Фадеев А.Ю.,
студент
Филиал РГППУ в г.Нижний Тагил
Волкова Е.А.,
к.п.н., доцент кафедры ИТ
Филиал РГППУ в г.Нижний Тагил
г. Нижний Тагил, Россия

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Аннотация

В статье рассматриваются основные факторы, которые определяют выбор программного продукта для разработки мобильного приложения. Также приведен пример жизненного цикла мобильного приложения и выделены основные его особенности.

Ключевые слова: мобильное приложение, Android Studio, жизненный цикл, анализ мобильных приложений

.

4+95***Fadeev A.Y.,***
student
Branch rgppu in Nizhny Tagil
Volkova E. A.,
Ph. D., associate Professor of the Department of it
Branch rgppu in Nizhny Tagil
Nizhny Tagil, Russia

COMPARATIVE SOFTWARE ANALYSIS FOR MOBILE APP DEVELOPMENT

Abstract

The article examines the main factors that determine the choice of software product for mobile application development. It also illustrates the life cycle of mobile applications and highlighted its main features.

Keywords: mobile application, Android Studio, life-cycle analysis of mobile applications

Современный человек делает все для того чтобы достигнуть максимального комфорта и удобства в своей жизни. Для этого сегодня специалистами в области информационных технологий разрабатываются мобильные приложения, которые позволяют решать огромное количество задач в разных областях жизнедеятельности человека.

Все мобильные приложения условно можно поделить на программы для рабочих целей и на развлекательные программы. Первые позволяют контролировать и оптимизировать рабочие процессы, составлять аналитическую отчетность, выполнять иные функциональные задачи. Вторые – позволяют интересно и разнообразно проводить время.

Однако, как показывает практика, большим спросом сегодня пользуется специализированный софт. Также именно на таких программах можно делать неплохие деньги, ведь современные компании не жалеют инвестиций в продукты, которые могли бы в какой-либо степени оптимизировать или упростить имеющиеся бизнес-процессы.

На протяжении последних лет показатель, характеризующий уровень спроса на мобильные устройства, постоянно растет. Такая статистика позволяет сделать вывод о том, что разработка мобильных приложений актуальна и целесообразна.

Мобильное приложение – это программный продукт, предназначенный для использования на мобильных устройствах оснащенных операционной системой. Мобильные приложения могут быть установлены на устройстве с завода изготовителя либо скачаны с флэш – носителей или загружены из онлайн магазинов, где за это может взиматься плата либо доступны в бесплатном доступе.

Для того чтобы наглядно рассмотреть достоинства и недостатки существующих типов мобильных приложений, приведем в качестве примера сравнительную таблицу основных критериев см. табл. 1.

Таблица 1

Сравнение основных видов мобильных приложений.

Вид мобильного приложения	Доступ к функционалу устройства	Скорость работы	Стоимость разработки	Распространение через магазин	Процесс одобрения
Нативный	полный	очень высокая	высокая	доступно	обязательный
Гибридный	полный	очень высокая	приемлемая	доступно	малозатратный
Веб – приложение	частичный	высокая	приемлемая	не доступно	отсутствует

Классифицировать мобильные утилиты можно по нескольким типам, например по разновидности работы:

1. Приложения переднего плана. К ним относят программы работающие в моменты, когда никаких других не активизировано, например мобильные игры.
2. Фоновые приложения. Используются в те моменты, когда требуется произвести настройку

3. Смешанные программы. Работают в обоих выше перечисленных режимах, хотя и располагают определенной степенью интерактивного воздействия. К ним можно отнести мобильные антивирусные программы.

4. Виджиты. Приложения отображающие информационные сообщения на рабочем столе. В качестве примера можно привести утилиты, предоставляющие информацию о заряде батареи мобильного устройства или время.

5. Сложные приложения. К ним относятся, например, программа – утилита, которая включает в себя следующие инструменты: очистка кэш памяти, отображение информации о мобильном устройстве, удаление установленных приложений.

Классификация мобильных приложений по роду деятельности:

1. Контентные приложения. Обладают большой популярностью, основные задачи которые они выполняют это: прослушивания музыки, просмотры фильмов и фотографий, чтения цифровых книг и журналов. Так же к ним можно отнести информационные приложения, например предоставляющие информацию о погоде, расписания городского транспорта, свежих новостях, рецепты или разработанные специально к каким либо намечающимся событиям, таких как спортивные чемпионаты, выставки или форумы. Ну и конечно специальные рекламные приложения.

