МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

Институт
информационных систем
и технологий

Кафедра
информационных систем

КУРСОВОЙ ПРОЕКТ

по дисциплине «**Проектирование информационных систем**»
на тему: «**Проектирование информационной системы поддержки разработки и обновления приложений для мобильных устройств**»

Направление **09.03.02 Информационные системы и технологии**

Руководитель,
ст. преподаватель

Овчинников П.Е.

«___» _____ 2018 г.

Студент,
группа ИДБ–15-14

Материкин В.В.

«___» _____ 2018 г.

Москва 2018 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящий момент разработка мобильных приложений является одним из приоритетных направлений деятельности многих компаний по разработке программного обеспечения, и, как и в других сферах разработки ПО, одной из важнейших задач становится автоматизация процесса разработки и обновления мобильного приложения (DevOps).

Для решения поставленной задачи необходимо разработать систему, позволяющую максимально автоматизировать выполнение таких процессов, как:

- Сборка приложения.
- Тестирование;
- Распространение.

Таким образом, объектом исследования в данной работе является процесс функционирования системы для автоматизации процесса DevOps мобильного приложения. Целью исследования является выделение автоматизируемых процессов, эффективность которых можно улучшить с помощью внедрения разрабатываемой системы.

Исследование выполняется путем построения следующих видов моделей:

- Функциональной (IDEF0).
- Потоков данных (DFD).
- Диаграммы классов (ERD).

Функциональная модель рассматривается с точки зрения руководителя отдела мобильной разработки, который непосредственно руководит разработкой приложения и имеет цель повысить эффективность работы своей команды. Для моделирования будет использоваться программная среда Ramus.

ГЛАВА 1. ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ (IDEF0)

Функциональная модель IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов [1].

Функциональная модель IDEF0 представляет собой набор блоков, каждый из которых представляет собой «черный ящик» со входами и выходами, управлением и механизмами, которые детализируются (декомпозируются) до необходимого уровня. Функциональные блоки соединяются между собой при помощи стрелок и описаний. При этом каждый вид стрелки или активности имеет собственное значение [2].

Таким образом, в модели IDEF0 все данные можно разделить на 4 различных типа, а именно:

- Внешние входные информационные потоки.
- Внешние выходные информационные потоки.
- Внешние управляющие потоки.
- Механизмы.

Внешними входными информационными потоками системы для автоматизации процесса DevOps являются:

- Исходный код.
- Описание конфигурации.

Внешними выходными информационными потоками системы являются:

- Установочный пакет приложения.

Внешними управляющими потоками процесса являются:

- Бизнес-требования.

Основными механизмами процесса являются:

- DevOps инженер.
- Команда разработчиков.
- Система контроля версий Git.
- SDK для разработки.

- Внешние сервисы.

На рисунках 1-3 представлены диаграммы IDEF0 с декомпозицией блоков A0, A2.

