**МИНИСТЕРСТВО ОБРАЗОВАНИЯ**

**РЕСПУБЛИКИ БЕЛАРУСЬ**

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра вычислительной математики

Жаков Алексей Юрьевич

**Анализ инструментов разработки мобильных приложений с точки зрения оптимизации затрат на тестирование**

Курсовой проект

студента 3 курса 13 группы

“Допустить к защите” **Руководитель**

**Руководитель проекта** *Махлаева Татьяна Игоревна*

*Махлаева Татьяна Игоревна*

“ “ 2013 г.

**Минск 2013**

# АННОТАЦИЯ

Главной целью данного курсового проекта является изучение средств разработки мобильных приложений; анализ инструментов тестирования, предоставляемых вместе со средствами разработки, а также выяснение наиболее оптимальных из них с точки зрения временных затрат.

**АНАТАЦЫЯ**

Галоўнай мэтай дадзенага курсавога праекта з'яўляецца вывучэнне сродкаў распрацоўкі мабільных прыкладанняў; аналіз інструментаў тэставання, якія прадстаўляюцца разам са сродкамі распрацоўкі, а таксама высвятленне найбольш аптымальных з іх з пункту гледжання часавых і грашовых выдаткаў.

**ANNOTATION**

The main objective of this project is to study the means of mobile application development, and analysis of test tools provided along with development tools

as well as how best of them in terms of time costs.

**Анализ инструментов разработки мобильных приложений с точки зрения оптимизации затрат на тестирование**

Задачи исследования:

1.       Подготовка обзора и изучение наиболее популярных инструментов разработки мобильных приложений

2.       Подготовка обзора и изучение инструментов тестирования, предоставляемых указанными ранее инструментами разработки мобильных приложений.

3.       Подготовка тестовых примеров, моделирование мини-приложения или части кода с заведомо известными дефектами при помощи выбранных для сравнения инструментов тестирования

4.       Проведение экспериментального исследования временных затрат на обнаружение всех известных дефектов в тестовых примерах

# РЕФЕРАТ

*Курсовой проект,* 22 с., 8 рисунка, 3 таблицы, 5 источников.

*Ключевые слова*—IDE, ПЛАТФОРМА, ANDROID, ТЕСТИРОВАНИЕ

*Объект исследования* – инструменты разработки мобильных приложений.

*Методы исследования* – анализ наиболее популярных платформ, написание тестовых примеров.

*Цель работы*  изучить наиболее распространенные инструменты разработки мобильных приложений и проанализировать возможности тестирования, предоставляемые ими.

*Результаты работы—* проведен анализ наиболее популярных инструментов разработки и тестирования мобильных приложений, получены временные оценки использования инструментов тестирования, предоставляемых в Android SDK и Xamarin

# CОДЕРЖАНИЕ

[АННОТАЦИЯ 2](#_Toc378075634)

[РЕФЕРАТ 4](#_Toc378075635)

[CОДЕРЖАНИЕ 5](#_Toc378075636)

[ОСОБЕННОСТИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ 7](#_Toc378075637)

[ИНСТРУМЕНТЫ РАЗРАБОТКИ 9](#_Toc378075638)

[ТРЕБОВАНИЯ К ИНСТРУМЕНТАМ РАЗРАБОТКИ МОБИЛЬНОГО ПО 9](#_Toc378075639)

[ОСНОВНЫЕ ПЛАТФОРМЫ 10](#_Toc378075640)

[ИНСТРУМЕНТЫ ТЕСТИРОВАНИЯ 12](#_Toc378075641)

[ANDROID 12](#_Toc378075642)

[iOS 12](#_Toc378075643)

[КРОСПЛАТФОРМЕННЫЕ СРЕДСТВА 13](#_Toc378075644)

[РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРИМЕРОВ 14](#_Toc378075645)

[Первое приложение 14](#_Toc378075646)

[Второе приложение 16](#_Toc378075647)

[Создание аналогов приложений на других платформах 19](#_Toc378075648)

[ЗАКЛЮЧЕНИЕ 21](#_Toc378075649)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc378075650)

**ВВЕДЕНИЕ**

В последнее время наблюдается рост доли мобильных устройств и, соответственно, рост доли мобильных приложений. Рынок быстро развивается, поэтому большое значение приобретает скорость выпуска новых приложений и максимально быстрая окупаемость средств, затраченных на их разработку и тестирование.