2. Бизнес приложения. Сделаны для помощи в офисной работе, расчетах, обмене служебными данными, а также обеспечивающие доступ к интернет –магазинам, платежным системам и банковским счетам. На данный\ момент сегмент бизнес –приложений является более интересным для инвесторов, но сложность состоит в переводе бизнес –задач на мобильные устройства.

3. Мобильные игры – это наиболее востребованный сектор мобильных приложений.

4. Мобильные социальные сети. Данный вид с каждым днем набирают все большую популярность, увеличивая многочисленную аудиторию во всех странах мира, чему способствует развитие мобильного интернета расширяющего свою доступность по всей планете.

Классификация мобильных приложений по виду монетизации:

1. Платное приложение, реализуемое посредством продажи в магазине.

2. Бесплатное приложение с платной подпиской.

3. Бесплатное приложение со встроенными покупками.

4. Бесплатное приложение с рекламой внутри приложения.

Жизненный цикл – множество процессов, происходящих с момента изначального утверждения решения о разработке программного продукта, до его окончательного вывода из использования.

Модель жизненного цикла – это схема выполнения определенных задач в процессах, поддерживающих разработку, эксплуатацию и сопровождение

программного обеспечения, а так же отражающая жизненный цикл, начиная от первоначальных требований к ней и довывода из эксплуатаций.

Разработка модели жизненного цикла основывается на изначальной идее проектируемого программного обеспечения, его стандартов, предоставляющих возможность сформировать схему исполнения работ по предпочтению разработчика и заказчика. Модель жизненного цикла делится на процессы осуществления, которые включают в себя работы и задачи, производимые в определенном процессе, и при их окончании реализовывать переход к следующим процессам модели. Моделей жизненного цикла существует множество, но только три из них классифицируются как основные: каскадная, спиральная, эволюционная.

Каскадная модель жизненного цикла (см. рис. 1) реализует, принцип одинарного исполнения каждого из основных процессов и этапов в их определенных рамках. Переход на следующий этап реализуется после того, как будет произведена работа на текущем этапе, и откатов на осуществленные стадии не предусмотрено. Каждый этап заканчивается приобретением определенного результата, который используется в качестве базовой информаций для следующего этапа. Достоинства и недостатки каскадной модели жизненного цикла можно рассмотреть в таблице 2.

Рис. 1. Каскадная модель жизненного цикла.

Таблица 2

Достоинства и недостатки каскадной модели жизненного цикла.

Критерий оценки:	Признаки:
Достоинства:	– на каждой стадии формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности; – выполняемые в логичной последовательности стадии работ позволяют планировать сроки завершения всех работ и соответствующие затраты.
Недостатки:	–реальный процесс создания программного обеспечения никогда полностью не укладывался в такую

	жесткую схему. Результаты очередной стадии часто вызывают изменения в проектных решениях, выработанных на более ранних стадиях.

Исходя из необходимости редактирования процессов и промежуточного продукта, была реализована спиральная модель (см. рис. 2).

Рис. 2. Спиральная модель жизненного цикла.

В данной модели жизненного цикла допускается анализ программного обеспечения на витке разработки, его проверку, оценивание правильности и принятия решения о переходе на виток реализаций выше либо откат для осуществления доработки. Различие данной модели от каскадной выражается в том что, спиральная модель обеспечивает многократное возвращение к изначальному этапу формулировки требований и повторному возобновлению разработки от любого этапа произведения работ. Каждая версия разработки системы соответствует витку спирали жизненного цикла. При возникновений необходимости редактирования системы в определенном этапе, обязательно осуществляются корректировки в заранее зафиксированные требования, после чего производится откат к предыдущему процессу витка спирали для дальнейшей разработки новой версии системы с учетом поправок. Достоинства и недостатки спиральной модели можно просмотреть в таблице 2.

Таблица 2

Достоинства и недостатки спиральной модели жизненного цикла.