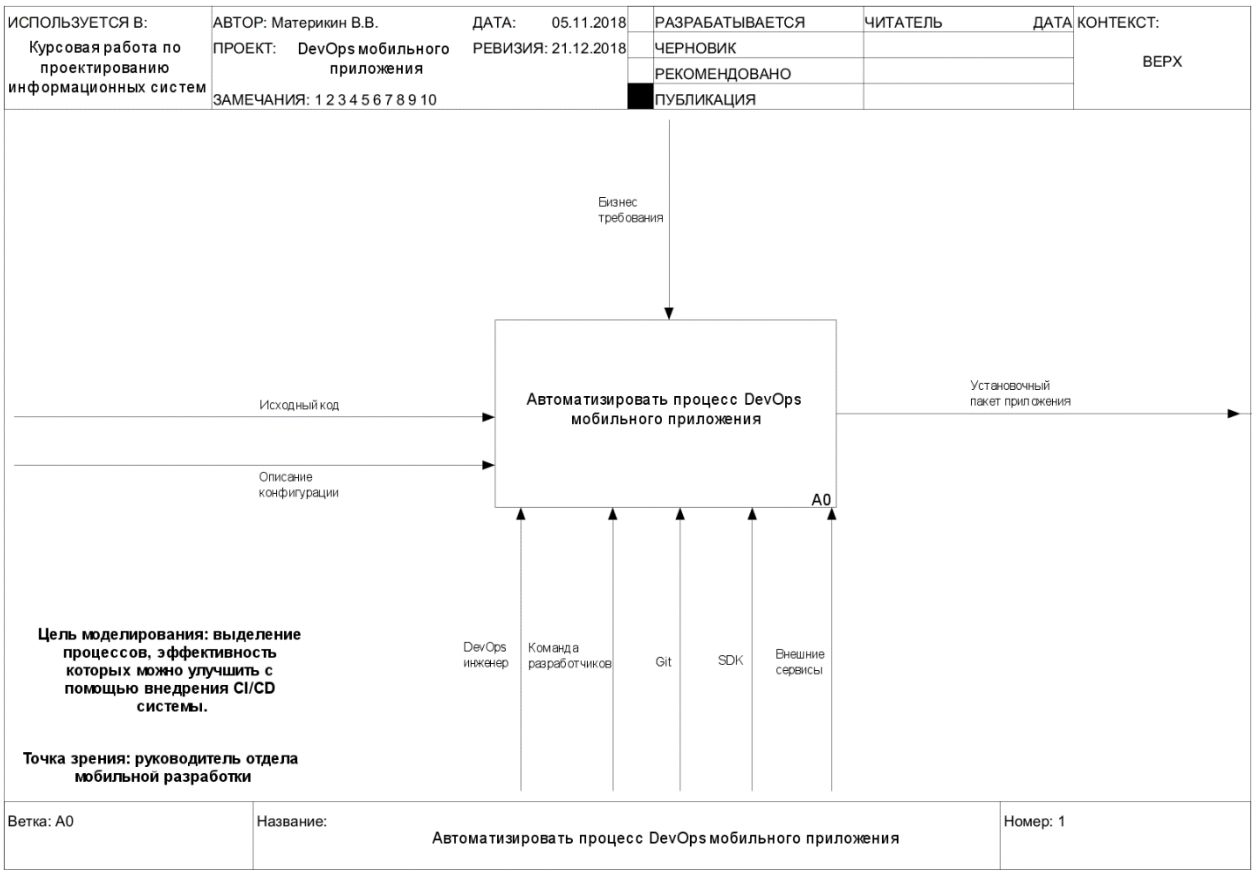

Рис. 1. Контекстная диаграмма

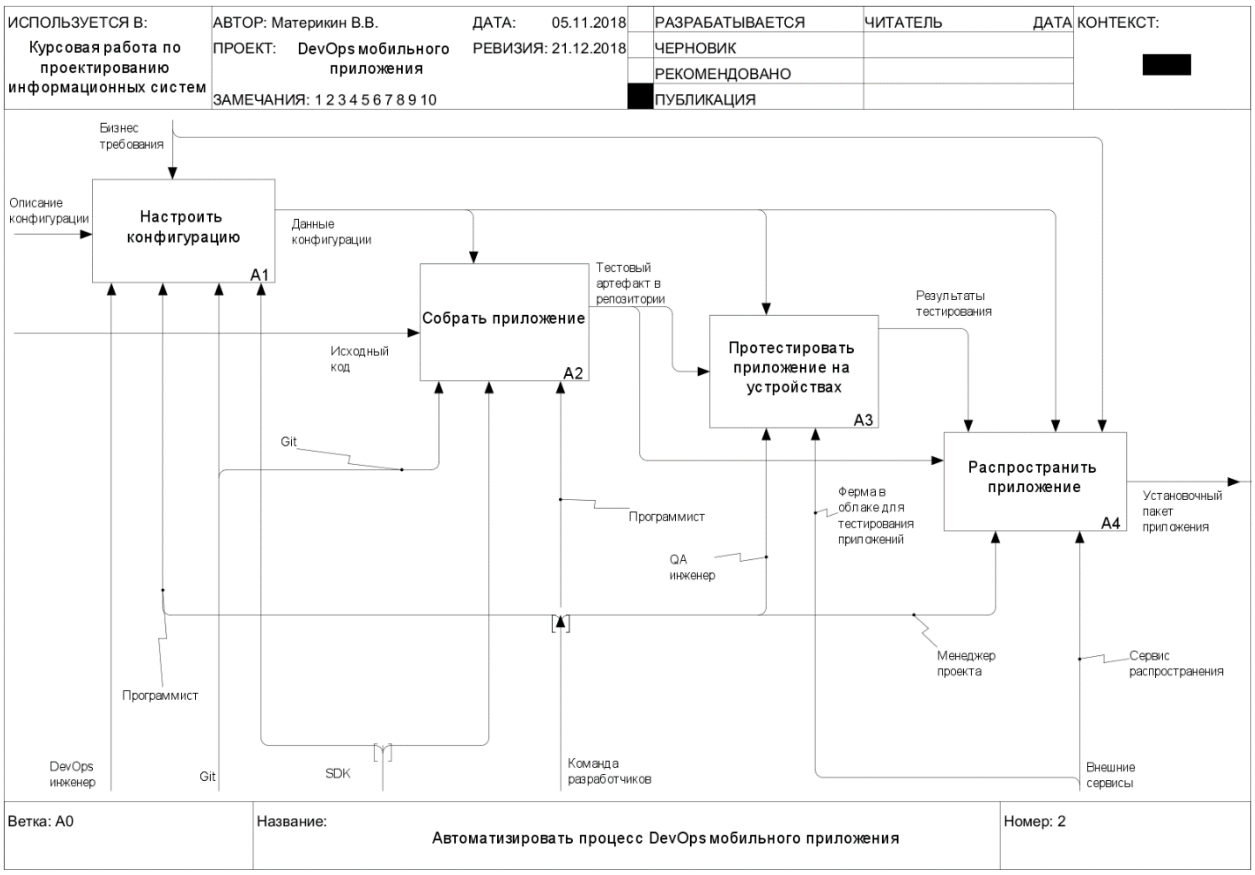

Рис. 2. Декомпозиция процесса A0

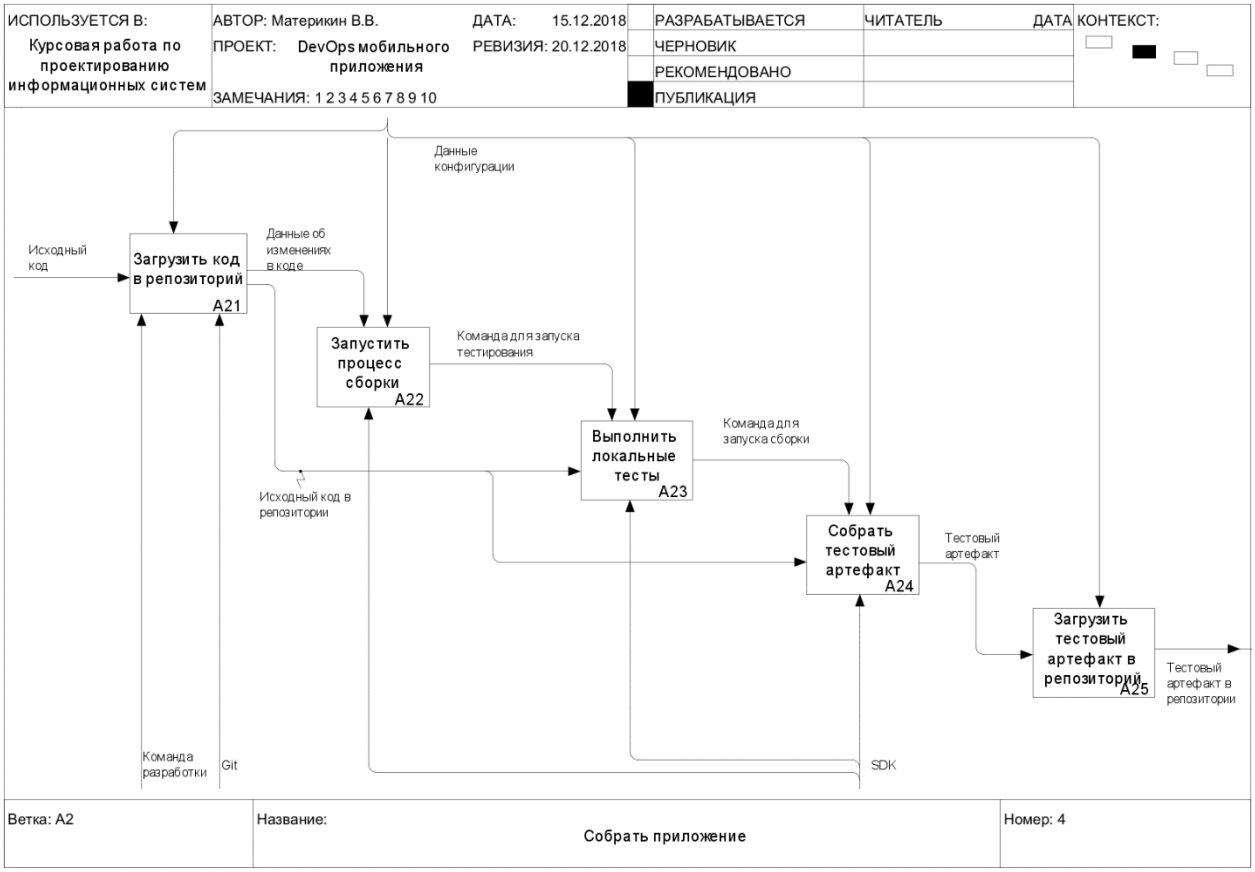

Рис. 3. Декомпозиция процесса A2: «Собрать приложение»

ГЛАВА 2. МОДЕЛЬ ПОТОКОВ ДАННЫХ (DFD)

DFD – общепринятое сокращение от англ. data flow diagrams – диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ [3].