В данной работе проведён анализ наиболее популярных инструментов разработки мобильных приложений и возможностей проведения тестирования, предоставляемых ими.

Большое значение имеет удобство и быстрота использования вышеупомянутых инструментов.

В работе будет проведено практическое исследование скорости разработки и тестирования мобильных приложений. В качестве примеров будут разобраны Android SDK и Xamarin.Android.

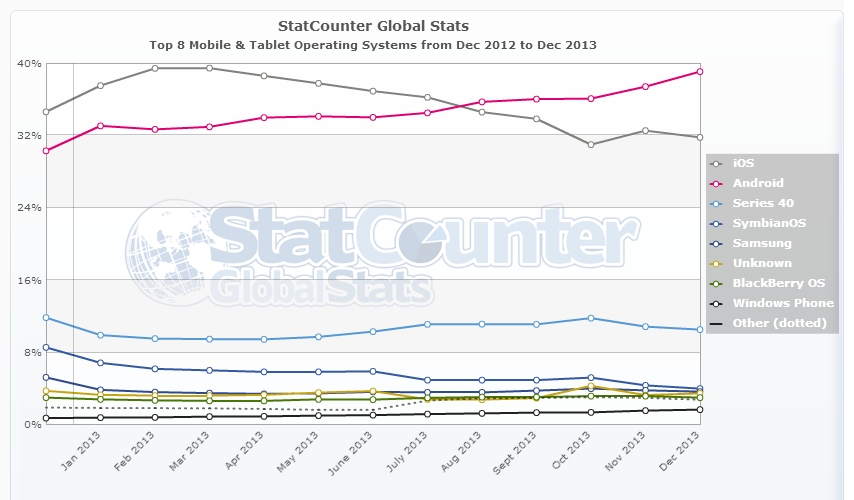
# ОСОБЕННОСТИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Разработка и тестирование мобильных устройств характеризуется большим количеством моделей устройств и их параметров. Следует обратить внимание на специфику использования мобильных устройств и приложений.

1. **Экран**. Современные устройства обладают экранами различных типов и размеров, а значит на разных типах экранов элементы будут иметь разные размеры и расположение.
2. **Память.** На мобильных устройствах зачастую приходится работать с малым объёмом памяти, как оперативной, так и постоянной. Появляется необходимость проверять правильность работы с SD картами и работу в условиях ограниченного объема памяти.
3. **Внешние события.** При работе приложения могут возникнуть условия, при которых может появиться необходимость изменения порядка работы приложения, например, изменение уровня заряда батареи, состояния сети, переход в режим ожидания и т д.

Выбор платформы и инструментов разработки и тестирования следует проводить исходя из текущего состояния рынка, который быстро меняется.

В качестве примера быстрого изменения рынка, рассмотрим систему Symbian.Symbian – операционная система, разработанная компанией Nokia. На данный момент проект закрыт. 5 лет назад эта ОС занимала 52.4% рынка, а на данный момент доля устройств с этой ОС составляет 4 %.[[1]](#footnote-1)



Так как ОС Android занимает основную долю рынка, в основном будут рассматриваться инструменты разработки именно для этой ОС.[[2]](#footnote-2)

# ИНСТРУМЕНТЫ РАЗРАБОТКИ

Для каждой мобильной ОС существует несколько платформ разработки, каждая из которых имеет интегрированную среду разработки. В составе IDE предоставляются инструменты, позволяющие разработчику создавать, тестировать и внедрять приложения на целевую мобильную платформу.

## ТРЕБОВАНИЯ К ИНСТРУМЕНТАМ РАЗРАБОТКИ МОБИЛЬНОГО ПО

Во многом, результативность создания мобильных приложений зависит именно от инструментов, используемых разработчиками и тестировщиками.

