

СРАВНЕНИЕ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ДЛЯ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Е.А. Попова

Воронежский государственный технический университет

Аннотация: в статье проводится сравнительный анализ нативного и кроссплатформенного подхода к разработке мобильных приложений и программного обеспечения для их реализации в зависимости от выбранной технологии.

Ключевые слова: мобильные приложения, кроссплатформенный подход, нативная технология, гибридные приложения, среда разработки, мобильная платформа, инструментальные средства, операционная система.

COMPARISON OF DEVELOPMENT TOOLS FOR MOBILE APPLICATIONS

Е.А. Popova

Voronezh state technical University

Abstract: the article provides a comparative analysis of the native and cross-platform approaches to the development of mobile applications and software for their implementation, depending on the selected technology.

Keywords: mobile applications, cross-platform approach, native technology, hybrid applications, development environment, mobile platform, tools, operating system.

Тактика мобильной разработки определяет много важных причин, одна из них - адаптированность с разными мобильными платформами, этот факт является одним из ключевых вопросов перед написанием приложения. Статистика показывает, что за последние несколько лет количество мобильных приложений выросло на несколько тысяч в год. При такой быстро возрастающей конкуренции, перед разработчиками возникает задача - как быстро разработать приложение, чтобы оно не успело устареть к моменту выпуска, чтобы конечный продукт был не только конкурентоспособным, а также имел возможность работать на различных платформах и устройствах.

Существуют следующие виды мобильных приложений:

Нативные приложения - прикладные программы, разработанные для использования на определенной платформе или на определенном устройстве. Для каждой платформы существует свой нативный язык, родной для каждой операционной системы и рекомендуемый производителем. Нативное приложение встраивается в программное обеспечение мобильного устройства и загружается через официальный магазин Apple, Google Play и Windows. Важным достоинством нативной разработки является возможность использования им всех функций системы, например, календарь, камера, микрофон и т.д., при этом программное приложение минимально тратит память устройства и заряд аккумулятора.

Мобильные операционные системы, имеющиеся в большом количестве, побудили к созданию кроссплатформенной технологии разработки. Нативный подход предполагает написание отдельного кода приложения под каждую мобильную платформу, а при использовании кроссплатформенной технологии программный код пишется один на все необходимые операционные системы. Кроссплатформенные приложения применяют механизм браузера, поэтому подходят для большого количества устройств. Немаловажным фактом является и то, что данный вариант требует меньше финансовых вложений при создании продукта.

Гибридные приложения [1] объединяют особенности нативной и кроссплатформенной разработки - это кроссплатформенное приложение внутри «родной» оболочки. Интерфейс так же, как и в кроссплатформенном приложении использует браузер телефона, но элементы, которые требуют отклика и высокой производительности разрабатываются на родных языках. Они создаются теми же технологиями, что и сайты (при помощи HTML, JavaScript,

CSS), а затем код «оборачивается» в нативную оболочку. Такая оболочка является хостом для браузера, который настраивается на запуск приложения по определенному URL.

Нативная разработка необходима если:

1. Имеет значение производительность приложения и удобство интерфейса;
2. Предполагается постоянное использование системных ресурсов и функций мобильных устройств: камер, GPS, акселерометра и прочих датчиков;
3. Важно низкое энергопотребление приложения;
4. Требуется сложный интерфейс и/или сложная анимация;
5. Требуется интеграция со сторонними ресурсами (API и т.д.).

Кроссплатформенные решения выгодны в случаях, если:

1. Требуется написать прототип приложения под несколько платформ за сжатые сроки, но при этом не планируется этот прототип развивать дальше;
2. При разработке сложных игр существуют мощные кроссплатформенные решения, например, Unity3D, имеющие богатый встроенный функционал для разработчиков;
3. Если приложение берет всю информацию с сервера, анимация и интерфейс не очень важны, а логика уже реализована на сервере;
4. Если скромная производительность не важна, а контент подгружается из интернета.

При разработке нативного Android - приложения код программы «лежит» в Java Virtual Machine, а она в свою очередь в операционной системе (ОС) Android. При использовании инструментальных средств для кроссплатформенных приложений происходит тоже самое, только цепочка вложения кода, следующая: JavaScript код программы, веб-браузер, код "нативного приложения" средств разработки, Java Virtual Machine, ОС Android.

Почти таким же образом реализована работа веб-приложений (JavaScript код программы) и на других платформах. ОС iOS и Windows Phone также компилируются средствами для кроссплатформенной разработки.

Инструментальные средства разработки нативных мобильных приложений:

Eclipse [2] – программное обеспечение (ПО) с открытым исходным кодом. Инструментом сборки проекта выступает Ant. Eclipse легко настроить и интегрировать с необходимыми компонентами, например, SDK Android, NDK и Java Machine. В большинстве случаев данная среда разработки используется для создания Java-приложений, но в ней существуют дополнения и для других языков, например, C/C++, Ruby, PHP и т.д. Компания Google выпустила комплекты разработки для этой среды, что позволяет создавать мобильные приложения для Android и App Engine.