Диаграммы потоков данных показывают, как каждый процесс преобразует свои входные данные в выходные, и выявляют отношения между этими процессами [4].

На диаграммах DFD используются следующие виды хранилищ данных:

- База данных системы на сервере БД.
- Репозиторий кода на сервере.
- Репозиторий артефактов на сервере.

Наименования объектов базы данных системы приводятся в формате «Таблица БД: Название таблицы».

В процессе декомпозиции функциональных блоков было построено 8 диаграмм потоков данных, которые представлены на рисунках 4-10. На данных диаграммах модули DFD являются экранными формами или скриптами. Вид модуля указан в его наименовании.

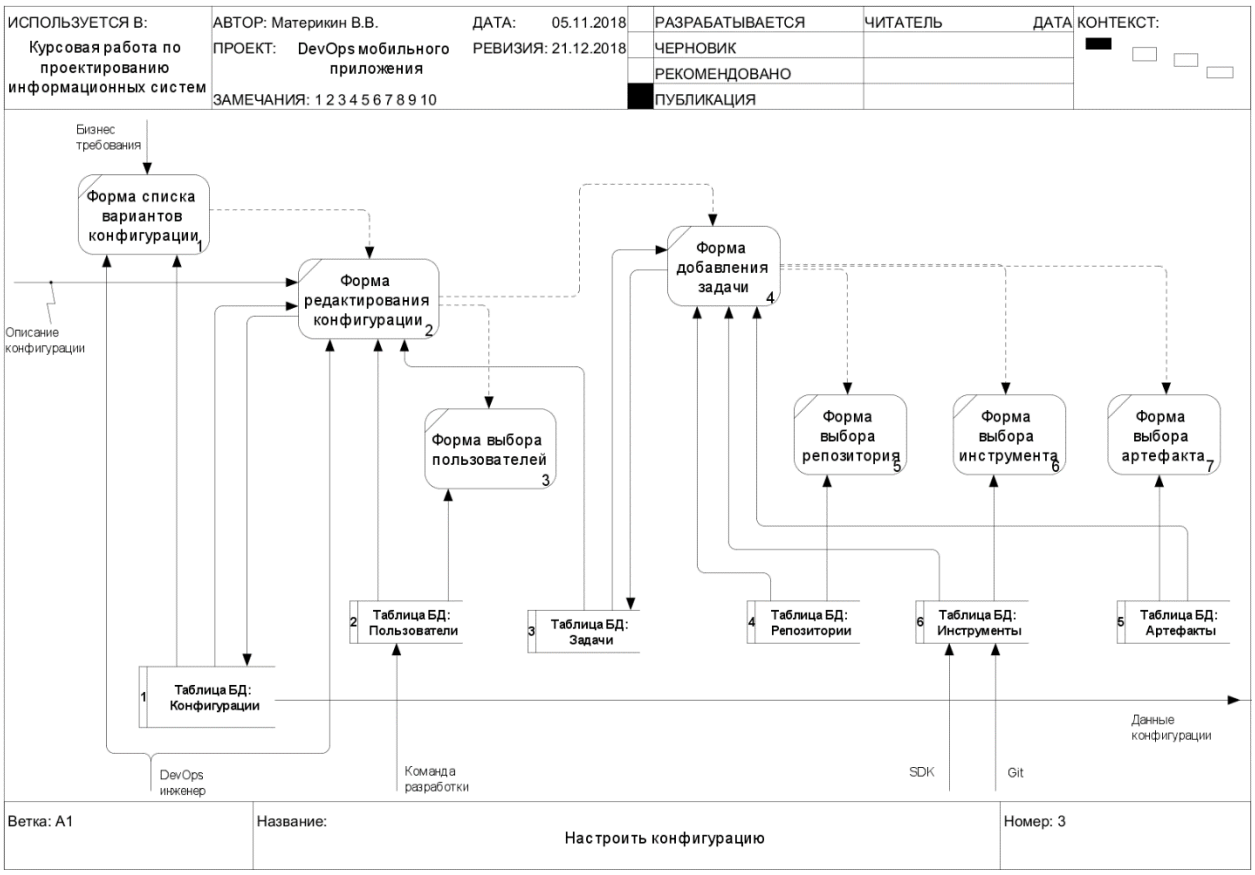

Рис. 4. Диаграмма потоков данных блока A1: «Настроить конфигурацию»

