

УДК

**ПРОГРАММНАЯ СИСТЕМА АВТОМАТИЗИРОВАННОГО  
ИМИТАЦИОННОГО ТЕСТИРОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ.  
МОБИЛЬНОЕ ПРИЛОЖЕНИЕ НА ПЛАТФОРМЕ ANDROID  
SOFTWARE SYSTEM OF AUTOMATED IMITATION TESTING OF  
MOBILE APPLICATIONS. ANDROID MOBILE APPLICATION**

В.А.ЧЕРЕЗОВ – студент бакалавриата, Институт информационных технологий и радиоэлектроники, кафедра ИСПИ, группа ИСТ-116, E-mail: wapost@mail.ru

В.В.ВЕРШИНИН – научный руководитель, к.т.н., Институт информационных технологий и радиоэлектроники, кафедра ИСПИ

V.A. CHEREZOV– bachelor student, Vladimir state university, E-mail: wapost@mail.ru

V.V. VERSHININ– candidate of technical sciences, Vladimir state university

**Аннотация:** Описаны процессы автоматизации разрабатываемой системой для имитационного тестирования. Рассмотрена концепция системы.

**Abstracts:** Automation processes by the developed system for simulation testing are described. The concept of the system is considered.

**Ключевые слова:** мобильное приложение, имитационное тестирование, концепция, автоматизация

**Keywords:** mobile application, simulation testing, concept, automatization

### **Введение**

Название проектируемой системы – «Программная система имитационного моделирования на платформе Android».

Система предназначена для автоматизации ручного тестирования мобильных приложений и улучшения качества работы с нестабильными дефектами. Целевая платформа — Android. Разрабатываемая система будет состоять из двух компонентов: рекордер для мобильных приложений и сервер.

Предполагается, что все разработанные тесты на мобильном устройстве могут быть синхронизированы с данными на сервере. Также стоит отметить, что одним из дополнительных уровней автоматизации предполагается обеспечение доставки тестов на необходимые мобильные устройства, которые были ранее зарегистрированы в системе. Важной составляющей также являются автоматизированный сбор журналов и статистики.

### **Определение цели разработки**

Главной целью создания ИС являются *повышение качества* создаваемых продуктов и *оптимизация* сопутствующих процессов.

Целью оптимизации является улучшение значения следующих показателей:

- 1) время проведения тестирования;
- 2) количество проводимых итераций тестирования;
- 3) время, затрачиваемое на перенос логов с устройства;
- 4) качество тестирования и его статистики.

Концепция системы предполагает использование продукта в DevOps методологии. Автоматизированные тесты удобным способом можно запускать на стадии проверки кода, что обеспечит тестирование почти на каждую мелкую функциональность или правку дефектов. Продукт отлично подойдет для организации регрессионного тестирования и для работы с нестабильными дефектами.

### **Определение решаемых задач**

В рамках проекта автоматизируется множество действий, поэтому в таблице 1 представлен список задач, которые могут быть решены с помощью разработки.

Таблица 1 — Список решаемых задач

Функциональность	Описание
Создание и обработка	Наличие в системе статистики о пройденных

статистики	тестах, возможности её сбора и анализа в автоматизированном режиме.
Удалённое хранение тестов и синхронизация устройства	Наличие в системе возможности сохранить тест для использования на другом устройстве. Также наличие синхронизации устройств, на которых тест был запущен.
Доставка тестов на устройства	После создания теста и отправки его на сервер важно отправить тест на устройства, которые были ранее зарегистрированы для тестирования конкретного приложения.
Сбор журналов с устройств	Автоматизированный сбор журналов приложения. Очень важная функциональность, так как при прохождении теста приложение может возникнуть критическая ошибка, и приложение завершит работу.
Возможность пользоваться рекордером для записи тестов	В системе предусмотрено мобильное приложение, с помощью которого можно произвести запись последовательности для создания теста.
Поддержка авторизации в системе	Продукт будет доступен широкому кругу специалистов как внутри компании клиента, так и за её пределами. Соответственно, системе необходимо обеспечивать должный уровень безопасности.

Если говорить о внедрении системы в работу специалистов, то необходимо отметить, что, если ведётся разработка по гибкой методологии, то проблем не будет. В методологиях данного типа есть возможность заложить необходимые для внедрения ресурсы без ущерба общему процессу

тестирования и разработки. Многие технологические компании перешли к использованию гибких методологий в разработке, поэтому продукт должен помочь широкому кругу компаний. На рисунке 1 можно найти схему работы в спринте(выполнено в нотации BPMN).

