Гукемухова Б.М., Чомаев Ш.Р.

Проблемы внедрения нового фреймворка SwiftUI для разработки приложений на устройства Apple

(Россия, Черкесск)

doi: 10.18411/lj-11-2020-07

idsp: ljournal-11-2020-07

Аннотация

Актуальность данной темы состоит в том, что сейчас перед разработчиками стоит выбор инструмента проектирования интерфейса программных продуктов. Цель исследования состоит в сравнении инструментов SwiftUI и UIKit. Проанализировав все возможности SwiftUI и сравнив его с UIKit мы пришли к выводу, что разработчики не готовы использовать данный инструмент в крупных проектах. Но в то же время, SwiftUI является наилучшим инструментом для разработки pet - проектов.

Ключевые слова: SwiftUI, UIKit, декларативный синтаксис, WWDC 2019

Новый фреймворк SwiftUI был представлен на конференции разработчиков WWDC 2019, он написан на языке Swift и предназначен для декларативного описания UI в коде.

SwiftUI – это инновационный, исключительно простой способ создания пользовательских интерфейсов на всех платформах Apple с помощью языка Swift. Появилась возможность создавать пользовательские интерфейсы для любого устройства Apple, используя только один набор инструментов и API.

Декларативный синтаксис Swift легко читаем и прост в написании, SwiftUI стабильно работает с новыми инструментами проектирования Xcode 11, что также позволяет синхронизировать код и дизайн приложений. Автоматическая поддержка динамического типа, темного режима, локализации и доступности означает, что проекты написанные на SwiftUI, уже имеют самый передовой метод отображения пользовательского интерфейса.

SwiftUI использует декларативный синтаксис, поэтому существует возможность легко указать, что должен делать пользовательский интерфейс приложения. Например, есть возможность написать, что нужен список элементов (Рисунок 1.1), состоящих из текстовых полей, а затем описать выравнивание, шрифт и цвет для каждого поля. Таким образом, код становится проще и легче для чтения, чем когда-либо прежде.

Этот декларативный стиль применим даже к таким сложным понятиям, как анимация. Присутствует возможность добавить анимацию практически к любому элементу, с помощью лишь нескольких строк кода. Во время выполнения система обрабатывает все шаги, необходимые для создания плавного движения, и даже обрабатывает прерывания для поддержания стабильности пользовательского интерфейса приложения.

```
1 import SwiftUI
2
3 struct Content: View {
4
5     @State var model = Themes.listModel
6
7     var body: some View {
8         List(model.items, action: model.selectItem){ item in
9             Image(item.image)
10            VStack(alignment: .leading) {
11                Text(item.title)
12                Text(item.subTitle)
13                .color(.gray)
14            }
15        }
16    }
17 }
```

Рисунок 1.1 – Пример описания таблицы элементов

Несмотря на положительные моменты в SwiftUI существуют проблемы, которые можно избежать при использовании UIKit:

- Ограниченное покрытие API;
- Ограниченное принятие;
- Ограниченная поддержка.

Ограниченное покрытие API. Одним из недостатков SwiftUI является то, что в настоящее время он не имеет такого же широкого охвата API, как UIKit.

Например, если показывать элементы в сетке, то в UIKit использовалось UICollectionView, но SwiftUI не имеет эквивалента. Если позволить пользователю вводить многострочный текст, который в UIKit используется как UITextView, в SwiftUI также не имеет аналога.

Это не значит, что Apple не проявляет заботу о разработках, а делает это преднамеренно, чтобы единовременно выпускать “обертки” для всех своих API-интерфейсов, а затем вносить изменения позже. Вместо этого используют гораздо более осторожный подход и постепенно добавляют API-интерфейсы. Это должно сократить количество серьезных изменений, которые предстоят в будущем, так как это дает инженерам Apple больше времени для оттачивания подмножества API, что они намереваются предоставить.

В большинстве случаев можно найти обходные пути, но это утомительно, когда какая-то конкретная вещь тривиальна в UIKit, но трудна или невозможна в SwiftUI.

Ограниченное принятие. SwiftUI был анонсирован только на WWDC 2019 и доступен на устройствах iOS 13 или более поздней версии. Это сразу означает, что:

- Почти каждое приложение, написанное на сегодняшний день, использует UIKit;
- Любое приложение, которое должно поддерживать iOS n-1 или n-2 (например, iOS 12 и iOS 11), даже не может начать переключаться на SwiftUI в течение года или более.