Netbeans - ПО для разработки на языке Java, но имеет возможность использовать и другие языки. Данная среда ускоряет процесс разработки настольных приложений и программ с drag&drop GUI Builder. Но данная возможность заметно влияет на производительность редактора кода. Программа распространяется бесплатно и имеет открытый исходный код.

Visual Studio - Родная среда разработки для ОС Windows. Предоставляет возможность в большом количестве выбора языков программирования (C/C++, C#, VB.NET, F# и др.), имеет функции проверки синтаксиса, автодополнения кода, конструктора форм, пошаговой отладки, создание схемы базы данных, диаграммы классов и т.д.

IntelliJ IDEA [3] - предоставляет инструменты для создания коммерческих, мобильных и веб-приложений. Программное обеспечение имеет в своем составе большое количество необходимых инструментов, для создания приложений: контроль версий, базы данных, средства сборки, терминал, автозаполнение кода, FTP, проверка синтаксиса и т.д.

Sublime Text - универсальный, быстрый редактор кода, поддерживающий большое количество языков программирования. Имеет возможность разделения экрана на проекты, создание макросов, множественного выделения текста, сворачивания кода и др. У редактора есть лицензионная версия, однако им можно пользоваться и без ее наличия, т.е. бесплатно.

Notepad++ - Легкий, но имеющий широкий и многочисленный функционал редактор кода для ОС Windows. Поддерживает практически все популярные языки программирования, создание макросов, FTP браузер, разделение экрана и другие важные функциональные особенности для удобной и комфортной работы.

Android studio - ПО, выпущенное как результат сотрудничества компаний Google и JetBrains. Среда предназначена для разработки мобильных приложений для ОС Android. Программа имеет много общего со средой разработки IDEA. Основным различием является то, что в качестве сборщика в Android studio применяется Gradle, а не Ant. Все обновления, которые выходят для одной из этих сред разработки позже добавляются и в другую.

При создании программного приложения с применением кроссплатформенной технологии происходит добавление к исходному коду определённой «оболочки», написанной на языке Java, транслирующей вызовы к системе и от неё. Фактически создаётся мобильный сайт, который «оборачивается» небольшим платформенным кодом.

Кроссплатформенные средства разработки:

Apache Cordova - ПО для создания приложений имеющая открытый исходный код. Данная среда использует HTML5, CSS3 и JavaScript технологии. Мобильные приложения выполняются внутри оболочки, покрывающей код, написанный программистом. Используемая "оболочка" кода направлена на работу на любой платформе и полагается на стандартные API для возможности доступа к датчикам, данным устройства и состоянию сети. Для такого подхода создано огромное количество фреймворков, но все они делают фактически одно и то же. Различие между ними в том, что Cordova не задаёт ограничений и шаблонов на логику и интерфейс (UI) для HTML5-проекта, а фреймворки оперируют собственными UI-элементами, имитирующими мобильные платформы, и своей логикой разработки.

PhoneGap - бесплатное и открытое ПО, работающее на основе HTML, CSS, JavaScript. Приложения написанные в данной среде работают как web - страницы внутри WebView - компонента платформы, позволяющего вкладывать web - страницы в приложения Android. Компонент API представляет разработчикам возможность пользоваться программными функциями устройства, например, звук, камера, геолокация, уведомления и т.д. Приложения, созданные в PhoneGap допускаются внедрить в нативную разработку, для последующей разработки гибридного приложения. В данном ПО можно написать программное приложение под любую мобильную операционную систему.

Xamarin - среда разработки от компании Microsoft для создания кроссплатформенных приложений, под популярные платформы: Android, iOS, Windows Phone. В Xamarin используется язык программирования C#. Код приложения не интерпретируется при выполнении, а сразу компилируется в нативный. Поэтому у приложений, разработанных в Xamarin производительность и реакция на отклики такие же, как у нативных продуктов. Улучшает качество приложения, но вместе с этим увеличивает время на разработку, тот факт, что дизайн создается при помощи стандартных для каждой платформы инструментов.

React Native - среда разработки с открытым исходным кодом от компании Facebook для создания кроссплатформенных приложений с использованием языка JavaScript. В React Native можно писать код только для Android и iOS платформ. С помощью данного ПО можно использовать не компонент WebView, а встроенные компоненты для взаимодействия с родными объектами каждой платформы. React Native использует JavaScript API поверх нативных элементов. Среда выполняет JavaScript код в изолированном потоке, то есть в JS потоке накапливаются различные команды и в определенный этап сгруппированные запросы

отправляются в главный поток. Это обеспечивает отсутствие блокирования главного потока для выполнения программы.

