

ОБЗОР ВОЗМОЖНОСТЕЙ ПЛАТФОРМЫ FLUTTER

Петровская В.В., Деменковец Д.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Деменковец Д.В. – старший преподаватель

В тексте работы анализируются возможности платформы Flutter SDK. Описываются преимущества данной платформы. Выполняется сравнение подходов к разработке и применение для создания приложений с использованием технологии Flutter.

Flutter SDK (от англ. software development kit) – бесплатное кроссплатформенное средство от Google с открытым исходным кодом для быстрой разработки мобильных приложений для iOS и Android. Позволяет быстро и качественно создавать пользовательские интерфейсы [1]. Был представлен Flutter в 2017 году, официально выпущен в декабре 2018 года компанией Google. Данная технология позволяет разработчикам программного обеспечения создавать кроссплатформенные приложения, которые можно запускать на различных операционных системах, таких как iOS или Android. Логотип платформы Flutter приведен на рисунке 1.

Главной целью разработчиков Flutter SDK было создать платформу проектирования пользовательских интерфейсов для любого размера экрана за счет поддержки технологии для мобильных (mobile applications), настольных (desktop applications), веб-приложений (web applications) и встраиваемых систем (embedded systems). Платформа получила широкое распространение еще до выхода первой версии. К примеру, наиболее популярные приложения, использующие технологию Flutter – Xianyu, GoogleAds, Hamilton Musical app, Reflectly [3].



Рисунок 1 – Логотип платформы Flutter

В 2015 году компания Facebook представила миру платформу для разработки мобильных приложений React Native. Ответом от Google явилось создание платформы Flutter, была представлена в 2017-ом. Обе технологии являются кроссплатформенными, однако подходы к разработке отличаются в значительной степени. React Native использует мост JavaScript. Данный мост служит для связи с модулями платформы и является так называемой прослойкой между приложением и устройством. Данный подход в значительной степени снижает производительность приложения. В платформе Flutter мост отсутствует, Flutter выполняет отрисовку визуальных компонентов пользовательского интерфейсов «сам», а обращение к сервисам устройства происходит через так называемый Platform Channels, буфер обмена между платформой и кодом программы [3].

Платформа Flutter использует язык программирования Dart. Команда разработчиков этого языка пыталась использовать больше десятка языков программирования, включая JavaScript, Objective-C и Java, однако остановилась на языке Dart, который был создан Google в октябре 2011. Dart позиционируется в качестве альтернативы JavaScript. Язык долгое время не пользовался популярностью среди разработчиков, и в 2018 году и вовсе был назван самым худшим языком для изучения. Однако Dart имеет ряд преимуществ, которые и обусловили такой выбор разработчиков Flutter.

Язык программирования Dart позволяет работать с JIT компиляцией и делать горячее развертывание или установку приложения (hot reload) во время разработки. Благодаря возможности обновления приложения с сохранением состояния в значительной степени сокращается цикл разработки приложения. Во время выполнения сборки итогового приложения (release build) код компилируется непосредственно в машинные инструкции, которые выполняются на устройстве. За счет этого достигается скорость выполнения операций, сравнимая с Objective-C, Swift, Java или Kotlin. Язык Dart обладает опциональной, строгой типизацией, поддерживает реактивный стиль программирования, обладает хорошей поддержкой управления объектами с помощью прогрессивного сборщика мусора. Кроме этого Dart обладает схожим с языком Java синтаксисом и прост в освоении.

Платформа Flutter стремится обеспечить производительность 60 кадров в секунду (fps) или производительность 120 кадров в секунду на устройствах с частотой обновления экрана 120 Гц. Отображением занимается графическая библиотека под названием Skia. Это быстрая 2D графическая библиотека, разработанная на языке C++. Кроме графической Skia под капотом у Flutter текстовый движок, заимствованный из Android. Это позволяет быстро обновлять исходный код при

внесении изменений [3].

Flutter не использует визуальные элементы интерфейса платформы, он предоставляет свои собственные. Создатели Flutter SDK разработали свои пакеты этих интерфейсов на языке Dart. Flutter отвечает за прорисовку каждого пикселя, за запуск анимации, за обработку жестов. Благодаря этому достигается максимальное сходство в работе на различных устройствах и минимальное влияние операционной системы на итоговый внешний вид приложения. Для дополнительной производительности в работе пользовательского интерфейса визуальные элементы перерисовываются в случае необходимости. Например, когда изменился внешний вид.