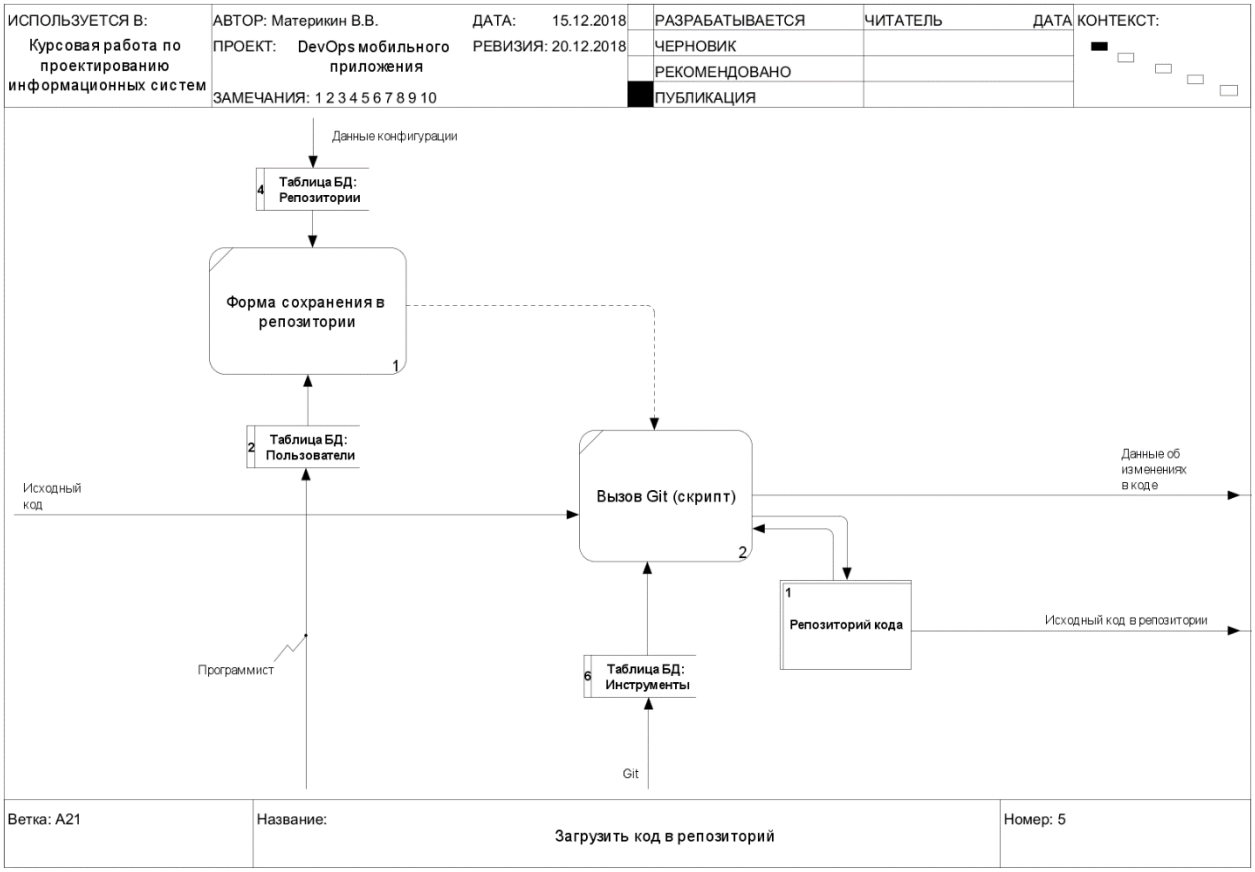

Рис. 5. Диаграмма потоков данных блока A21: «Загрузить код в репозиторий»

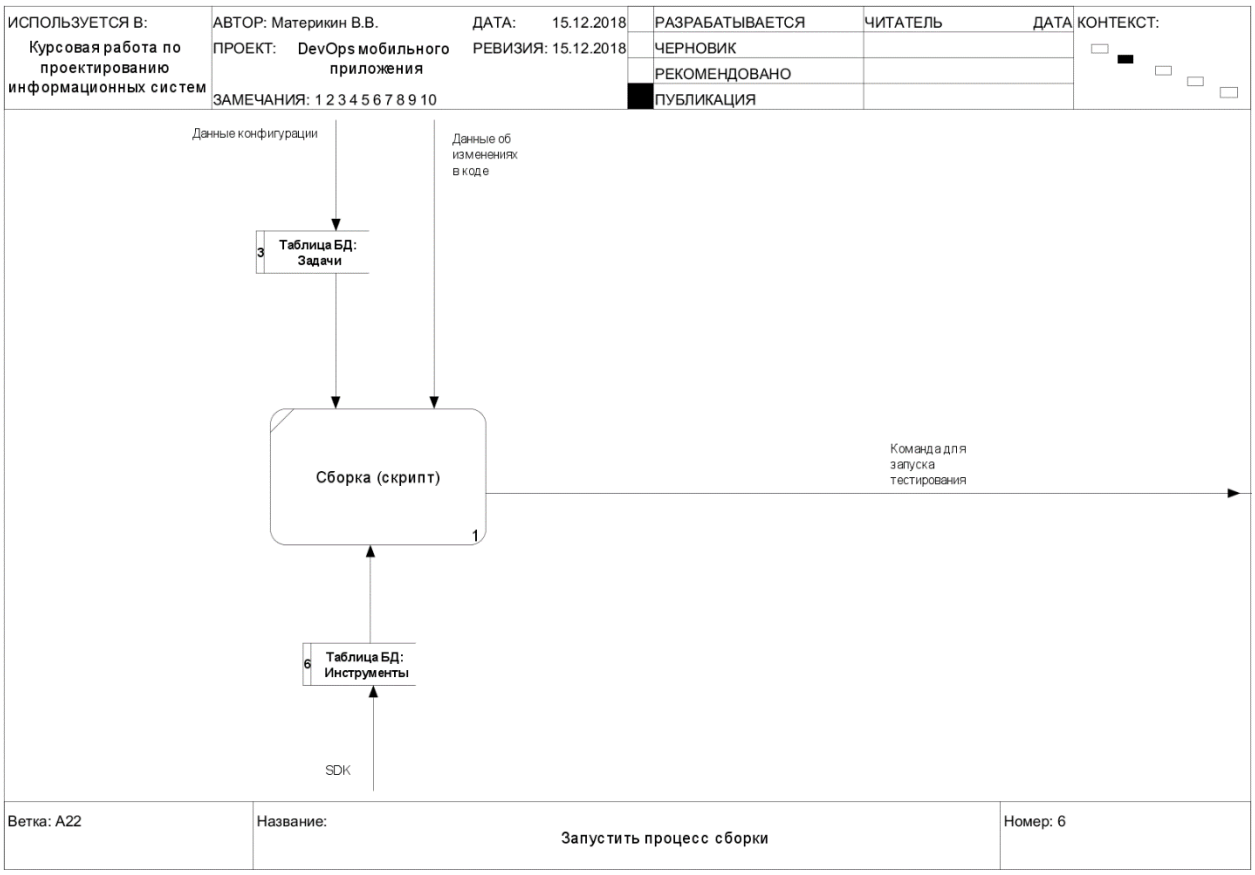

Рис. 6. Диаграмма потоков данных блока A22: «Запустить процесс сборки»

