



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.4'2

АНАЛИЗ СУЩЕСТВУЮЩИХ ИНСТРУМЕНТОВ ДЛЯ КРОСПЛАТФОРМЕННОЙ РАЗРАБОТКИ ПРИЛОЖЕНИЙ

¹Магомедов О. Р., ²Чернышёв С. А.

^{1,2}Санкт-Петербургский государственный экономический университет, Россия (191023, г. Санкт-Петербург, ул. Садовая, 21), e-mail: ¹smart7even@yandex.ru, ²chernyshev.s.a@bk.ru.

²Санкт-Петербургский государственный университет промышленных технологий и дизайна, Россия (191186, г. Санкт-Петербург, ул. Большая Морская, 18)

В статье исследована область электронной коммерции как наиболее популярной и большой сферы, для которой разрабатываются приложения. Обоснована актуальность использования кроссплатформенных технологий, рассмотрены решения, позволяющие разрабатывать кроссплатформенные приложения.

Ключевые слова: электронная коммерция, кроссплатформенные технологии, кроссплатформенные фреймворки, разработка приложений.

ANALYSIS OF EXISTING TOOLS FOR CROSS-PLATFORM APPLICATION DEVELOPMENT

¹Magomedov O. R., ^{1,2}Chernyshev S.A.

¹Saint Petersburg state university of economics, Russian Federation, (191023, Saint-Petersburg, Sadovaya str. 21), e-mail: ¹smart7even@yandex.ru, ²chernyshev.s.a@bk.ru.

²Saint Petersburg State University of Industrial Technologies and Design, Russian Federation (191186, Saint-Petersburg, Bolshaya Morskaya str. 18), e-mail: chernyshev.s.a@bk.ru.

The article investigates the field of e-commerce as the most popular and large area for which applications are developed. The relevance of using cross-platform technologies is substantiated, and solutions for developing cross-platform applications are considered.

Keywords: e-commerce, cross-platform technologies, cross-platform frameworks, application development.

Введение

Почти любой бизнес в 2022 году стремится развиваться в ИТ: создавать свои онлайн-магазины, создавать ПО для обеспечения работы предприятия, налаживания более эффективной коммуникации между сотрудниками, вести аккаунты в социальных сетях и так далее. Сфера ИТ развивается благодаря бизнесу, а сами информационные технологии позволяют бизнесу получать множество новых покупателей и увеличивать прибыль. Большинство приложений разрабатывается для E-commerce и для обеспечения E-commerce процессов. Поэтому рассмотрим E-commerce и особенности разработки, ориентированной на

Е-commerce, а затем перейдем к анализу кроссплатформенных фреймворков, которые набирают популярность в разработке программных продуктов.

Электронная коммерция

Е-commerce (электронная коммерция) - это сфера экономики, которая включает в себя все финансовые и торговые транзакции, осуществляемые при помощи компьютерных сетей, и бизнес-процессы, связанные с проведением таких транзакций [1]. Е-commerce занимает большую часть рынка – по информации РБК на электронную торговлю в 2021 году будет приходиться 18,1% от всех розничных продаж в мире, а объем продаж составит почти 5 трлн. долларов [2]. Самыми яркими примерами и лидерами сферы являются такие компании как Alibaba, Amazon, eBay и ряд других. На вышеперечисленных площадках продаются и покупаются физические товары, но также есть сервисы, которые продают цифровые товары (Spotify и Apple music предлагают подписку на музыку, Netflix предлагает подписку на сериалы и кино), и сервисы по предоставлению услуг (Например, Booking.com предоставляет услугу по бронированию мест в отелях). Стоит отметить, что пандемия COVID-19 поспособствовала росту сферы Е-commerce. При этом замечается тенденция, что в странах, у которых и ранее были развиты интернет-продажи, рост оказался незначительным относительно роста стран, в которых Е-commerce находится на стадии становления (особенно Индия и Россия) [3].



