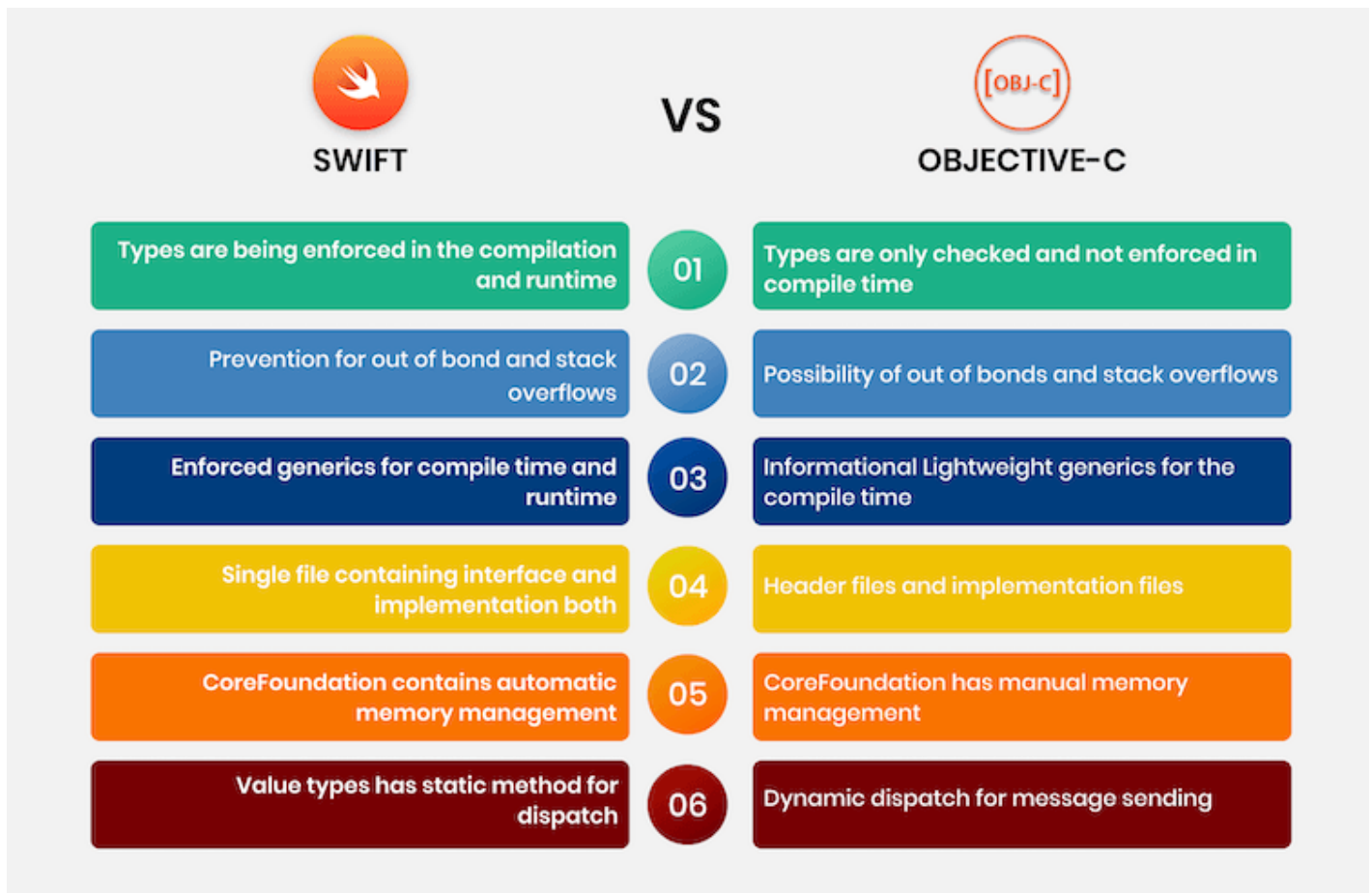
Talking about iOS app creator then Swift is considered as the most powerful programming language. In 2018, Swift was ranked 14th for being the most popular programming language.

Swift 5.1 allows to build a share binary frameworks. Its features allows to create a better design and APIs that has good protocol extension and makes the code writing easy. The app development process has become easier for developers to perform from the launch of Swift

5.1. Developers who have immense of knowledge in Swift and in creating iOS apps can handle the whole app building process easily and with full security.

Swift faster than Objective-C - let's have a look at this -

Swift and Objective-C both of these programming languages are utilized for developing iOS apps. Both of these manages are based on the infrastructure of LLVM compiler and there is only a single iOS SDK for both of them. This is the reason that there is not much difference between how these languages work with the Cocoa frameworks.

SWIFT		OBJECTIVE-C
Types are being enforced in the compilation and runtime	01	Types are only checked and not enforced in compile time
Prevention for out of bond and stack overflows	02	Possibility of out of bonds and stack overflows
Enforced generics for compile time and runtime	03	Informational Lightweight generics for the compile time
Single file containing interface and implementation both	04	Header files and implementation files
CoreFoundation contains automatic memory management	05	CoreFoundation has manual memory management
Value types has static method for dispatch	06	Dynamic dispatch for message sending

There is no doubt in saying that Swift is faster than Objective-C as it removes all the limitations of C language and has been improved using advanced technologies. Swift was purposely designed to operate faster and create beautiful and feature-loaded iOS mobile applications.

What will be next in Swift 5.1?

Going forward, Swift will get more technological changes in its programming language. There is no doubt in saying that Swift has many potential improvements to be made in its area and

that will make the whole writing and running Swift code more enjoyable and can continue to unlock the new use cases for the Swift programming language.

It has been said that a new diagnostic engine has been already developed for Swift 5.2 that will aim to improve the type checker and all types of warning and error messages that are being generated by the compiler of Swift when any expression failed to get compiled. The new features of Swift 5.1 will help to create more attractive and feature-rich apps and for that reason, you will be needing the best iOS app development services.

Find what we've shared with you here quite insightful? Interested to learn more about the Swift best practices, tips, and tricks? *Have a look at Intuz's [ultimate guide](#) on iOS app development.*