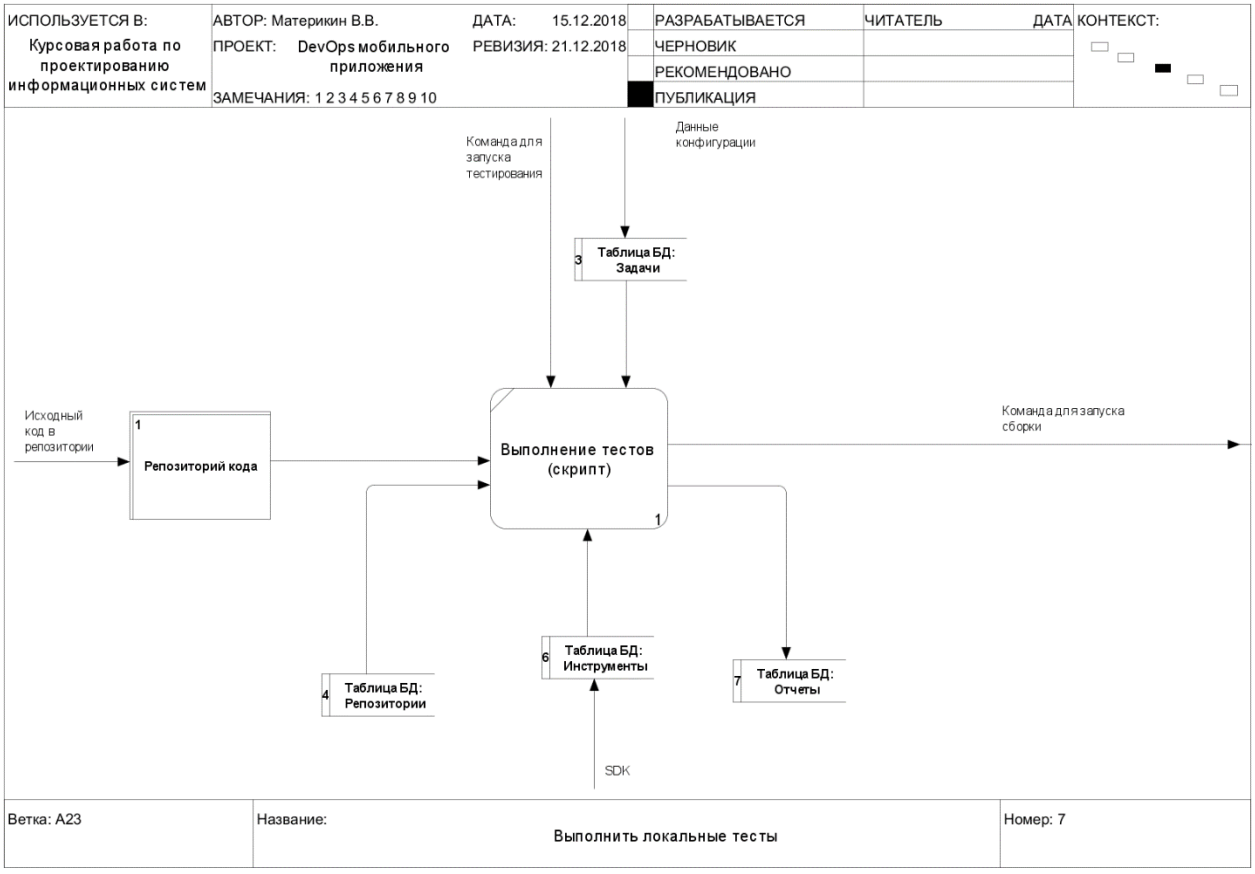

Рис. 7. Диаграмма потоков данных блока A23: «Выполнить локальные тесты»

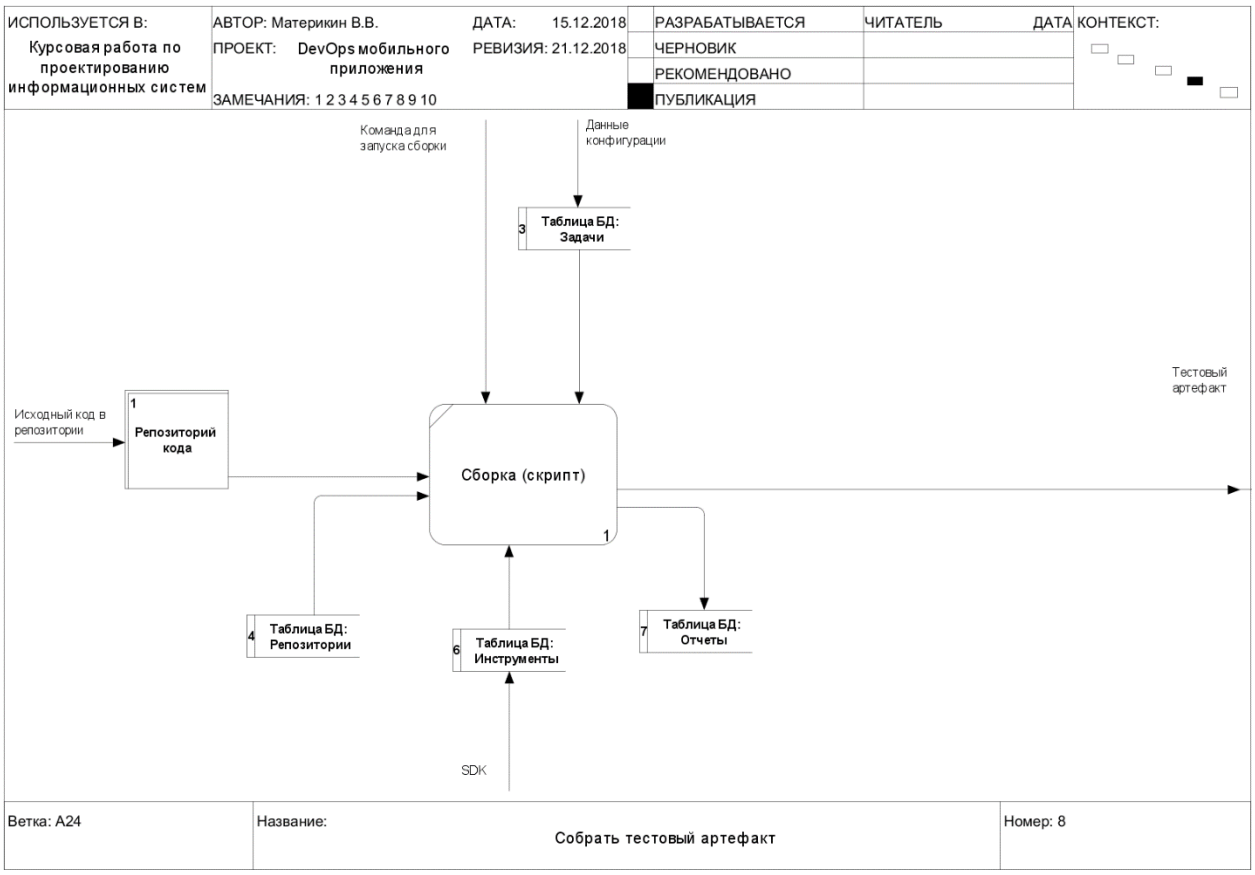

Рис. 8. Диаграмма потоков данных блока A24: «Собрать тестовый артефакт»