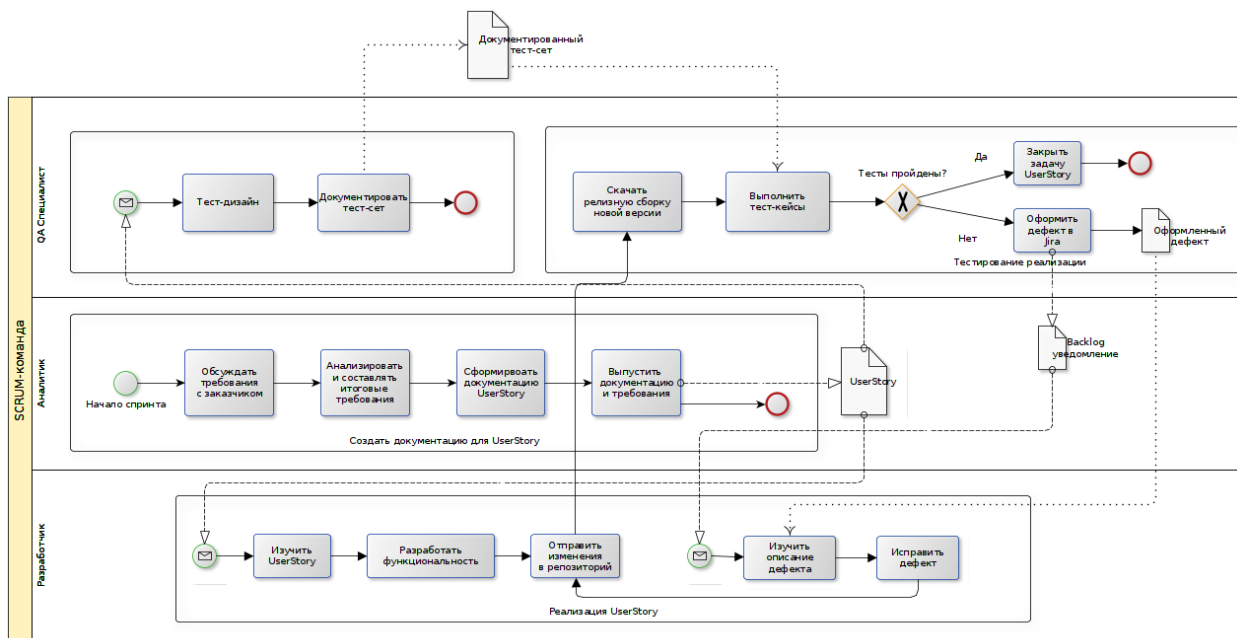


Рисунок 1 — Схема работы в спринте

### Расчёт снижения трудоёмкости

Для наиболее наглядного отображения изложенной информации был проведён анализ уменьшения трудоёмкости. Рассматривается случай тестирования реализации. Данный процесс возникает на каждом спринте при работе по гибкой методологией. Специалист по тестированию составляет тест один раз и в дальнейшем сможет запускать его неограниченное количество, а при коллаборации со средствами DevOps инструментов появляется возможность и вовсе исключить ручное тестирование. По форму 1.1 можно рассчитать уменьшение трудоёмкости:

$$p = \frac{T_{до} - T_{после}}{T_{до}}, \quad (1.1)$$

где  $T_{до}$  — трудоёмкость до автоматизации,

$T_{после}$  — трудоёмкость после автоматизации.

На рисунке 2 представлен результат составления таблицы для анализа. Расчёты произведены в открытой среде LibreOffice Calc. Рассматривается блок «Тестирование реализации» из рисунка 1.

Задача	Трудоемкость до автоматизации (мин.)	Трудоемкость после автоматизации (мин.)	Уменьшение трудоемкости
Создание тестового сценария	20	10	0.50
Создание теста	10	3	0.70
Проведение тестирования	30	5	0.83
Ведение статистики	15	5	0.67
Анализ	12	3	0.75
Сбор логов	10	0	1.00
Передача тестов на другие устройства	10	2	0.80

Рисунок 2 — Уменьшение трудоёмкости

Из рисунка можно заметить, что сокращение трудоёмкости представлено по всем видам деятельности специалиста по обеспечению качества ПО. Данный фактор положительно скажется на производительность проводимого тестирования и позволит повысить скорость работы одного сотрудника относительно прежних показателей.

### **Макет приложения**

Также был разработан концептуальный макет дизайна системы. Он очень просто. Мобильная часть системы представлена как библиотека к основному приложению, в котором требуется тестирование. Библиотека добавит экраны записи и списка тестовых прогонов. На рисунке 3 можно увидеть получившийся макет.