Эффективность среды разработки мобильных приложений, как, в общем, и остальных, определяется полнотой решения ей следующих задач:

**Интеграция:**  платформа разработки мобильных приложений должна предоставлять возможности для коллективного управления продуктами на всем протяжении их жизненного цикла.

**Гибкая разработка:** множество коротких итераций позволит постепенно реализовать запланированный план разработки.

**Общий доступ к обновлениям:** наборы изменений должны быть доступны всем членам группы разработчиков и тестировщиков, что необходимо для устранения конфликтов в процессе окончательной интеграции этих наборов с основным кодом.

Приведённые выше задачи критичны при работе над проектом в составе команды. В случае, когда над приложением работает один человек, оценка инструментов производится по другим параметрам:

**Степень контроля над ресурсами:** разработчик должен иметь быстрый доступ ко всем ресурсам, включённым в проект.

**Интеграция дополнительных инструментов:** IDE должна быть способна поддерживать жизненный цикл приложения (разработка – тестирование – исправление дефектов). В среде разработки должна быть возможность добавления к ней модулей, упрощающих отладку, тестирование, исправление дефектов.

## ОСНОВНЫЕ ПЛАТФОРМЫ

**Android**. Разработка приложений для этой ОС может вестись на различных языках, но в основном это Java. Официальный SDK позволяет вести разработку на Java с использованием C/C++ вставок, однако существуют платформы, позволяющие вести разработку на JavaScript/HTML5/CSS, C#, Visual Basic.

**Android SDK**. Официальный SDK, выпущенный Google. Является бесплатной платформой, разработка ведётся на языке Java с возможностью добавления участков на C/C++, используя Android NDK(Native Development Tools). В составе данной платформы предоставляется эмулятор устройств.

Существует следующие IDE:

* Eclipse
* IntelliJ IDEA
* Kenai – плагин для NetBeans
* Android Studio (режим раннего доступа)[[3]](#footnote-3)
* AIDE

**iOS.** Разработка приложений для этой ОС ведётся на языках Objective-C, Object Pascal. Существуют платформы, позволяющие вести разработку на JavaScript/HTML5/CSS, C#.

**iOS SDK** Официальный SDK, выпущенный компанией Apple. Распространяется бесплатно, однако для запуска приложений на реальном устройстве требуется приобретение лицензии разработчика стоимостью 99$.

Основные IDE:

* Xcode. Данная среда разработки используется при написании мобильных приложений на Objective-C. В ней присутствует отладчик и симулятор iPhone. Данная IDE работает только под управлением OS X.
* Embarcadero Delphi. IDE, работающая под управлением Windows, Mac OS. Позволяет вести разработку приложений для iOS на языке Object Pascal под управлением Windows, однако компиляция и внедрение должны происходить на Mac. [[4]](#footnote-4)

**Кроссплатформенные решения.**

1. **Xamarin**. Является платным фреймворком. Позволяет вести разработку Android и iOS приложений на языке C#. В качестве IDE может использоваться Visual Studio или Xamarin Studio, в каждой из которых присутствует возможность использования эмулятора устройств Android. Для разработки интерфейса и отладки программ для iOS потребуется Xcode. Как уже сказано выше, решение платное, цена варьируется от 299$ до 1899$. Существует бесплатная версия, которая накладывает ограничение на размер IL кода в 32 килобайта.[[5]](#footnote-5)
2. **PhoneGap.** JavaScript/HTML5/CSS. Приложения, написанные с использованием данного инструмента, являются, по сути, простыми браузерными приложениями, для написания котоых не требуется знание “родных” языков программирования.Решение является бесплатным. PhoneGap также может применяться для создания приложений на BlackBerry, Windows Phone и других платформах.

# ИНСТРУМЕНТЫ ТЕСТИРОВАНИЯ

## ANDROID

В основном, приложения для Android пишутся на Java.Для проведения модульного тестирования, не связанного с Android API используется фреймворк JUnit. Для проведения тестирования, связанного с Android API и UI тестирования используются следующие инструменты:

- **android.test**. Фреймворк, позволяющий тестировать UI и связанные с ним классы. Предоставляется в составе Android SDK. Позволяет выполнять тестовые сценарии на эмуляторе и реальном устройстве. Данный фреймворк базируется на JUnit.