Критерий оценки:	Признаки:
Достоинства модели:	– позволяет быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований;

Критерий оценки:	Признаки:
	– допускает изменение требований при разработке информационной системы, что характерно для большинства разработок, в том числе и типовых; – обеспечивает большую гибкость в управлении проектом; – позволяет получить более надежную и устойчивую систему; – позволяет совершенствовать процесс разработки – анализ, проводимый в каждой итерации, позволяет проводить оценку того, что должно быть изменено в организации разработки, и улучшить ее на следующей итерации; – уменьшаются риски заказчика. Заказчик может с минимальными для себя финансовыми потерями завершить развитие неперспективного проекта.
Недостатки модели:	– увеличивается неопределенность у разработчика в перспективах развития проекта. Этот недостаток вытекает из предыдущего достоинства модели; – затруднены операции временного и ресурсного планирования всего проекта в целом.

Эволюционная модель (см. рис. 3) реализуется в виде последовательности блоков структур. Действия в этапах разработки в этой модели выполняется многократно, но в единой последовательности, что для общего блока структуры.

В следствие того что, промежуточные блоки структуры соответствуют реализации определенных целей, их реализация осуществляется на этапах сопровождения и эксплуатации, т.е. в параллели с процессом разработки блоков. Достоинства и недостатки эволюционной модели можно посмотреть в таблице 4.

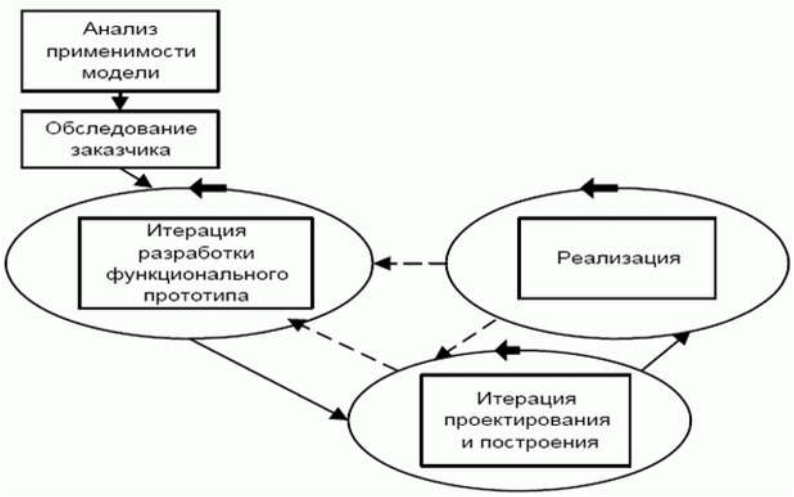

Рис. 3. Эволюционная модель жизненного цикла.

Таблица 3.

Достоинства и недостатки эволюционной модели жизненного цикла.

Критерий оценки:	Признаки:
Достоинства:	– быстрая реализация некоторых функциональных возможностей системы и их апробирование;

	–использование промежуточного продукта в следующем прототипе; –выделение отдельных функциональных частей для реализации их в виде прототипа; –возможность увеличения финансирования системы; –обратная связь устанавливается с заказчиком для уточнения функциональных требований; –упрощение внесения изменений в связи с заменой отдельной функции.
Недостатки:	–реализация всех функций системы одновременно может привести к громоздкости; –ограниченные человеческие ресурсы заняты разработкой в течение длительного времени.

Процессы жизненного цикла предназначены для: анализа и установления оптимальных параметров системных требований, решения задач по проектированию верхнего уровня системы. Обычно разработка жизненного цикла для любых программ начинается с определения основной идеи, далее проделывает путь через все этапы разработки, производства, эксплуатации и мониторинга программного продукта. При этом жизненный цикл постоянно редактируется в зависимости от назначения программы.

Жизненный цикл мобильного приложения по мнению Романа Белодеда (основателя компаний по разработке программного обеспечения для мобильных устройств e – Legion) особо не чем не отличается от каскадной модели жизненного цикла для программ персональных компьютеров. Но всё же есть определенные нюансы, разберем их подробней см. табл. 4.

Таблица 4.

Этапы жизненного цикла мобильного приложения.

Этапы жизненного цикла	Функций этапа
Установка цели	– определение основной аудитории пользователей; – исследование маркетинговых задач; – разработка концепции и путей привлечения аудитории.
Анализ требований	– определение требований к программе; – технического задания.
Проектирование интерфейса и строение программного кода	– разработка дизайна; – схема экранов; – черновых прототипов; – план использования программы; – условия сервера; – модель классов высшего уровня.