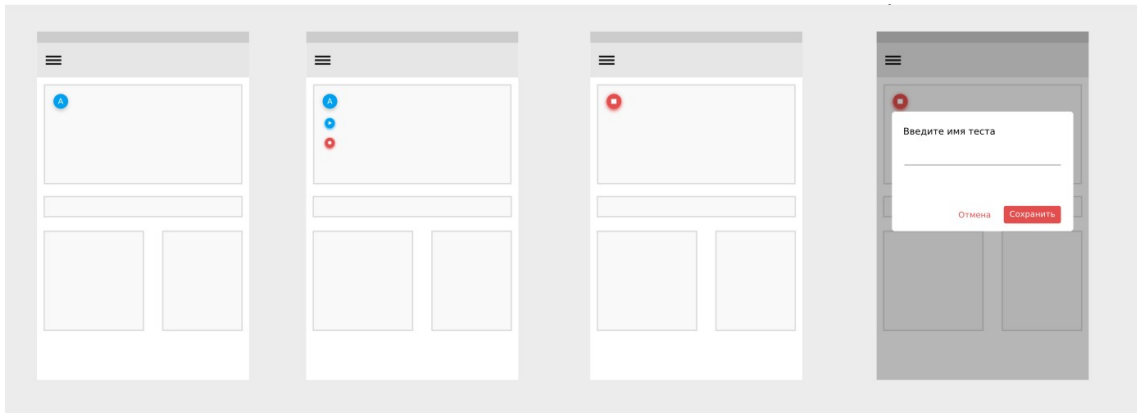


Рисунок 3 — Базовый макет

### Взаимодействие с системой и её компонентов

Для наглядности происходящих процессов можно обратить внимание на рисунки 4 и 5. На них отображены схемы последовательности для сценариев «Создание теста» и «Запуск теста». Следует отметить, что «AUUI» - название системы на этапе разработки.

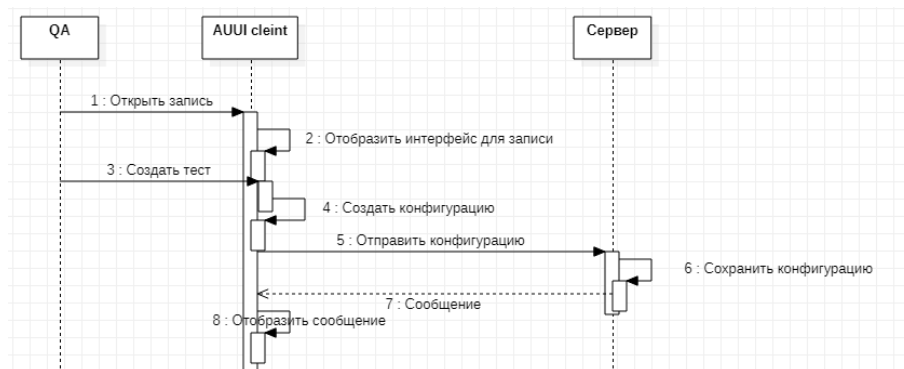


Рисунок 4 — Создание теста

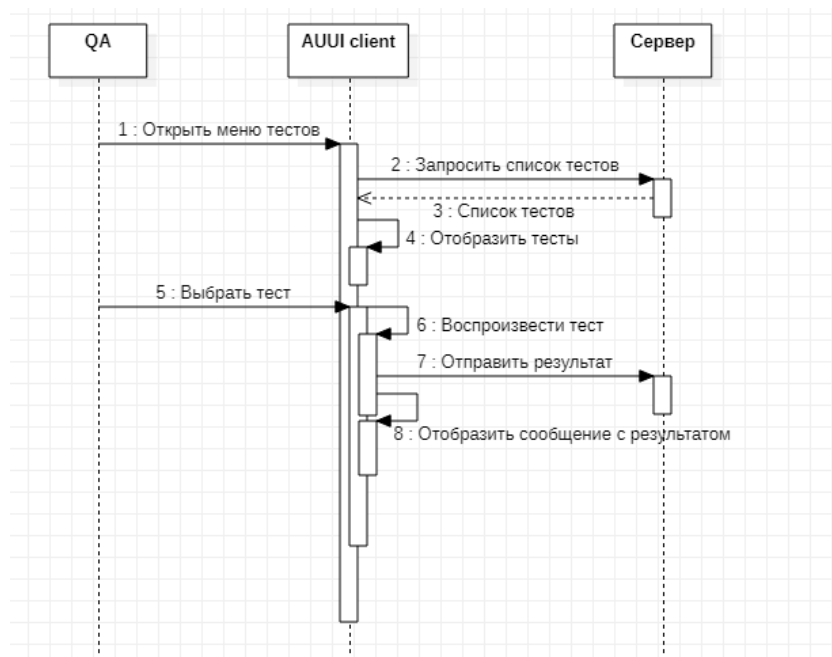


Рисунок 5 — Запуск теста

### Вывод

В данной работе рассмотрены основные моменты разработки программной системы автоматизированного имитационного тестирования мобильных приложений на платформе Android в рамках мобильного приложения.

### Список используемой литературы:

1. The Architecture of Open Source Applications – [электронный ресурс] - <http://www.aosabook.org/en/index.html>
2. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – Питер, 2019. — 395 с.
3. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. — Минск: Четыре четверти, 2017. — 312 с.
4. Проектирование информационных систем: Методические указания к практическим занятиям/ сост. Р.И. Макаров, В.И. Мазанова – Владим. гос. ун-т. Владимир, 2008. 152 с.