
ОЦЕНКА ТЕХНОЛОГИЙ ПРОЕКТИРОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА ОСНОВЕ ANDROID

Опивалов Сергей Александрович,

старший разработчик программного обеспечения, Кубанский государственный университет

Россия, г. Краснодар

6hundreds@gmail.com

Аннотация

В статье проанализированы возможности платформ и применяемых в их рамках технологий. Уточнены сферы их применения в практической деятельности. Выявлены достоинства и недостатки.

Ключевые слова: мобильное приложение, Android, Just-in-time-компиляция, компиляция кода

ASSESSMENT OF TECHNOLOGIES FOR DESIGNING MOBILE APPLICATIONS BASED ON ANDROID

Sergey A. Opivalov,

Senior Software Developer.

Kuban State University,

Russia, Krasnodar

6hundreds@gmail.com

ABSTRACT

The article analyzes the capabilities of the platforms and the technologies used within them. The spheres of their application in practice are clarified. Advantages and disadvantages are revealed.

Keywords: mobile application, Android, Just-in-time compilation, code compilation

На современном этапе большую популярность набирает проектирование мобильных приложений на основе операционной системы Android. Для этих целей достаточно технологий, позволяющих проектировать мобильные приложения и информационные системы для Android. Правильный выбор технологии разработки влияет на качество разработанного программного продукта. Де-факто используемая версия Android зависит от конкретного типа мобильного устройства.

Анализ практики показывает, что проектирование любых операционных систем ведется в двух основных направлениях – это, прежде всего, нативная и кроссплатформенная разработка, имеющая свою специфику. Сильной стороной нативной разработки является оптимизация для операционных систем, позволяющая им работать быстро. В сравнительном аспекте преимуществом кроссплатформенной разработки является возможность работы сразу с несколькими платформами, что делает их более универсальными. На современном этапе большинство технологий для подобных разработок в большей степени ориентированы на iOS и Android. Их отличает высокая скорость разработки и реализации продукта. Проблемным их моментом является нестабильная работа приложения, ограниченный функционал и архитектура. Сложилось несколько подходов к способам разработки. Традиционный подход учитывает необходимость написания кода и такие этапы, как разработка, тестирование, покрытие тестами. Второй подход, называемый zero-code, напоминает сборку конструктора, то есть готовых решений, не требующих написания кода, а приложение можно собрать из готовых для этого блоков.

Целью статьи является анализ возможностей технологий для проектирования мобильных приложений для Android.

В настоящее время в практике используется значительное разнообразие нативных, кроссплатформенных разработок для платформ, и технологий zero-code и low-code.

Так, в настоящее время установлено, технология zero-code не заменит традиционную разработку сложных приложений. Признание и широкое распространение получили такие платформы, как Webflow, Adalo, Bubble и др. К примеру, посредством Adalo при разработке создается пустой экран, куда добавляются элементы (списки, поля, кнопки и изображения) [2]. Для каждого из экранов характерна отдельная функция (регистрация, загрузка информации о товаре и др.). За экранами закрепляются действия, имеющие тесную связь с коллекциями данных. Выбор цветовых решений распространяется только лишь следующие области: для верхней части приложений, для кнопок, текста и фона, для цвета основного фона и основного текста. Работая с бесплатной версией, разработчик ограничен в выборе всего четырьмя типами шрифтов, хотя платная версия предлагает широкую номенклатуру шрифтов, представленных в библиотеке Google. Работая в приложении-конструкторе, у пользователя появляется возможность его настройки на сообщение с сервером с помощью GET-запроса, посылаемого через адрес URL. Кроме того, приложение можно создавать в виде файла с расширением apk с целью размещения в Play Store, также как и файлы Progressive Web Application. Однако, последние невозможно загрузить в приложении магазина. Для технологии zero-code платформа Adalo имеет базовое значение, так как дает возможность размещать свой продукт на Play Market, но в то же время по номенклатуре инструментов проигрывает конкуренцию платформам Webflow и Bubble. Соответственно, технология zero-code с применением бесплатных тарифов оптимально подходит для оперативного создания будущих образцов мобильных приложений.