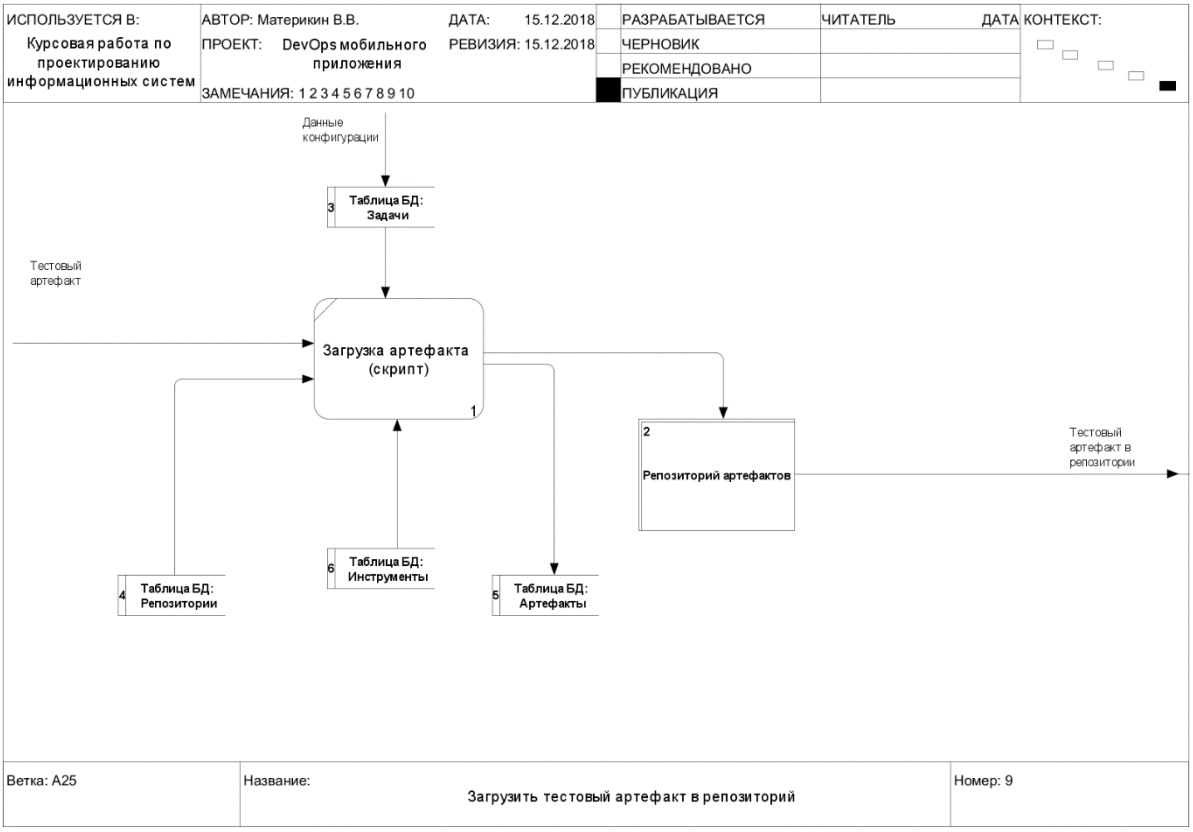

Рис. 9. Диаграмма потоков данных блока A25: «Загрузить тестовый артефакт в репозиторий»

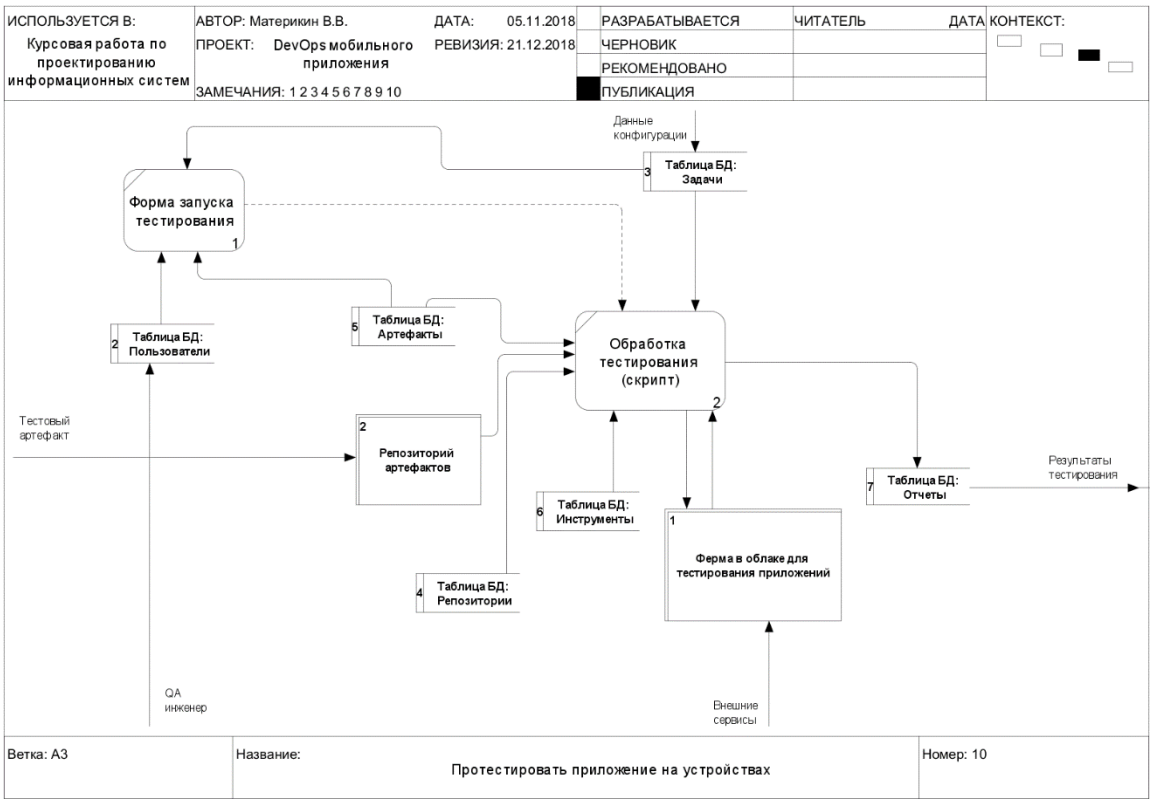

Рис. 10. Диаграмма потоков данных блока A3: «Протестировать приложение на устройствах»