Источник данных: Statista, 2020; Россия – Data Insight, март 2021

Рисунок 1 – Сравнение объема и роста крупнейших рынков E-commerce в 2020 году [3].

В соответствии с вышеперечисленным, Е-commerce это обширная сфера, которая набирает обороты. Возможно, можно утверждать, что электронная коммерция создает новую культуру, когда в сознании людей формируется мысль о том, что купить товар или заказать

услугу можно, просто воспользовавшись устройством с выходом в Интернет, почти из любой точки мира.

Разработка E-commerce сервисов

Давайте рассмотрим вопрос с точки зрения разработки e-commerce сервисов, которых становится больше. Для того, чтобы создать успешную площадку, нужно разработать качественный и доступный программный продукт. Сервис должен быть доступен с любого устройства – телефона, планшета, ноутбука, компьютера, выдерживать большой поток пользователей, а также иметь красивый и удобный интерфейс. В основе e-commerce приложений в большинстве случаев используется клиент-серверная архитектура. В качестве клиентского приложения используются как web, так и мобильные приложения, а также приложения для персональных компьютеров. Большим плюсом веб приложения является то, что оно открывается с любого устройства – нужен лишь браузер. Но на практике у любого популярного e-commerce приложения есть хотя бы мобильное приложение, и в редких случаях приложение для персонального компьютера. Мобильное приложение как правило менее затратно по ресурсам, чем веб приложение, и дает больше возможностей для бизнеса и пользователя – позволяет отправлять push уведомления, хранить часть информации на телефоне, что позволяет пользоваться приложением даже оффлайн. Поэтому e-commerce приложение будет привлекательнее для пользователей и охватит большую аудиторию, если будет доступно на нескольких платформах (главные из которых Web, Android и iOS).

Кроссплатформенные решения

С точки зрения бизнеса поддерживать 3 различных платформы трудозатратно, потому что для каждой платформы нужна команда разработчиков, которая будет поддерживать проект. В настоящее время популярность набирают кроссплатформенные фреймворки, которые реализуют принцип «одна кодовая база – много платформ». Это означает, что написанное на таком фреймворке приложение компилируется в языки программирования, нативные для других платформ, и работает на различных платформах. Кроссплатформенные решения позволяют отказаться от необходимости писать отдельное приложение под каждую платформу. При описании кроссплатформенных фреймворков невозможно опустить тот факт, что они имеют некоторые ограничения. Каждая платформа имеет свои особенности взаимодействия с файловой системой, сетью, другими сервисами, и для обеспечения поддержки платформы все-таки нужны модули, написанные на нативных языках программирования. Однако разработчики фреймворков предоставляют возможность написания нативных модулей и предоставляют в своих решениях механизм взаимодействия кроссплатформенной кодовой базы с нативными модулями. Также стоит подчеркнуть, что благодаря развивающемуся сообществу разработчиков, использующих кроссплатформенные технологии, многие нативные модули уже написаны и лежат в открытом доступе. Таким образом, кроссплатформенные фреймворки, несмотря на свои ограничения, начинают занимать уверенное место среди технологий разработки клиентских приложений. Кроссплатформенных решений становится много и на этапе выбора технологий встает вопрос, какой кроссплатформенный фреймворк больше всего подходит для разрабатываемого

продукта. Для того, чтобы ответить на данный вопрос, проведем анализ и сравнение существующих фреймворков.

Анализ кроссплатформенных фреймворков

В настоящее время насчитывается более десятка кроссплатформенных фреймворков, но многие из них не пользуются популярностью среди разработчиков. Выделим наиболее используемые решения для анализа. В качестве критериев сравнения фреймворков выберем поддерживаемые платформы, удобство разработки, качество выходного продукта, развитость сообщества, а также сложность входа в разработку.