В числе поставщиков технологий для традиционной разработки популярность получили такие среды разработки, как Flutter и Android Studio (Google). Помимо нативной разработки под Android существуют кроссплатформенные платформы React Native, Xamarin (Microsoft).

JavaScript является основой для разработки приложений с применением React Native. Взаимодействие основного потока и потока JavaScript образуют каждое приложение. Функцией основного потока является отображение элементов пользовательского интерфейса и жестов. В отличие от основного потока JavaScript обеспечивает бизнес-логику, и влияет на работу приложения [5]. Несмотря на сходства с базовой технологией React Native, JavaScript работает в фоновом режиме прямо на конечном устройстве,

взаимодействуя с собственной платформой посредством социализированных данных, асинхронного и пакетного мостов, обеспечивающих взаимодействие потоков. Компоненты React выступают оболочкой для существующего нативного кода, и взаимодействуют с нативным FPI (Functional Programming Interface) с помощью парадигмы декларативного пользовательского интерфейса JavaScript и React. Несмотря на тот факт, что существующие стили React Native отличает синтаксис, сходный с CSS, он не нашел применения в HTML либо CSS. Этот синтаксис заменяется сообщениями из потока JavaScript, которые применяются для управления собственными представлениями. Кроме того, возможности React Native позволяют написать оригинальный код для Android на основе возможностей Java и Kotlin, поэтому он позволяет разрабатывать мобильные приложения и полезен для команд, специализирующихся на web-разработке с применением фреймворка React.

На платформе Flutter применяется оригинальный язык Dart, который компилируется в бинарный код. Возможности данной платформы позволяют ей самостоятельно внутри своего движка отрисовывать интерфейс пользователя взамен применения нативных компонентов. Также на платформе Flutter применяется технология hot reload («горячая перезагрузка»), использование которой обеспечивает быстро изменить пользовательский интерфейс в ходе разработки, добавлять функции и корректировать ошибки, тестировать полученные изменения без необходимости переустановки приложений на мобильное устройство. Особенностью этой функции является способность работать с помощью внедрения обновленных файлов исходного кода в работающей виртуальной машине Dart [1]. В результате обновления виртуальной машиной классов с новой версией полей и функций, программная платформа Flutter автоматически перестраивает дерево виджетов, давая возможность просмотреть результаты организованных изменений. Принцип работы заключается в том, что виджеты должны описывать визуальное их представление с учетом текущего состояния и конфигурации приложения. В ситуациях изменения состояния виджета, он способен самостоятельно перестраивать свое представление. Изначально программная платформа Flutter спроектирована как расширяемая многоуровневая система, и функционирует как серия внешне независимых библиотек, зависящих от нижестоящего уровня. Выделяют три основных уровня: 1) фреймворк (представлен виджетами, анимацией и другими элементами, связанными с проектированием мобильного приложения, движок, обработку событий системы, виртуальная машина Dart); 2) уровень платформы; 3) уровень механизмов, имеющих отношение к платформе запуска). Ни один из уровней не является привилегированным для доступа к нижестоящим уровням, а каждая часть уровня фреймворка является заменяемой [4]. Соответственно, возможности платформы Flutter позволяют оперативно проектировать кроссплатформенные мобильные приложения высокого качества и с нативным интерфейсом на основе использования лишь одной команды специалистов для разработки, а дополнительные функции сокращают время на разработку.