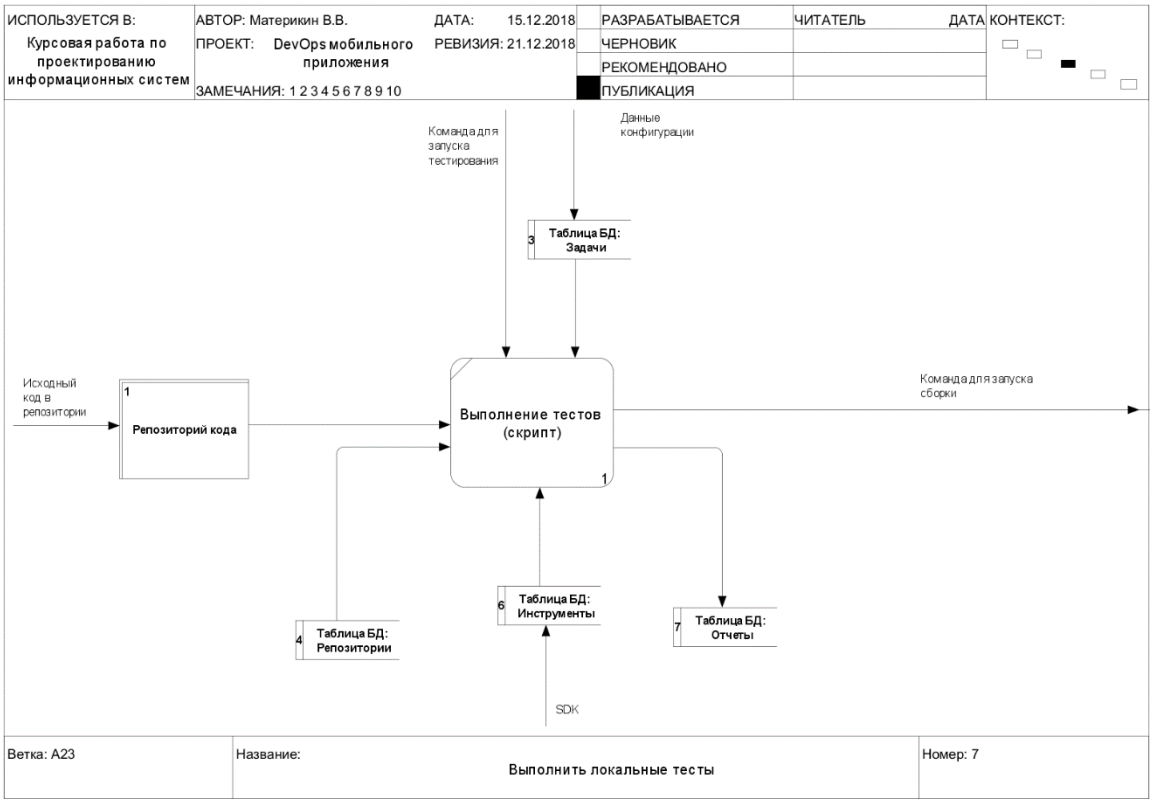

Рис. 11. Диаграмма потоков данных блока A23: «Распространить приложение»

На основе полученных модулей и хранилищ данных был произведен расчет числовых показателей трудозатрат на разработку информационной системы. Было рассчитано количество не выровненных функциональных точек (UFP), что отражено на рисунке 12.

Номер	Наименование	Модулей	Хранилищ данных	UFP
A0	Система для автоматизации DevOps			
A1	Настроить конфигурацию	7	6	70
A2	Собрать приложение	6	21	171
A3	Протестировать приложение на устройствах	2	7	57
A4	Распространить приложение	2	7	57
				355

Рис. 12. Расчёт UFP

Расчеты, выполненные методом FPA IFPUG (рис. 13) на основании данных функциональной модели, позволяют оценить сложность требуемых для создания информационной системы программных средств в 330 выровненных функциональных точек (DFP), а объем программного кода на языке программирования PL/SQL - в 16508 строк кода.

FPA IFPUG			
Характеристики			
1	Обмен данными	2	0-5
2	Распределенная обработка	2	0-5
3	Производительность (время отклика)	2	0-5
4	Ограничения аппаратные	2	0-5
5	Транзакционная нагрузка	2	0-5
6	Взаимодействие с пользователем	2	0-5
7	Эргономика	2	0-5
8	Интенсивность изменения данных	2	0-5
9	Сложность обработки	2	0-5
10	Повторное использование	2	0-5
11	Удобство инсталляции	2	0-5
12	Удобство администрирования	2	0-5
13	Портируемость	2	0-5
14	Гибкость	2	0-5
		28	
	VAF:	0,93	
	UFP:	355	
	DFP:	330	
	SLOC:	16508	
	KLOC:	17	

Рис. 13. Оценка объема работ методом FPA IFPUG

Расчеты, выполненные методом COCOMO II (рис. 14), позволяют оценить общие трудозатраты проекта разработки программных средств в 64 человеко-месяца, а ожидаемую продолжительность проекта – в 12 месяцев.

COCOMO II			
Масштаб			
1	опыт аналогичных разработок	3,72	6.20 4.96 3.72 2.48 1.24
2	гибкость процесса	3,04	5.07 4.05 3.04 2.03 1.01
3	разрешение рисков	4,24	7.07 5.65 4.24 2.83 1.41
4	сработанность команды	3,29	5.48 4.38 3.29 2.19 1.10
5	зрелость процессов	4,68	7.80 6.24 4.68 3.12 1.56
	SF:	18,97	
	E:	1,10	
Трудоемкость			
1	квалификация персонала	1,00	2.12 - 0.5
2	надежность продукта	1,00	0.49 - 2.72
3	повторное использование	1,00	0.95 - 1.24
4	сложность платформы разработки	1,00	0.87 - 2.61
5	опыт персонала	1,00	1.59 - 0.62
6	оборудование коммуникаций	1,00	1.43 - 0.62
7	сжатие расписания	1,00	1.43 - 1.00
	EM:	1,00	
	PM:	64	ч/мес
	TDEV:	12	мес

Рис. 14. Расчёт трудозатрат методом COCOMO II

В ходе работы над курсовым проектом был выполнен расчет ожидаемого эффекта проекта, представленный в таблице 1.

Таблица 1.

Расчёт эффекта от проекта

Рассматриваемый период – 1 месяц, 21 рабочий день	
Расчет экономии времени и денежных ресурсов при реализации проекта для блока А2 «Собрать приложение»	
Стоимость часа работы программиста: 10 руб/мин В среднем программист собирает тестовую версию проекта 1 раз в час Средний рабочий день: 8 часов Итого в среднем 8 сборок в день Рассматривается большой проект с несколькими модулями и множеством тестов	
С использованием ИС	**С использованием отдельных прикладных программ**
Блок А2 в ИС автоматически выполняет все действия по сборке после запуска процесса пользователем	Необходимо использовать несколько прикладных программ и управлять ими вручную
Время полного завершения процесса сборки ~ 2 минуты За один рабочий день: 8 * 2 мин. = 16 мин. 16 мин * 10 руб/мин = 160 руб	Время сборки ~ 3 минуты Время на анализ результатов ~ 2 минуты Время работы с Git ~ 2 минуты Время на загрузку артефакта ~ 1 минута За один рабочий день: 8 * 8 мин. = 64 мин. 64 мин. * 10 руб/мин = 640 руб.