Утверждение дизайна интерфейса	– разработка модели высокой детализации; – испытание его пользователем.
Программирование и настройка функций	– клиент; – сервер; – безопасность; – тестирование на модульных тестах.
Тестирование	Исследования на тест – кейсах на соответствие первоначальным предпочтениям.
Опубликование в магазинах	Презентации мобильного приложения.
Техническая поддержка	Взаимодействие с пользователями в плане улучшения программы и фиксаций появившихся ошибок.
Развитие и обновление (новый релиз)	Подразумевает под собой повторение всех циклов.
Отказ от поддерживания	Утилизация продукта.

Прежде чем провести сравнение сред для разработки, нужно отметить что основным набором инструментом которым должна обладать платформа для программирования мобильных приложений, является Android SDK. В состав Android SDK входят такие виды инструментов как:

– SDK manager (загружает и устанавливает компоненты Android SDK);

– Debug Monitor (предназначен для отладки графического интерфейса);

– Android Emulator (инструмент для тестирования приложения непосредственно на компьютере);

– AVD manager (создает виртуальные Android устройства)

– Android Debug Bridge (инструмент для управления эмулятором)

В качестве первого примера рассмотрим официально рекомендуемую Google платформу Android Studio.

Android Studio

Android Studio основана на IntelliJ IDEA. Является официальной платформой для программирования Android приложений, доступна в бесплатном доступе. Обладает уже встроенным Android SDK.

Платформа Android Studio является обще признано самой удобной средой для тестирования и разработки приложений для Android. Компания Google сделала этот программный продукт с максимально полезным набором инструментов для разработки проектов под мобильные устройства. Процесс создания каждого приложения стал динамичней и проще, по сравнению с Eclipse. Это стало возможным, благодаря возможности отображения главных рабочих элементов в самой структуре будущего приложения, что позволяет более рационально подойти к разработке.

Пользователи отмечают такую полезную функцию, как просмотр в режиме реального времени всех дополнений. Так же среда позволяет разрабатывать приложения для разных версий Android.

Процесс работы на данной платформе значительно удобней Eclipse, благодаря доработке пользовательского интерфейса. В следствие чего написание кода стало более продумано, что позволяет легко ориентироваться при разработке больших по объёму проектов. Есть функция перетаскивания функциональных элементов в самой программе, что упрощает редактирование информации.

Главные функции Android Studio:

– присутствие справочника;
– наличие динамичного эмулятора для устройств на базе платформы Android;
– понятный интерфейс;
– отправка push-сообщений для приложений через любые облачные сервисы сразу на устройства под Android.
– возможность быстро локализовать приложения;
– есть опция маркировки кода;
– доступно большое число вариантов смены разрешения, размеров экрана;
– наличие инструментов для повышения качества проектов и монетизации;
– поддержка отслеживания эффективной работы рекламных объявлений;
– дружественное отношение с бета-тестерами;
– отображение всех действий (изменений) в проекте в режиме реального времени.

Инструменты Android Studio:

– Плагин Gradle для сборки приложений;
– Облачная среда Google;
– Функция ProGuard;
– Редактор WYSIWYG;
– Инструмент lint, создан для мониторинга проблем связанных с производительностью и совместимости версий;

– мастера основанные на шаблонах для разработки конструкций и компонентов Android.

– Google Cloud Messaging и App Engine сервисы могут быть интегрированы с помощью поддержки Google Cloud Platform.

Одним из основных преимуществ Android Studio является система сборки Gradle, которая интенсивно развивается компанией Google. Gradle обладает такими полезными функциями как:

– Создание различных вариантов сборки вашего приложения.

– Создание простых задач в виде скрипта.

– Возможность управления зависимостями и автоматически подгружать их.

– Настройка хранилища ключей.