- **UIAutomator**. Фреймворк для проведения автоматического тестирования UI. Также предоставляется в составе Android SDK. Основной особенностью данного фреймворка – возможность автоматического распознавания Android UI элементов. Присутствует возможность запуска сценариев сразу на нескольких тестовых устройствах.

В SDK имеется эмулятор устройств Android. Скорость его работы весьма низкая, поэтому имеет смысл использовать сторонние реализации, а именно:

-Android-x86

-AndroVM

-Genymotion

-Manymo.

## iOS

Приложения для iOS в основном пишутся на Objective C. Средством unit тестирования, предоставляемым в составе SDK, является SenTestingKit – фреймворк, включенный в состав Xcode.

**- UIAutomation** – решение от Apple, которое позволяет выполнять написанные на JavaScript тестовые сценарии. Позволяет проводить функциональное тестирование. Решение бесплатное, предоставляется в составе SDK.

В составе SDK предоставляется также симулятор iPhone, имитирующий внешний вид устройства. Симулятор iPhone является слабым инструментом тестирования, поэтому Apple рекомендует тестировать мобильные приложения на реальных устройствах.

## КРОСПЛАТФОРМЕННЫЕ СРЕДСТВА

**Xamarin**. Для написания приложений под этой платформой используется C#. Фреймворк для Unit тестирования модулей без API целевых платформ – NUnit. Тестирование UI и связанных с ним классов должно проводиться с помощью инструментов целевых платформ.

**PhoneGap.** Для проведения тестирования приложений для этой платформы достаточно наличия браузера и отладчика JavaScript, так как приложения, разработанные на данной платформе являются браузерными и не могут использовать нативные возможности устройства.

# РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРИМЕРОВ

В качестве платформы для написания тестовых примеров я выбрал Android, т.к. у меня есть устройство для проведения ручного тестирования и в дальнейшем мне бы хотелось заняться разработкой приложений именно для этой платформы.

## Первое приложение

Сумма двух чисел

Приложение Test1 позволяет быстро сложить два положительных рациональных числа. Вы просто вводите два числа, нажимаете кнопку и получаете результат.

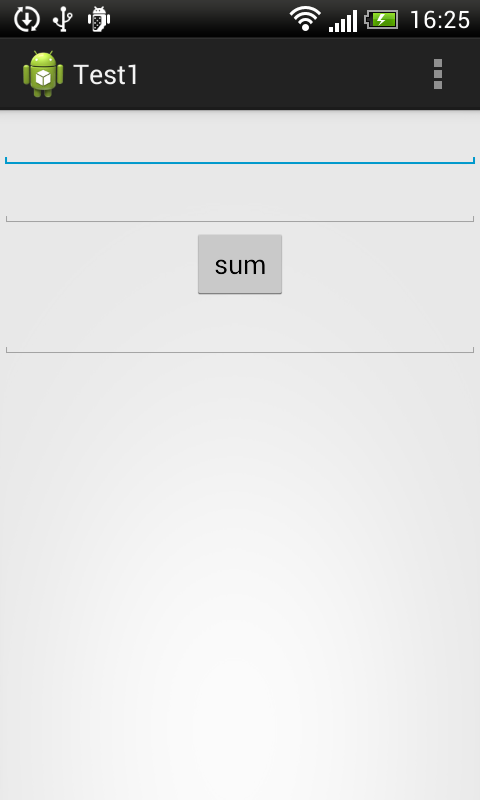
Данное приложение представляет собой одну Activity с расположенными на ней четырьмя элементами пользовательского интерфейса:

- 2 элемента EditText для ввода данных

- Button с надписью sum

- EditText для выходных данных

При нажатии на кнопку sum в третьем текстовом поле должна отображаться сумма значений двух верхних полей.

. 

