

УДК 004.9

ФРЕЙМВОРКИ И ПОДХОДЫ ДЛЯ РАЗРАБОТКИ КРОССПЛАТФОРМЕННЫХ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С ТОЧКИ ЗРЕНИЯ БИЗНЕСА

Воронина М.С., студент гр. ПИб-192, III курс

Научный руководитель: Тайлакова А.А., старший преподаватель
Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

Почти у каждого человека есть мобильное устройство. При наличии такого гаджета появляется необходимость в удобных мобильных приложениях, которые с легкостью позволяют получать нужную информацию всего за несколько кликов. В то же время у многих разработчиков рано или поздно появляется желание написать свое первое мобильное приложение для широкого использования. Но учить новые языки программирования, их особенности и семантику просто нет времени, рынок и бизнес нуждается в скором решении поставленной задачи. Тогда на помощь приходят фреймворки для мобильного сегмента разработки. В данной статье рассмотрим фреймворки, использующие гибридно-нативную разработку и гибридную веб-разработку.

Современные тенденции заставляют человека использовать все новые устройства и технологии для получения полного объема информации. Но, согласитесь, не у каждого есть ноутбук или персональный компьютер. Да и в вопросах удобства выигрывает мобильный телефон. Тогда появляется необходимость в разработке программного обеспечения для смартфонов, но выбор языка программирования и подхода всегда ставит разработчика перед выбором. Открываются плюсы и минусы каждой стороны, поэтому приходится сравнивать различные подходы и выбирать из них единственный. Но как же сделать правильный выбор? Никто не устоит перед возможностью потратить меньше времени и ресурсов, получив хороший результат. Для этого рассмотрим сегмент кроссплатформенной мобильной разработки, сравнивая различные фреймворки.

Кроссплатформенный подход подразумевает под собой написание единого кода, который в дальнейшем будет работать на разных операционных системах. Однако все еще широко распространены нативная разработка, но сил и времени на нее уходит гораздо больше.

Однако будем рассматривать ситуацию не только со стороны удобства клиента, но и со стороны бизнеса. Очевидно, что писать одно и то же приложение для разных устройств является дорогим удовольствием. Поэтому существуют гибридно-нативная разработка и гибридная веб-разработка. Первый вид включает в себя нативный пользовательский интерфейс и общий код, а также общую кодовую базу и нативный код. Второй вид осуществляется че-

рез веб-интерфейс с общими компонентами, единой кодовой базой. В целом гибридная разработка востребована среди крупных мировых компаний. Многие известные приложения являются кроссплатформенными (The New York Times, eBay, Aliexpress) [1].

Вернемся к теме бизнеса. После проведенных опросов среди различных компаний было выявлено частичное непонимание необходимости мобильной разработки для отдельно взятых компаний. Ответ заказчика по вопросу необходимости чаще всего содержал в себе повышение лояльности и имиджа компании, а также желание уникальности готового продукта на общем рынке. Это не совсем верно. Во-первых, стоит понимать цель создания приложения и дальнейшие прогнозы его востребованности. Иногда вложения больших денег не приносят никакой эффективности для бизнеса. Прежде всего стоит выявить направление и примерную аудиторию. После передать требования и пожелания в сторону специалистов-аналитиков, которые построят аналитический план развития и потребности продукта. При положительных коэффициентах стоит передать запрос в дизайнерские организации. Там нужно подобрать дизайн не только исходя из текущих тенденций, но и сопоставить контент сайта с UI/UX частями проектного менеджмента. В вопросе купли-продажи важно ориентироваться на общемировые стандарты. Согласитесь, не стоит изобретать Word совсем не похожий на Word, так как клиент ожидает получить похожий продукт с более приятным дизайном и дополнительными фишками. И только тогда есть смысл сотрудничества с компанией разработчиков. Выгоднее всего прибегнуть к аутсорсингу, чтобы в короткое время найти хороших специалистов. Из дополнительных вещей можно выделить подходы к ведению разработки. Одним из интересных наборов практик является DevOps. Сюда входит системное администрирование и прямая разработка приложения с организацией внутрерабочего процесса. Внедрение методик DevOps позволяет автоматизировать и оптимизировать процессы с помощью технологий, но все начинается с культуры внутри организации людей, которые в ней работают [2].