React Native – кроссплатформенный фреймворк, разработанный Facebook [5]. Фреймворк поддерживает все популярные платформы. Каждое React Native приложение работает в трех потоках. Основной поток обрабатывает отображение элементов пользовательского интерфейса и жесты пользователя, в то время как JavaScript поток выполняет JavaScript в отдельном движке, а за взаимодействие потоков отвечает отдельный модуль – bridge (мост). Третьим потоком является Shadow Thread, который отвечает за построение виртуального дерева пользовательского интерфейса для синхронизации состояния приложения с пользовательским интерфейсом. Во-первых, это позволяет использовать нативные для платформ элементы графического интерфейса, а для управления состоянием приложением использовать возможности веб фреймворка React и JavaScript библиотек (менеджеров состояний, таких как Redux и Mobx, и других вспомогательных библиотек). Также React Native обладает технологией Hot reload, что значительно ускоряет разработку приложений. Hot reload («горячая замена») – обновление приложения без его полной перезагрузки. Когда разработчик добавляет элемент интерфейса или вносит другие изменения приложения, то происходит перестройка элементов пользовательского интерфейса, но приложение остается на той же странице и разработчик сразу видит изменения. Изучение фреймворка React Native будет несложным для фронтенд разработчиков, в особенности для тех, кто использовал в разработке web фреймворк React. Однако для перехода с нативной разработки будет необходимо освоить особенности языка JavaScript и концепции фреймворка. React Native применяется во многих приложениях, таких как Facebook, Instagram, Skype и Discord.

Xamarin является кроссплатформенным фреймворком Microsoft [6]. Приложение Xamarin строится на архитектуре MVVM (Model-View-ViewModel). View представляет из себя слой презентации, View-Model представляет из себя слой бизнес-логики, который осуществляет работу со слоем данных и обрабатывает пользовательские события, Model, слой данных, отвечает за получение данных из источников данных (файловой системы, Интернета, сервиса геолокации и других). Примечательно, что Xamarin предоставляет возможность сделать отдельно пользовательские интерфейсы под разные платформы с помощью библиотек Xamarin.Android и Xamarin.iOS, но также дает возможность декларативно описать один пользовательский интерфейс, который на этапе компиляции преобразуется в нативные элементы для каждой платформы. Для разработки на Xamarin необходимо владеть языком C#. На Xamarin пишутся приложения для различных сфер: доставки, банков, медицины, социальных сетей и других.

Kotlin Multiplatform Mobile (КММ) – это кроссплатформенный фреймворк, разработка компании JetBrains [7]. Фреймворк создан для мобильной разработки под iOS и Android. КММ устроен таким образом, что часть приложения, которая не зависит от платформы, пишется на Kotlin, а пользовательский интерфейс пишется на нативном языке платформы. К общему коду относится взаимодействие с сетью, сериализация и десериализация данных, бизнес-логика, модели данных, хранение данных, состояние приложения. За счет использования нативных элементов пользовательского интерфейса скорость работы приложений, написанных с использованием фреймворка, близка к скорости полностью нативных приложений. Механизм реализации платформозависимой логики устроен через интерфейсы. Объявляются функции, классы, интерфейсы, перечисления и свойства, которые должны быть реализованы для каждой платформы отдельно. Для написания программ на Kotlin Multiplatform Mobile необходимо знать язык Kotlin. Kotlin стал основным языком для разработки Android приложений, поэтому КММ будет несложен в освоении для Android разработчиков. Kotlin Multiplatform Mobile используется в достаточно широком списке компаний: Philips, Yandex, Netflix и других.