Внешний вид приложения

**Для проведения тестирования использовались следующие инструменты**:

- эмулятор устройств android, входящий в состав SDK

- android.test для проведения автоматического тестирования

- мобильное устройство, работающее под управлением Android 4.1

**Намеренно были внесены следующие дефекты**:

1)Headline: вылет приложения при вводе пустого поля

Description:

- запустить приложение

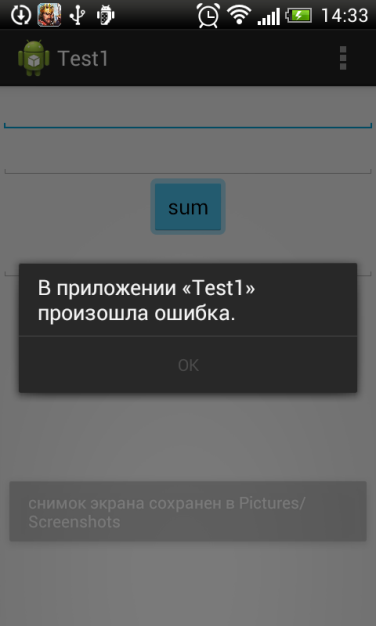
- нажать кнопку sum

2)Headline: возможность редактирования поля вывода

Description:

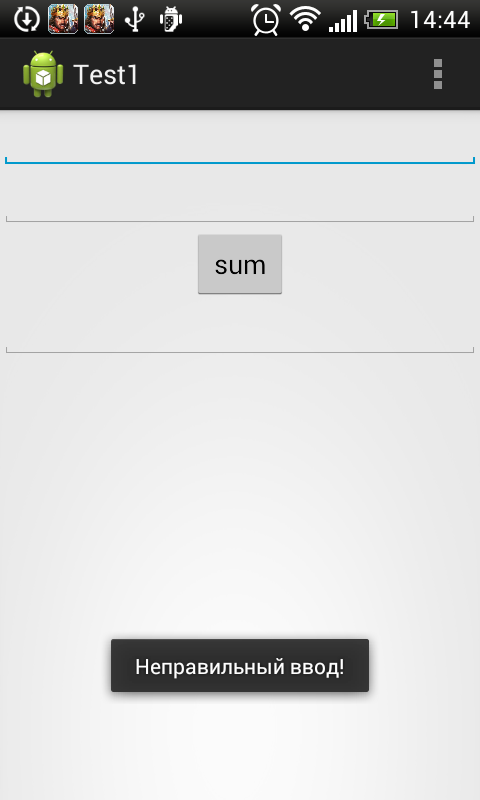
- запустить приложение

- нажать на поле вывода



Дефект 1

Было написано 6 автоматических тестов для проверки отображения элементов, корректности ввода, корректности запуска приложений, проверки начальных значений полей, функциональности. Все дефекты были обнаружены при проведении ручного и автоматического тестирования и впоследствии исправлены.



Дефект 1 исправлен!

Ниже приведена таблица временных затрат на разработку и тестирование первого приложения.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Тесты на реальном устройстве | Тесты на “холодном” эмуляторе | Тесты на готовом эмуляторе | Написание примера | Написание тестов |
| Test1 | **7 секунд** | **2 минуты** | **30 секунд** | **20 минут** | **30 минут** |

## Второе приложение

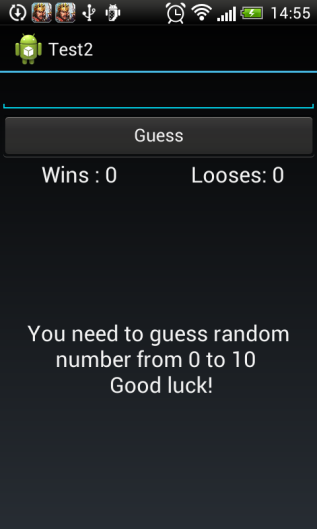
Игра “Угадай число”

Приложение Test2 представляет собой игру, в которой пользователю предлагается угадать число от 0 до 9, которое случайным образом генерирует устройство. На протяжении всей игры также ведётся статистика, так что вы можете проверить свою удачу и равномерность распределения случайных величин генерируемых вашим устройством.