<table>
<tr><td>За рассматриваемый период:
21 * 16 мин. = 336 мин.
336 мин * 10 руб. = 3360 руб.</td><td>За рассматриваемый период:
21 * 64 мин. = 1344 мин.
1344 мин * 10 руб/мин = 13440 руб.</td></tr>
<tr><td colspan="2">Итого:
1344 мин – 336 мин = 1008 мин ~ 17 часов (~10%)
13340 руб - 3360 руб = 9980 руб (~10%)</td></tr>
<tr><td colspan="2">Расчет экономии времени и денежных ресурсов
при реализации проекта для блока А3 «Протестировать приложение на устройствах»</td></tr>
<tr><td colspan="2">QA-инженер – 1
Средняя стоимость работы QA-инженера: 8 руб/мин
Тестирование запускается 8 раз в день в среднем по 5 минут на каждое устройство
Среднее время тестирования в день на каждом устройстве: 40 минут = 0,7 часа</td></tr>
<tr><td>С использованием ИС</td><td>С использованием физических устройств и ручного запуска тестов</td></tr>
<tr><td>При запуске тестирования ИС автоматически запускает тестирование на ферме устройств во внешнем сервисе и обрабатывает результаты</td><td>QA-инженер вручную запускает тесты на локальных устройствах и анализирует результаты</td></tr>
<tr><td>Средняя стоимость сервиса, предоставляющего виртуальные фермы:
42 руб/ч = 0,7 руб/мин

Время для запуска тестирования: 1 мин
Время для анализа результатов: 5 мин

За один рабочий день:
8 * 11 мин. = 88 мин.
8 * 5 мин * 0,7 руб/мин = 28 руб.
88 мин * 8 руб/мин = 704 руб.

За рассматриваемый период:
21* 88 мин. = 1848 мин.
21 * 8 * 5 мин * 0,7 руб/мин = 588 руб.
21 * 88 мин * 8 руб/мин = 704 руб.</td><td>Средняя стоимость устройства: 30000 руб
Новые устройства закупаются каждый год
Стоимость тестирования на 1 устройство:
30000 / (0,7*21*12) = 170 руб/ч = 2,8 руб/мин

Время запуска тестирования: 10 мин
Время для анализа результатов : 15 мин

За один рабочий день:
8 * 30 мин. = 240 мин.
8 * 5 мин * 2,8 руб/мин = 112 руб.
240 мин * 8 руб/мин = 1920 руб.

За рассматриваемый период:
21 * 240 мин. = 5040 мин.
21 * 8 * 5 мин * 2,8 руб/мин = 2352 руб.
21 * 240 мин * 8 руб/мин = 40320 руб</td></tr>
<tr><td colspan="2">Итого:
5040 мин – 1848 мин = 3192 мин ~ 53 часа (~32%)
(40320+2352) – (704+588) руб = 45380 руб (~41%)</td></tr>
</table>

ГЛАВА 3. ДИАГРАММЫ КЛАССОВ (ERD)

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их кооперацией, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования [5].

В результате моделирования были созданы 3 диаграммы классов: для потоков (рис. 15), для модулей (рис. 16) и для ролей (рис. 17).

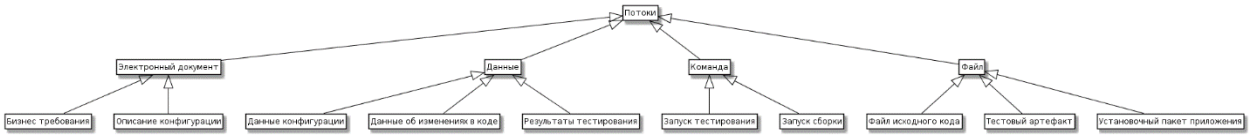

Рис. 7. Диаграмма классов для потоков

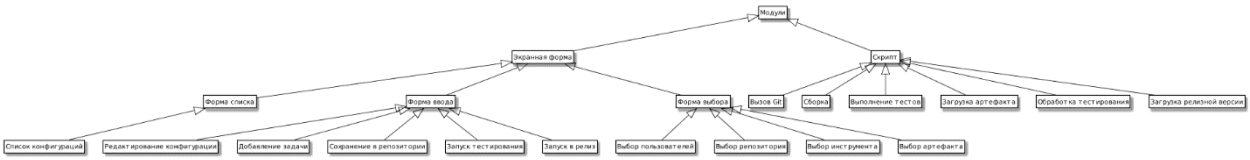

Рис. 8. Диаграмма классов для модулей

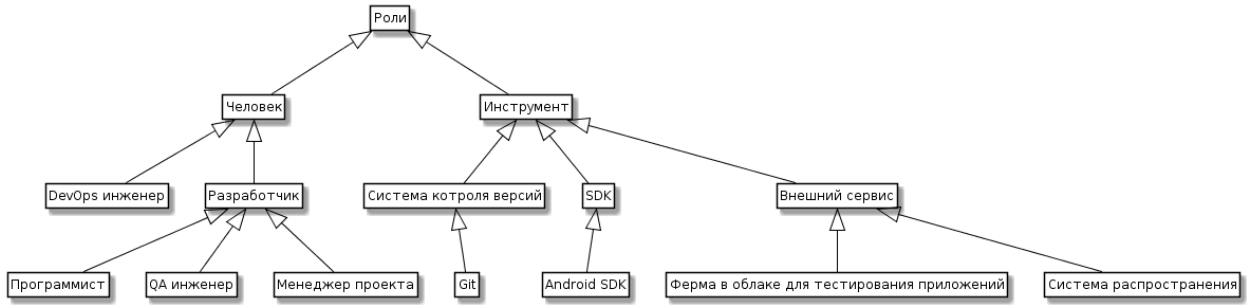

Рис. 9. Диаграмма классов для ролей

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы было выполнено моделирование информационной системы для автоматизации процесса разработки и обновления мобильного приложения. В процессе моделирования были выделены процессы, с помощью автоматизации которых можно добиться ощутимого экономического эффекта от использования проектируемой ИС.

Также были выполнены расчеты числовых показателей эффективности, которые показали, что в среднем при внедрении ИС можно сократить время разработки на 10-32 % и уменьшить денежные затраты на 10-41%.

В дальнейшем, полученные результаты моделирования будут использованы в работе над выпускной квалификационной работой «Создание информационной системы поддержки гибкой разработки программных средств для мобильных устройств».

СПИСОК ЛИТЕРАТУРЫ

1. IDEF0 – Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/IDEF0, свободный. Дата обращения: 20.12.2018 г.

2. Знакомство с нотацией IDEF0 и пример использования / Блог компании Trinion / Хабр [Электронный ресурс] – Режим доступа: https://habr.com/company/trinion/blog/322832/, свободный. Дата обращения: 20.12.2018 г.

3. DFD – Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/DFD, свободный. Дата обращения: 20.12.2018 г.

4. НОУ ИНТУИТ | Лекция | Моделирование бизнес-процессов средствами BPwin (часть 2) [Электронный ресурс] – Режим доступа: https://www.intuit.ru/studies/courses/2195/55/lecture/1632?page=2, свободный. Дата обращения: 20.12.2018 г.

5. Диаграмма классов – Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Диаграмма_классов, свободный. Дата обращения: 20.12.2018 г.