Теперь рассмотрим некоторые фреймворки, выявим их плюсы и минусы, которые помогут и бизнесу, и разработчику.

Facebook предоставляет инструмент React Native. Используется JavaScript и библиотека React. Если ваша специализация frontend, изучение пройдет незаметно. Согласно официальным данным React Native применяют технологические гиганты, например, Instagram, Facebook, Tesla, UberEast и другие. Главным принципом обозначили «научись один раз, используй везде». Конкретно для разработчиков придумали много удобств. Можно выделить прямое добавление нового кода и правок во время выполнения, что полезно при настройке пользовательского интерфейса. Из большинства собратьев этот фреймворк отличается высокой производительностью.

Google предоставляет фреймворк Flutter. Код пишется единой базой для мобильных устройств, веба и десктопа. В основе лежит язык Dart, что является некоторой проблемой, так как придется потратить время на изучение чего-

то нового. Если без JavaScript трудно представить программирование, особенно фронтенд-разработчику, то Dart все же известен далеко не каждому. Из хороших новостей – язык напоминает C# или Java, поэтому backend-разработчикам этот фреймворк придется по душе. Из популярных приложений стоит выделить Google Ads, Greentea и Hamilton Musical. Из значительных минусов отмечается отсутствие автоматической адаптации для разных стилей отдельных элементов. Тут придется покопаться, либо не выбирать это решение в случае набора разнотипных компонентов. Как и у React Native имеется функция прямого добавления кода без повторной сборки, а также высокие показатели производительности.

Теперь перейдем к относительно новому участку программирования кроссплатформенных приложений, к Ionic. Это фреймворк хочется назвать чудом света, ведь с ним очень просто работать. Нужны знания верстки, т.е. HTML, CSS, препроцессоры, JS (TS). Если вы знаете Angular, React или Vue, то можно наложить их друг на друга. Результат совмещения - быстрая разработка, простая организация хранения структурного кода, легкая передача данных между компонентами. У фреймворков, описанных ранее, предлагалась концепция единого кода. Ionic не отстает, наоборот, даже выводит разработку на новый уровень! Все компоненты автоматически адаптируются к платформе использования, а значит, разработка становится еще быстрее. Но, к сожалению, производительность является слабой стороной, поскольку для визуализации используются веб-технологии, а также отсутствуют нативные компоненты. Так что придется выбирать - или быстрая разработка с замедленной работой, или нативные фреймворки с трудностями в разработке, но хорошими показателями на выходе [3].

Что касается личного опыта использования, можно сказать, что фреймворки на основе HTML, CSS, препроцессоров, JS (TS) являются оптимальным решением, так как изучать много нового не приходится. Основные компоненты, вложенные и структуры будут заранее известны, исходя из опыта простой разработки. Дальше остается дело за малым - взвесить все «за» и «против».

Помните, что каждый инструмент уникален, обладая своими плюсами и минусами в сторону исполнения конкретной задачи. Если вы не имеете опыта или уверенности, стоит обсуждать проект с высококвалифицированными разработчиками, в ходе рассуждения выбирая лучший вариант.

Список литературы:

1. Лучшие фреймворки для разработки кроссплатформенных мобильных приложений [Электронный ресурс] // URL: <https://blog.sibirix.ru/crossplatform-frameworks/> (дата обращения 20.03.2022).
2. Что такое DevOps [Электронный ресурс] // URL: <https://azure.microsoft.com/ru-ru/overview/what-is-devops/#devops-overview> (дата обращения 23.03.2022).

3. Сравнение фреймворков для кроссплатформенной мобильной разработки: React Native, Flutter, Ionic, Xamarin и PhoneGap [Электронный ресурс]
// URL: <https://tproger.ru/translations/cross-platform-frameworks-for-mobile-development/> (дата обращения 22.03.2022).