Для тех, кто намерен получить работу в качестве разработчика iOS, необходим опыт работы с UIKit, так как является обязательным, потому что это то, что используют существующие кодовые базы. Фактически можно полагать, что UIKit все еще будет доминирующей платформой UI через четыре года.

Множество людей пытаются провести параллели между принятием Swift и SwiftUI, что бесполезно. Принятие Swift было быстрым, потому что оно работало на всех платформах, поддерживаемых Apple (UIKit, SpriteKit и т. д.), а также уже поддерживало iOS n-1, поэтому многие компании могли сразу перейти на него.

В то же время SwiftUI является идеальным кандидатом для личных приложений, приложений для pet-проектов, прототипов приложений или для общих экспериментов.

Ограниченная поддержка. UIKit существует уже более десяти лет, а это означает, что:

- почти каждая проблема, с которой можно было бы столкнуться, вероятно, уже была решена;
- существует множество библиотек, предоставляющие расширения и настройки.

Хотя некоторые учащиеся могут представить, что старшие разработчики держат в голове огромное количество UIKit, простая истина заключается в том, что все разработчики используют Google, StackOverflow, HackingWithSwift и многие другие ресурсы, чтобы найти решения проблем. Если разработчик столкнулся с проблемой, он может в любое время опубликовать ошибку на web-сайте, но независимо от того, какой ответ будет получен, это сэкономит много времени на их поиск в Интернете.

Существование SwiftUI каким-то образом не сделало UIKit устаревшим: так как в течении ближайших лет знание UIKit будет являться строгим требованием для разработчиков, которые хотят получить работу по разработке под iOS.

1. Структура и интерпретация компьютерных программ / Харольд Абельсон, Джеральд Сас-сман - Добросвет, КДУ, 2018. - 608 с.
2. Foundation Functions Reference – Apple Developer [Электронный ресурс] / Apple Inc. – Режим доступа: https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Miscellaneous/Foundation_Functions/. – Загл. с экрана. – яз. англ.
3. Core Data Programming Guide: Persistent Store Types and Behaviors [Электронный ресурс] / Apple Inc. – Режим доступа: <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CoreData/>. – Загл. с экрана. – яз. англ.
4. GitHub – GitHub [Электронный ресурс] / GitHub Inc. – Режим доступа: <https://github.com/>. – Загл. с экрана. – яз. англ.
5. Swift Pocket Reference: Programming for iOS and OS X / Anthony Gray, 2014. – 871 с.

Муравейко А.Ю., Лазарев И.В.
Принципы SMM в социальных сетях

*Липецкий государственный технический университет
 (Россия, Липецк)*

doi: 10.18411/lj-11-2020-08

idsp: ljjournal-11-2020-08

Аннотация

Массовое распространение социальных сетей привело к тому, что практически весь рынок рекламы перешёл в интернет. Люди, не имеющие средств к заключению договоров с большими рекламными агентствами, пробуют развиваться в этой сфере самостоятельно. Маркетинг в социальных сетях (Social Media Marketing) подходит для этого как никогда лучше. Работа на целевую аудиторию, продвижение собственного бренда, ведение собственного instagram аккаунта одновременно, разве это не то, о чём мечтали все маникюрные салоны вашего города?

Ключевые слова: social, media, marketing, социальные сети, instagram, vk, вконтакте, продвижение, целевая аудитория.

Abstract

The massive spread of social networks led to the fact that almost the entire advertising market moved to the Internet. People who do not have the means to conclude contracts with large advertising agencies are trying to develop in this area on their own. Social Media Marketing is better for this than ever. Working for the target audience, promoting your own brand, maintaining your own instagram account at the same time, isn't that what all the nail salons in your city have dreamed of?

Keywords: social, media, marketing, social networks, instagram, vk, vkontakte, promotion, target audience.

Так что же такое SMM и с чего следует начать? Сама суть маркетинга в социальных сетях (Social Media Marketing) [1] заключается в привлечении клиентов (целевой аудитории) из социальных сетей. Современной аудитории и пользователям гораздо удобнее следить за брендом или компанией через социальные сети (по статистике от Marketing Sherpa) и в связи с этим мире начал появляться большой спрос на SMM специалистов, ведь практически каждая компания ведёт свой профиль в социальной сети. Для большинства организаций это единственный канал продвижения в интернете.