Для построения пользовательского интерфейса (user interface) во Flutter используется декларативный подход. В распоряжении разработчиков два набора визуальных элементов для основных мобильных платформ: Material для Android и Cupertino для iOS. Таким образом, разработчики Flutter заново разработали и выполнили отображение на устройствах все пользовательские визуальные элементы (UI-компоненты), полностью повторив их поведение. Непосредственное аппаратное взаимодействие устройства (географическая локация, звук, bluetooth, GPS, акселерометр) с платформой происходит через Platform Channels [4].

Приложения разработанные с помощью платформы Flutter легко тестировать. Технология поддерживает модульное тестирование (unit testing), тестирование визуальных компонентов пользовательского интерфейса (widget testing) и интеграционное тестирование. Модульные тесты выполняют тестирование одиночных функций, методов или классов. Тестирование визуальных компонентов позволяет убедиться, что пользовательский интерфейс выглядит и работает корректно. Интеграционные тесты проверяют взаимодействие модулей и всего приложения в целом.

Интерфейс командной строки платформы Flutter позволяет настраивать концепцию непрерывной интеграции и доставки CI/CD (англ. Continuous integration & Continuous delivery). Концепция, применяемая для выявления нарушений в работе приложения при внесении изменений. Для приложений Flutter имеется встроенную поддержку CI/CD, где используется Nevercode - платформа на основе облачных вычислений.

Flutter позволяет создавать сложные анимации, добавлять в приложения локализацию и менять ее в режиме выполнения приложения (run time). Технология активно развивается, разработчики создают еще больше готовых библиотек, которые легко подключаются в разрабатываемый проект. Не маловажным плюсом Flutter является документация, подробная информация служит отличным помощником разработчикам приложений.

В настоящее время компания Google разрабатывает новую операционную систему Fuchsia. Fuchsia использует платформу Flutter в качестве движка прорисовки пользовательского интерфейса. Google также планирует адаптировать все существующие приложения для Android на Fuchsia (включая среду выполнения приложений платформы Android). Особенностью разрабатываемой системы является способность работать на любой архитектуре центрального процессора. В основе этой системы – микроядро Zircon от Google, которое способно работать на смартфонах, настольных компьютерах и встроенных системах.

Flutter – это новая, современная и актуальная технология. Приложения со временем устаревают, появляются новые идеи и решения. Flutter позволяет воплотить идею в готовый продукт с быстрой скоростью и меньшими ресурсами. Платформа находит отклик и вызывает симпатию как у разработчиков, так и у дизайнеров и бизнеса. Flutter позволяет контролировать каждый пиксель на экране и создавать одинаковый интерфейс для разных устройств, для разработки приложения, как правило, необходима только одна команда разработчиков и относительно небольшие временные и материальные ресурсы.

Существует много источников, позволяющие найти нужную и понятную информацию о Flutter: хорошая документация, простые и информативные руководства, лаконичные и наглядные видео из серии «Flutter Widget of the Week», увлекательное и необычное шоу «The Boring Flutter Development Show» и многое другое.

Flutter – молодая технология, поэтому имеет небольшое, но активно развивающееся сообщество. Можно сказать, что данная технология «ловит волну успеха», вдохнув вторую жизнь в язык программирования Dart. Разработчики платформы предлагают новые интересные архитектурные решения и эти факторы предвещают длительное и успешное существование этой платформы.

Список использованных источников:

1. Flutter уроки. От идеи до приложения // Flutter уроки [Электронный ресурс]. – 2020. – Режим доступа: <https://flutter.su/> Дата доступа: 20.03.2020
2. Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. // Flutter [Электронный ресурс]: веб-сайт – Режим доступа: <https://flutter.dev> Дата доступа: 23.03.2020
3. Дыцяк, В. Стоит ли инвестировать во Flutter. Сравнение Flutter и React Native [Электронный ресурс] / В. Дыцяк // Портал сообщества разработчиков. – Режим доступа: <https://dou.ua/lenta/articles/flutter-for-mobile-apps>. – Дата доступа: 23.03.2020
4. Jagtap, S. Flutter vs React Native: A Developer's Perspective [Электронный ресурс] / S. Jagtap // Nevercode blog. – Режим доступа: <https://nevercode.io/blog/flutter-vs-react-native-a-developers-perspective>. – Дата доступа: 23.03.2020