Основные возможности среды «Android Studio» :

– Посредством пользовательского интерфейса можно перетаскивать компоненты;

– для ускорения разработки доступен многофункциональный редактор с различными инструментами;

– удобный плагин Gradle, позволяющая производить сборку автоматически;

– для проверки совместимости с различными платформами, а так же для анализа производительности возможно проведение тестирование;

– инструменты улучшения функций рекламы и управления монетизацией в приложениях;

– инструменты для обозначения и обработки кода;

– Google Cloud Messaging – push уведомления для ваших приложений посылающихся с сервера на мобильные устройства;

– рекординг видео с экрана, данная опция доступна только для Android 4.4.2 и выше;

– комфортная локализация приложений;

– для разработки кода доступны шаблоны и помощники;

– реорганизация кода;

Eclipse IDE

Платформа имеет стандартный набор для разработчика, но Eclipse различается от других IDE по нескольким основным аспектам. Данная среда разработки абсолютно нейтральна к платформе и языку программирования. Eclipse поддерживает языки: Cobol, Java, C++, C. Но в добавок к этому есть функция добавления интересующего вас языка, к примеру таких как: C#, PHP, Python, Ruby. Проекты по реализаций данных языков уже доступны на данный момент.

Среда Eclipse доступна, при помощи Eclipse Consortium под видом скомпилированного исполняемого файла для Windows, Linux и др..

Eclipse представляет из себя платформу, в которой разрабатываются плагины, далее встраиваемые в неё. Одним из таких является Android Development Tools (ADT). Плагин ADT намного расширяет возможности

данной среды разработки, с помощью него можно быстрей разрабатывать свои проекты под Android, создавать интерфейсы приложений, импортировать компоненты Android Framework API, отлаживать приложения, использовать Android SDK инструменты, а так же можно экспортировать подпись (без знака) APKs в порядке распространения своего приложения.

Плагин ADT включает в себя множество различных инструментов и несмотря на то что является дополнительным модулем, всё же обладает большим объёмом кода. Далее подробней разберем основные инструменты ADT:

– Редактор макетов Android. Макеты интерфейса в ADT создаются на языке XML. Среда предоставляет пользователю визуальный редактор для просмотра макетов. Когда вы открываете файл шаблона, плагин ADT автоматически запускает этот редактор для просмотра и редактирования файла. У данного инструмента удобный интерфейс предоставляющий удобное переключение между XML – редактором и визуальным редактором. На данный момент редактор макетов претерпел много изменений по сравнению с предыдущими версиями которые отличались более скромным функционалом и поэтому редко использовались. Теперь после его редактирования и дополнения редактирования макетов Android считается как основной метод работы. Для более корректной работы макетов на устройствах предусмотрена их авто – спецификация.

– Редактор описаний Android. Файл описания входит в состав проекта Android. Его роль заключается в том, что он информирует о том, как установить и использовать архивные программы, в котором состоит разработанный проект. В плагине ADT присутствует XML – редактор специально для изменения описаний. И это не единственный инструмент в котором можно изменять описания, так же это можно сделать в компоновщике приложений.

– Сборка приложении Android. Автоматизированная сборка в Eclipse позволяет объединять в готовый продукт исходный код и ресурсы проекта реализуя его к развёртыванию на устройстве, либо на эмуляторе. В ADT инструментом для выполнения таких операций является система Ant. В Android конечным результатом сборки проекта является файл APK.

– Запуск и отладка приложений Android. Инструментом для запуска и отладки в Eclipse является adb и DDMS позволяющие развёртывать проект на реальном или виртуальном устройстве. DDMS реализует обмен информаций с AVD, так же в нём учавствует среда времени исполнения Dalvik. DDMS

– Виртуальные устройства Android. QUME – подобные эмуляторы служат основой для виртуальных устройств в Eclipse, эмитирующие аппаратное обеспечение Andriod. Для конфигурирования виртуальных устройств Android используется диспетчер SDK и AVD, задающий такие параметры, как объем эмулируемых запоминающих устройств и параметры экрана. Кроме того, он позволяет указывать, какой образ системы Android будет использоваться с каким эмулируемым устройством.

– Виртуальные устройства Android обеспечивают тестирование программ в довольно широком диапазоне системных параметров. Для обеспечения такого широкого диапазона потребовалось бы достаточно большое количество реальных устройств, достать которые для тестирования может быть затруднительно. Поскольку QEMU – подобные эмуляторы оборудования являются универсальными в них можно тестировать устройства и образы систем которые пока не являются доступными.

– Диспетчер SDK и AVD. Android SDK управляет конфигурацией QEMU с помощью специального пользовательского интерфейса.

– Layoutopt – инструмент диагностирования проблем связанных с компоновкой элементов Andriod, написанных на языке XML.

– Monkey – это компонент для автоматизированного тестирования, работающий на эмуляторе или устройстве. В состав SDK входит система adb которая активирует Monkey.