Flutter – кроссплатформенный фреймворк, разработанный Google для создания приложений под мобильные устройства, компьютеры и web [8]. Flutter позиционируется как фреймворк для быстрой разработки с выразительным и гибким интерфейсом и производительностью, не уступающей нативным приложениям. Flutter написан на языке Dart и приложения с использованием данного фреймворка тоже пишутся на Dart. Для отрисовки приложений на Flutter используется графический движок Skia, который предоставляет независимый от платформы интерфейс взаимодействия с ним. Flutter не использует нативные элементы графического интерфейса, но предлагает обширный список элементов (в Flutter они называются виджетами), которые максимально похожи на нативные элементы, хотя ими не являются. Flutter из коробки предоставляет разработчику 2 библиотеки виджетов – Material и Cupertino. Первая представляет собой набор элементов пользовательского интерфейса, реализующий систему дизайна Material, разработанную Google, вторая – виджеты, повторяющие дизайн элементов пользовательского интерфейса приложений Apple. Несмотря на использование для отрисовки пользовательского интерфейса отдельного движка, производительность приложений на Flutter близка к нативной. Разработка на Flutter ведется в режиме отладки, для замера производительности приложения и выявления проблемных мест используется режим Profile, а для запуска в эксплуатацию приложение компилируется под различные платформы и этот режим называется Release. Для отладки используются точки останова, которые можно поставить в любом месте исполняемого кода. Они позволяют просмотреть состояние программы и исполнить программу по шагам. Для просмотра границ виджетов и анимации в замедленном режиме, а также оценки производительности приложения используются инструменты разработки Dart. Они помогают понять, какая часть программы использует слишком много ресурсов, что может негативно сказываться на работе приложения. Запустившись в 2017 году, Flutter успел собрать вокруг себя быстро развивающееся сообщество разработчиков, которые помогают друг другу в освоении фреймворка, добавляют функциональность в фреймворк и исправляют ошибки и создают и поддерживают пользовательские библиотеки, в том числе библиотеки с виджетами, библиотеки, реализующие работу в фоновом режиме, работу с уведомлениями и другие. Для Dart сделана система управления библиотеками Pub. Разработчики библиотек могут

выкладывать свои библиотеки в систему, а пользователи могут использовать данные библиотеки. Для просмотра и поиска библиотек используется сайт pub.dev, на котором можно найти библиотеку и посмотреть ее документацию. У фреймворка также есть свой YouTube канал, на котором команда Flutter выпускает новости фреймворка, гайды, а также обзоры популярных библиотек и полезных виджетов. Некоторую сложность входа в разработку на Flutter может нести язык Dart. Этот язык программирования очень похож на другие строго-типизированные объектно-ориентированные языки программирования, но не похож на JavaScript, поэтому будет сложен в освоении для JavaScript разработчиков. С другой стороны Flutter очень похож на React Native по применяемым в фреймворке концепциям, основная из которых – дерево элементов пользовательского интерфейса. Стоит отметить, что в документации Flutter есть руководства для Android, iOS, React Native, web, Xamarin разработчиков для более плавного освоения фреймворка. Flutter для разработки своих приложений используют крупные компании: Google, eBay, Philips, BMW, Toyota и другие.

Таблица 1 - Сравнение кроссплатформенных фреймворков

| | React Native | Xamarin | Kotlin Multiplatform Mobile | Flutter |
|---------------------|--|---|---|--|
| Команда разработки | Facebook | Microsoft | JetBrains | Google |
| Удобство разработки | Hot reload, Dev tools, Режим отладки | Hot reload | Hot reload | Hot reload, Dev tools, Режим отладки |
| Сообщество | Большое сообщество | Большое сообщество | Развивающееся сообщество | Самое большое сообщество |
| Производительность | Высокая, близкая к нативной | Высокая, близкая к нативной | Высокая, близкая к нативной | Очень высокая |
| Отрисовка UI | Нативные элементы | Нативные элементы | Не занимается отрисовкой UI | Виджеты |
| Платформы | Android TV, iOS, macOS, tvOS, Web, Windows и UWP | Android, iOS, Windows, macOS | Android, iOS | Android, iOS, Linux, macOS, Web, Windows |
| Сложность входа | Легкая для JavaScript разработчиков, для остальных сложная | Легкая для Android разработчиков, для остальных средняя | Легкая для Android разработчиков, для остальных средняя | Легкая для React Native разработчиков, для остальных средняя |

Заключение

В ходе рассмотрения кроссплатформенных фреймворков было выяснено, что самые популярные фреймворки обладают близкой друг к другу производительностью, хотя в разных задачах одни фреймворки несколько быстрее других [4], и инструментами разработки, и при выборе фреймворка необходимо прежде всего ориентироваться на то, какие языки и фреймворки знает команда разработки, насколько большое у него сообщество и насколько активно идет его поддержка. В статье были проанализированы самые удачные кроссплатформенные решения и каждое из них стоит того, чтобы к нему присмотреться, если перед вами стоит задача разработки приложений под несколько платформ.