Другой технологией, используемой в кроссплатформенной разработке, является Xamarin от Microsoft. Ее основное назначение связано с iOS и Android [3]. Разработка ведется на языке C#. Платформа Xamarin учитывает следующие основные продукты: Xamarin.Android и Xamarin.iOS. Для iOS исходный код компилируется прямо в собственный код сборки ARM, в то время как для Xamarin первым делом исходный код компилируется до промежуточного языка, и лишь потом в собственный код сборки в ходе выполнения Just-in-time-компиляции. Для разработчиков Xamarin предлагает создавать мобильные приложения с помощью ряда методов. В рамках первого метода предлагается применять Xamarin.iOS и Xamarin.Android как более функциональные платформенные продукты. В рамках второго метода применяются Xamarin.Forms как более продвинутая версия, применение которой дает возможность оперативно проектировать логотипы и

приложения, однако с меньшей функциональностью. По этой причине Xamarin.Forms оптимально подходит для приложений, где совместное использование кода имеет большее значение, чем настраиваемый пользовательский интерфейс. Эта особенность позволяет повторно применять до 99% кода, то есть создание интерфейса приложения один раз (на каждую платформу нет потребности создавать индивидуальный интерфейс). Кроме того, посредством Xamarin.Forms можно проектировать отдельные элементы пользовательского интерфейса, а другие элементы создавать с помощью своего набора инструментов пользовательского интерфейса. Таким образом, можно предположить, что данная платформа совместима не со всеми языками.

Подведение итогов статьи позволяет сделать вывод о том, что у каждой из платформ есть свои достоинства и недостатки. Возможности Webflow и Adalo позволяют в короткие сроки построить концепт приложения. React Native хорошо подходит для разработки кроссплатформенных приложений сразу для нескольких платформ. Xamarin подходит для разработок на языке C#.

Список литературы:

1. Заметти Ф. Flutter на практике: Прокачиваем навыки мобильной разработки с помощью открытого фреймворка от Google / пер. с англ. А. С. Тищенко. – М.: ДМК Пресс, 2020. – 328 с.
2. Крамаренко Т.А. Разработка бизнес-приложений / Т.А. Крамаренко, Е. А. Иванова / Учебное пособие. – Краснодар: КубГАУ, 2021. – 233 с.
3. Меньшиков В.Е., Омельченко Д.А., Фешина Е.В. Тенденции разработки мобильных приложений // Информационное общество: современное состояние и перспективы развития: сб. материалов XII междунар. студенческого форума. – Краснодар: КубГАУ, 2019. – С. 350–352.
4. Нагорный Н.Н. Применение agile подходов в разработке программного обеспечения веб-приложений / Наука и бизнес. №
5. Способы защиты мобильного приложения под Android / В.А. Груммет, О.А. Лисовин, Е.В. Фешина, С.А. Куштанок // Наука XXI века: проблемы, перспективы и актуальные вопросы развития общества, науки и образования: материалы междунар. межвуз. осенней науч.-практ. конф., пгт. Яблоновский, 25 сентября 2020 года. Краснодар: Краснодарский ЦНТИ – филиал ФГБУ «РЭА» Минэнерго России, 2020. С. 53–57.
6. Bonnie Eisenman Learning React Native, 2nd Edition November 2017. O'Reilly Media, Inc. 432 p.

References:

1. Note F. Flutter in practice: Upgrading mobile development skills using an open framework from Google / transl. from eng. A. S. Tishchenko. M.: DMK Press, 2020. 328 p.
2. Kramarenko T.A. Development of business applications / T.A. Kramarenko, E. A. Ivanova / Textbook. Krasnodar: KubGAU, 2021. 233 p.
3. Men'shikov V.E., Omelchenko D.A., Feshina E.V. Trends in the development of mobile applications // Information society: current state and development prospects: Sat. Materials XII Intern. student forum. Krasnodar: KubGAU, 2019. Pp. 350-352.

4. Nagorny N.N. Applying Agile Approaches to Web Application Software Development / Science and Business. No.
5. Ways to protect a mobile application for Android / V.A. Grummet, O.A. Lisovin, E.V. Feshina, S.A. Kushtanok // Science of the XXI century: problems, prospects and topical issues of the development of society, science and education: materials of the international. interuniversity autumn scientific and practical. conf., town. Yablonovsky, September 25, 2020. Krasnodar: Krasnodar TSNTI - branch of the Federal State Budgetary Institution «REA» of the Ministry of Energy of Russia, 2020. Pp. 53–57.
6. Bonnie Eisenman Learning React Native, 2nd Edition November 2017. O'Reilly Media, Inc. 432 p.