– Keytool используется для создания временных отладочных ключей. Он генерирует ключи шифрования.

– Zipalign в готовых версиях приложений обеспечивает оптимизированный доступ к данным.

– Draw9patch – это специализированный инструмент для рисования состоящий в арсенале ADT.

Intel XDK

Инструмент для разработки кросс – платформенных приложений, так как используется язык HTML5 (сочетает в себе HTML язык разметки, CSS, JavaScript). Поддерживает все ступени разработки, то есть редактирования кода, функция эмулятора мобильного устройства, отладка, профилирование и публикация в магазине. Одной из полезных возможностей XDK является постройка приложения в облачном сервисе. В нём не нужно устанавливать дополнительные плагины, как Android SDK в Android studio или XCODE для IOS, просто код пересылается на сервер и там собирается автоматически. Так же XDK поддерживает все основные платформы мобильных устройств, что выгодно его выделяет перед нативными средами разработки.

XDK поддерживает такие игровые среды как: Cocos2d, Phaser, Pixi и EaselJS. С помощью XDK разрабатывать игры стало ещё удобней.

Данная среда содержит удобные инструменты для отладки, тестирования, сборки и анализа ваших приложений.

Intel Mobile Development Kit for Android

Специализированная среда разработки под Android от Intel. Обладает мощными инструментами для создания отличных приложений и игр, содержащие все основные компоненты платформы Intel System Studio. Поддерживает языки C, C ++, C #, Fortran, Java.

Благодаря мощным инструментам для отладки графической составляющей программного продукта, отлично подходит для разработчиков игр, которые в свою очередь положительно отзываются об инструментах MDK for Android. Рассмотрим, какими основными компонентами обладает данная среда:

– Intel VTune Amplifier. Инструмент для оценки системы, обладает расширенным анализом и настройкой производительности ЦП.

– Intel Energy Profiler. Средство разностороннего анализа энергопотребления и производительности для разработчиков системного ПО.

– Intel C++ Compiler. Передовой отраслевой компилятор C++ для совершенствования производительности высокооптимизированных систем Android и оригинального кода C++.

– Intel Integrated Performance Primitives. Обширная библиотека высокопроизводительных компонентов для создания программного кода, обработки сигналов, данных и мультимедиа.

– Анализатор видеосистем. Оптимизирует производительность видеосистем на базе GPU. Осуществляет анализ системной производительности в режиме реального времени.

– Анализатор платформ. Позволяет выполнять исчерпывающий автономный анализ разрабатываемого приложения.

– Анализатор кадров. Анализирует нагрузку видеосистем с получением подробных данных на уровне схем Open GL ES.

Intel Beacon Mountain

По сравнению с XDK данная среда ориентирована только на Android платформу. Имеет полезную функцию автоматического обновления, что позволяет пользователю не сосредотачиваться на поддержке актуальности своей платформы. Beacon Mountain разработана на базе Eclipse и оптимизирована рядом инструментов произведенных Intel:

– Intel* Threading Building Blocks (Intel* TBB) – очень популярная библиотека шаблонов C++;

– Intel* Integrated Performance Primitives (Intel* IPP) Preview – библиотека адаптированной обработки информаций и графики.

– Intel* Graphics Performance Analyzers (Intel* GPA) System Analyzer – позволяет производить мониторинг загруженности системы при работе с элементами OpenGL в реальном времени.

– Процессор ускоряющий работу эмулятора с помощью технологии Intel*VT.

XCode

Среда программирования для платформ IOS, OS X, WatchOS, tvOS разработанная компанией Apple. Поддерживаемые языки программирования: C, C++, Objective-C,Objective-C, Swift, Java, AppleScript, Python, Ruby.

Основные инструменты XCode:

– IOS Simulator. Инструмент для быстро просмотра разрабатываемого приложения.

– Dash. Это менеджер сниппетов и браузер документаций.

– TextExpander – это популярная утилита для управления отрывками кода, привязанных к пользовательским сочетаниям клавиш. Здесь даже представлены заглушки для большей кастомизации.

– TestFlight – отличный сервис для передачи тестовых билдов команде бета-тестеров. Этот сервис собирает отчеты об ошибках, пользовательские отзывы, и позволяет вам отслеживать пользовательские сессии.

– GDB. Отладчик кода.

«1С: Предприятие 8. Расширение для КПК».