Список литературы

1. А.В. Юрасов. Основы электронной коммерции: учеб. пособие – Горячая линия-Телеком, 2016. – 500 с.
2. Что такое e-commerce и как устроена онлайн-торговля [Электронный ресурс]. – Режим доступа: <https://trends.rbc.ru/trends/industry/607fe4549a7947027eaffbe6>
3. Как рынок eCommerce изменился за 2020 год, и какие настроения у потребителей в 2021 году? [Электронный ресурс]. – Режим доступа: <https://www.metacommerce.ru/blog/ecommerce/#:~:text=7%20%D1%82%D1%80%D0%BB%D0%BD%20%D1%80%D1%83%D0%B1,%D0%B7%D0%B0%202020%20%D0%B3%D0%BE%D0%B4.,%D0%B3%D0%BE%D0%B4%D0%BE%D0%BC%2C%20%D1%80%D0%BE%D1%81%D1%82%20%D1%81%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D0%B8%D0%BB%2078%25>.
4. An empirical investigation of performance overhead in cross-platform mobile development frameworks [Электронный ресурс]. – Режим доступа: <https://link.springer.com/article/10.1007/s10664-020-09827-6>
5. React Native [Электронный ресурс]. – Режим доступа: <https://reactnative.dev/>
6. Xamarin [Электронный ресурс]. – Режим доступа: <https://dotnet.microsoft.com/apps/xamarin>
7. Kotlin Multiplatform [Электронный ресурс]. – Режим доступа: <https://kotlinlang.org/docs/multiplatform.html>
8. Flutter [Электронный ресурс]. – Режим доступа: <https://flutter.dev/>

References

1. A.V. Yurasov. Osnovy elektronnoj kommercii: ucheb. posobie – Goryachaya liniya-Telekom, 2016. – 500 p.
2. What is e-commerce and how Internet commerce works [Electronic resource]. - Available at: <https://trends.rbc.ru/trends/industry/607fe4549a7947027eaffbe6>.
3. How has the e-commerce market changed in 2020 and what is the consumer sentiment in 2021? [Electronic resource]. - Available at: <https://www.metacommerce.ru/blog/ecommerce/#:~:text=7%20%D1%82%D1%80%D0%BB%D0%BD%20%D1%80%D1%83%D0%B1,%D0%B7%D0%B0%202020%20%D0%B3%D0%BE%D0%B4.,%D0%B3%D0%BE%D0%B4%D0%BE%D0%BC%2C%20%D1%80%D0%BE%D1%81%D1%82%20%D1%81%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D0%B8%D0%BB%2078%25>.

%B4%D0%BE%D0%BC%2C%20%D1%80%D0%BE%D1%81%D1%82%20%D1%81%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D0%B8%D0%BB%2078%25.

4. An empirical study of performance outperformance in cross-platform mobile development frameworks [Electronic resource] - Available at: <https://link.springer.com/article/10.1007/s10664-020-09827-6>.
 5. React Native [Electronic resource]. - Accessed at: <https://reactnative.dev/>.
 6. Xamarin [Electronic resource]. - Available at: <https://dotnet.microsoft.com/apps/xamarin>
 7. Kotlin Multiplatform [Electronic resource]. - Available at: <https://kotlinlang.org/docs/multiplatform.html>
 8. Flutter [Electronic resource]. - Available at: <https://flutter.dev/>
-