Для игры просто введите число и нажмите кнопку “Guess”. В нижней части экрана будет отображаться статистика текущего раунда. Под кнопкой Guess будут отображаться ваши победы и проигрыши.

Приложение представлено одной Activity с расположенными на ней пятью элементами UI:

* поле ввода числа.
* кнопка подтверждения ввода
* два текстовых поля, на которых отображаются количество побед и проигрышей.
* Текстовое поле результата текущего раунда



Внешний вид приложения

**Для проведения тестирования использовались инструменты**:

- мобильное устройство, работающее под управлением Android 4.1

- эмулятор устройств android, входящий в состав SDK

**Намеренно были внесены следующие дефекты**:

1)Headline: вылет приложения при вводе пустого поля

Steps to reproduce:

- запустить приложение

- нажать кнопку Guess

Headline: изменение формата текстовых блоков “Wins/Looses”

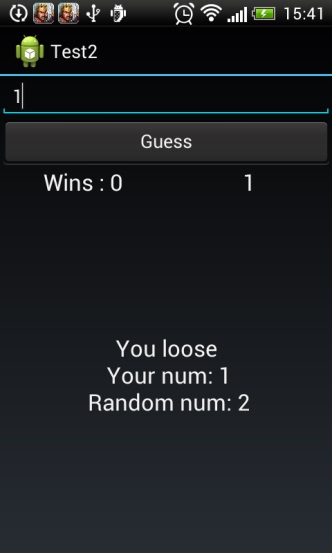
Steps to reproduce:

- запустить приложение

- ввести число в поле ввода

- нажать кнопку Guess

Expected result: “Wins(Looses) : x”



Дефект 2

Данное приложение было написано на платформе Xamarin на языка C#. Автоматические тесты провести не удалось, из-за отсутствия такой возможности у данной платформы. Unit тесты не проводились из-за цельности приложения и связанности функционала с UI элементами. Приложение тестировалось вручную и при помощи эмулятора. Все дефекты были найдены и исправлены в короткий срок: не более 10 минут, однако эмулятор работал гораздо медленнее и сильно замедлял процесс тестирования.

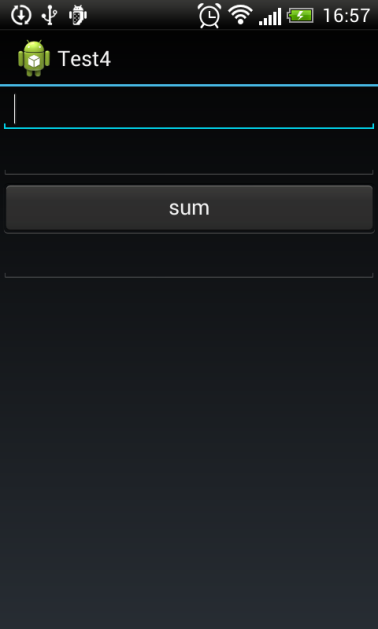
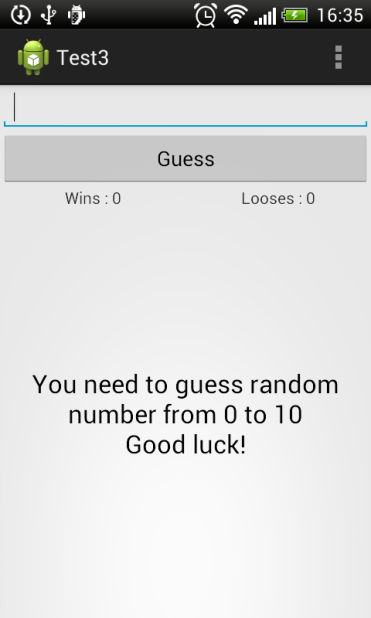
Отсутствие возможности проведения автоматического UI тестирования на данной платформе является серьёзным минусом. Также у данной платформы **ниже показатели скорости развертывания** приложения на устройстве. Если при использовании Android SDK на установку приложения тратится несколько секунд, то при использовании Xamarin развёртывание приложения может занимать до двух минут. Связано это с проведением установки дополнительных библиотек, таких как Mono Shared Runtime, Mono Android Support.