Нельзя обойти вниманием и российского производителя программных продуктов компанию 1С, которая разработала платформу «1С:Предприятие 8. Расширение для карманных компьютеров». Среда предназначена для работы с базами данных «1С Предприятия 8» на мобильных гаджетах, а так же разработки приложений для них. Обладает хорошим инструментарием для разработки:

– Редактор мобильных приложений;
– Платформа для исполнений мобильных приложений;
– Сервер мобильных приложений;
– Компонента обмена данных.

Данная платформа имеет довольно узкую специфику по сравнению с западными аналогами, но при разработки проекта связанного с базами данных и бизнес логикой, является отличным решением.

Ниже приведена таблица 5 сравнения сред для разработки мобильных приложений, составленная на основе анализа выше перечисленной информации. Основными критериями, которой являются: многообразность языков программирования, удобство пользовательского интерфейса, разнообразие для разработки мобильных платформ, монетизация среды разработки.

Таблица 5.

Сравнение сред для разработки мобильных приложений.

Среда разработки	Язык программирования	Удобство интерфейса	Мобильные платформы для разработки	Плата
Android Studio	Java, C/C++, Delphi	да	Android	нет
Eclipse IDE	Java, C/C++, PHP, Ruby, Python, Cobol	да	Android, IOS, windows phone, symbian	нет
Intel XDK	HTML5	да	все	да
Intel Beacon Mountain	Java, C, C++	нет	Android	нет
«1С: Предприятие 8. Расширение для КПК»	Язык программирования 1С	да	Windows Mobile, Android	да

Intel Mobile Development Kit for Android	C, C ++, C #, Fortran, Java , ASM	нет	Android	да
XCode	C, C++, Objective-C, Objective-C, Swift, Java, AppleScript, Python, Ruby	да	IOS, OS X,	нет

Исходя из проведенного сравнения, более рентабельно использовать платформу Android Studio. В первую очередь, потому что у неё самый удобный пользовательский интерфейс, много доступного материала для обучения, вполне достаточный спектр языков программирования, бесплатность пользования и самый большой рейтинг целевой платформы в мире (согласно Kantar Worldpanel Comtech доля пользователей Android в мире составляет 68% на первый квартал 2016). Так же можно отметить, что эта среда постоянно развивается и совершенствуется благодаря компаний Google и за её актуальность можно не переживать.

ЛИТЕРАТУРА

1. Войт Н. Н. Информатика и вычислительная техника. – Ульяновск: УлГТУ 2013. 362 с.
2. Голощапов А. А. Google Android: программирование для мобильных устройств. – Спб.: БХВ – Петербург 2014. 163 с.
3. Пискунова Н. В. Заработать миллионы с Iphone и Android пользователей. – М.: Финансы и статистика 2015. 162с.
4. Соколов В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений. Учебное пособие. – М.: Юрайт 2016. 176 с.
5. Dale P., Morgan Hein Android для программистов. Создаем приложения. – Спб.: Питер 2012. 560 с.
6. Murata C. Империя приложений. – М.: Альпина Паблишер 2013. 236 с.
7. Nielsen J., Budiu R. Mobile Usability. – Спб.: Эксмо 2013. 256 с.
8. Stevens C. Миллионеры из AppStore. Секреты разработчиков приложений бесцеллеров. – М.: Манн, Иванов и Фербер 2012. 256 с.
9. Vale E. HTML5. Разработка приложений для мобильных устройств. – Спб.: Питер 2015. 225 с.
10. 1С Предприятие 8. Расширение для карманных компьютеров [Электронный ресурс] URL: http://v8.1c.ru/metod/books/files/KPK.pdf (дата обращения 07.04.2016)
11. Лекция 1: Мобильные устройства и их характеристики. Платформа Windows Mobile [Электронный ресурс] URL:

http://www.intuit.ru/studies/courses/574/430/lecture/9745 (дата обращения 13.03.2016)

12. Android Studio [Электронный ресурс] URL: http://developer.android.com/intl/ru/sdk/index.html (дата обращения 28.03.2016)

13. Intel MDK for Android [Электронный ресурс] URL: https://software.intel.com/en-us/android/intel-mobile-development-kit-for-android (дата обращения 25.02.2016)

14. Xcode [Электронный ресурс] URL: https://developer.apple.com/xcode/ (дата обращения 03.04.2016)