Flutter - среда разработки превращающая код, написанный разработчиком, в нативный. Далее этот код не выполняется платформой, а программа сама отрисовывает все элементы дизайна. Языком программирования выступает, новая версия JavaScript от компании Google, Dart. У этого есть как преимущества (внешне идентичные интерфейсы), так и недостатки (прорисовка интерфейса требует определённых затрат памяти и процессорного времени).

На данный момент среди ПО для разработки мобильных приложений представлено большое количество средств, позволяющих снизить затраты и ресурсы на создание программных продуктов. Результаты анализа представлены в таблице 1.

Таблица 1

Сравнительный анализ средств разработки мобильных приложений

ПО	Кроссплат.	Языки программирования	Платформы	Распространение	Доп. ПО	Этапы ЖЦ	Доступ к нативным UI эл-там
Eclipse	Нет	C, C++, Fortran, Perl, PHP, JS, Python и др.	Android	Беспл.	Да	Разработка, тестирование	Нет
Netbeans	Нет	Java, PHP, JS, Python, C, C++, Ада, Ruby и др.	Android	Беспл.	Да	Разработка, тестирование	Нет
Visual Studio	Нет	CSS, HTML, Java, C++, C#, Python, Swift, Ruby и др.	Android, iOS	Беспл. и плат.	Нет	Весь цикл	Нет
IntelliJ IDEA	Нет	Kotlin., Java, JS, C, C++, PHP, Python и др.	Android	Беспл. и плат.	Да	Разработка, дизайн, тестирование	Нет
Sublime Text	Нет	C, C++,C#, CSS, HTML, Java, JS, PHP, Python и др.	Android, iOS	Беспл. и плат.	Нет	Дизайн, разработка,	Нет
Notepad ++	Нет	C, C++, Java, C#, XML, PHP, HTML, CSS и др.	Android, Windows Phone	Беспл.	Нет	Разработка	Нет
ПО	Кроссплат.	Языки программирования	Платформы	Распространение	Доп. ПО	Этапы ЖЦ	Доступ к нативным UI эл-там
Android studio	Нет	Kotlin, Java, C++	Android	Беспл.	Да	Весь цикл	Есть
Apache Cordova	Да	HTML5, CSS, JavaScript	Android, webOS, iOS,	Беспл.	Нет	Дизайн, разработка, тестирование	Есть
Phone	Да	HTML5,	Windows	Беспл.	Нет	Разработка	Есть

Gap		CSS, JavaScript	Mobile, Symbian, BlackBerry, Windows Phone 7				
Xamarin	Да	C#	iOS, Android, Windows Phone	Беспл. и плат.	Нет	Разработка, тестирование	Нет
React Native	Да	Java Script	iOS, Android	Беспл.	Да	Разработка	Есть
Flutter	Да	Dart	Google Fuchsia, iOS, Android	Беспл.	Да	Дизайн, разработка, тестирование	Нет

Нативные мобильные приложения собираются при помощи компилятора Java, кроссплатформенные собираются при помощи дополнительных средств, которые в свой шаблонный код Java приложения, вставляют JavaScript код, написанный разработчиком, а затем вызывают Java компилятор, чтобы собрать приложение. Когда запускается приложение, оно обращается к встроенному в телефон веб-браузеру и «просит» выполнить JavaScript код. Из этого вытекает кроссплатформенность.

В настоящее время сложно сделать выбор между платформами iOS и Android, так как процент популярности и использования между ними примерно одинаков. И если речь касается мобильных приложений, то, скорее всего, придется писать приложение для обеих платформ. При использовании кроссплатформенной разработки объем трудозатрат уменьшается вдвое. Учитывая рыночные цены разработки под iOS, которая выше цены разработки под Android, можно сделать вывод, что кроссплатформенный подход для реализации типичного мобильного приложения дает экономию финансовых вложений больше чем вдвое.

Библиографический список

1. Ерошенко М. Н. Гибридные мобильные приложения: основные характеристики и особенности // Вестник магистратуры – 2015. №5 (44) Том 1 – С. 19 – 20
2. Сосновская А.С. Обзор технологий для разработки приложений для операционной системы Android // Россия сегодня: безопасность, сотрудничество, развитие. Взгляд молодых. – 2016. – С. 236–238
3. . Березовская Ю.В. [и др.] Введение в разработку приложений для ОС Android. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 433 с.

Информация об авторах

Попова Елена Александровна – магистрант кафедры систем управления и информационных технологий в строительстве, Воронежский государственный технический университет (394006, Россия, г. Воронеж, ул. 20 лет Октября, 84), e-mail: EvaAilen@mail.ru

Information about the authors

Popova Elena Alexandrovna, Master's Degree student at the Department of management systems and information technologies in construction, Voronezh state technical University (84, 20 let Oktyabrya str., Voronezh, 394006, Russia), e-mail: EvaAilen@mail.ru