Ниже приведена таблица временных затрат на разработку и тестирование первого приложения.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Проведение тестирования на устройстве | Проведение тестирования на эмуляторе | Написание примера |
| Test2 | 6 минут | 7 минут | 40 минут |

## Создание аналогов приложений на других платформах

В качестве приложений 3 и 4 рассматриваются аналоги уже написанных приложений. Для приложения 1 использовался Xamarin, для приложения 2 – Android SDK. Программы, их интерфейс, дефекты получились идентичными, была лишь добавлена разница в оформлении.



Приложение 3 Приложение 4

Для данных приложений проводились те же тесты, что и для приложений 1, 2.

Ниже приведена таблица временных затрат на разработку и тестирование приложений.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test1  (Sum, Android) | Test2  (Guess, Xamarin) | Test3  (Guess, Android) | Test4  (Sum, Xamarin) |
| Разработка | 10 минут | 25 минут | 20 минут | 10 минут |
| Автоматическое тестирование на эмуляторе | 2 минуты 30 секунд | - | 2 минуты 30 секунд | - |
| Автоматическое тестирование на устройстве | 7 секунд | - | 7 секунд | - |
| Ручное тестирование на устройстве | 3 минуты | 6 минут | 5 минут | 6 минут |
| Ручное тестирование на эмуляторе | 5 минут | 7 минут | 5 минут | 7 минут |

В строках, связанных с автоматическим тестированием, стоит также учитывать время, потраченное на написание тестов. В среднем это составило 25 минут.

Вывод по полученным данным:

1. Проведение автоматических тестов на реальном устройстве даёт большой выигрыш во времени по сравнению с эмулятором.
2. Ручное тестирование приложений на Android занимает меньше времени, так как при развёртывании Xamarin приложения на устройство или эмулятор устанавливаются дополнительные библиотеки.
3. На разработку Xamarin и Android приложений требуется примерно одинаковое время.

# ЗАКЛЮЧЕНИЕ

В результате проделанной работы были изучены платформы разработки мобильных приложений. На основе статистических данных были выбраны основные платформы и проведён анализ инструментов тестирования, предоставляемых ими.

Для платформ Android и Xamarin были написаны тестовые приложения и проведён анализ временных затрат на их разработку и проведение тестирования, как ручного так и автоматического.

Общие выводы:

1. Xamarin не предоставляет инструменты для автоматического тестирования. На длительных больших проектах и проектах со стабильной функциональностью отсутствие возможности проведения автоматического тестирования может стать причиной неоправданного завышения временных и денежных затрат на тестирование.
2. С помощью Android SDK быстрее проводится ручное тестирование, так как Xamarin устанавливает на устройство дополнительные библиотеки.
3. Наличие возможности писать кроссплатформенные приложения у Xamarin сильно замедляет скорость развёртывания на устройстве, что при примерно одинаковых временных затратах на разработку делает более предпочтительным использование Android SDK.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. [http://gs.statcounter.com/#mobile+tablet-os-ww-monthly-201212-201312](http://gs.statcounter.com/" \l "mobile+tablet-os-ww-monthly-201212-201312)
2. <http://ru.wikipedia.org/wiki/Symbian_OS>

1. <http://developer.android.com/sdk/installing/studio.html>

1. [http://xamarin.com/](http://xamarin.com/android)

1. <http://en.wikipedia.org/wiki/Mobile_application_development>

1. Источник <http://ru.wikipedia.org/wiki/Symbian_OS> [↑](#footnote-ref-1)
2. Источник [http://gs.statcounter.com/#mobile+tablet-os-ww-monthly-201212-201312](http://gs.statcounter.com/" \l "mobile+tablet-os-ww-monthly-201212-201312) [↑](#footnote-ref-2)
3. Источник <http://developer.android.com/sdk/installing/studio.html> [↑](#footnote-ref-3)
4. Источник <http://en.wikipedia.org/wiki/Mobile_application_development> [↑](#footnote-ref-4)
5. Источник http://xamarin.com/ [↑](#footnote-ref-5)