

Министерство науки и высшего образования Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)  
Институт прикладной математики и компьютерных наук  
Кафедра программной инженерии

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК  
Руководитель ООП  
д-р физ.-мат. наук, профессор  
\_\_\_\_\_  
О.А. Змеев  
« 29 » \_\_\_\_\_ 2019 г.

### МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

#### РЕАЛИЗАЦИЯ СИСТЕМЫ АДАПТАЦИИ УНИФИЦИРОВАННОГО ПРОЦЕССА К ТРЕБОВАНИЯМ КОНКРЕТНОГО ПРОЕКТА

по основной образовательной программе подготовки магистров  
«Управление проектами по разработке программного обеспечения»  
направление подготовки  
02.04.02 Фундаментальная информатика и информационные технологии

Трушникова Елена Николаевна

Научный руководитель ВКР  
доцент, канд. тех. наук  
\_\_\_\_\_  
А.М. Бабанов  
« 29 » \_\_\_\_\_ 2019 г.

Автор работы  
студент группы № 931709  
\_\_\_\_\_  
Е. Н. Трушникова

Томск–2019

## **Реферат**

Магистерская диссертация 89 с., 32 рис., 1 приложение, 11 источников.

УНИФИЦИРОВАННЫЙ ПРОЦЕСС, TYPESCRIPT, ANGULAR,  
АНАЛИЗ ТРЕБОВАНИЙ, ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ  
СИСТЕМ

Цель работы – реализовать систему адаптации унифицированного процесса к требованиям конкретного проекта.

Результат работы – реализована система адаптации унифицированного процесса к требованиям конкретного проекта.

## СОДЕРЖАНИЕ

<b>Реферат .....</b>	<b>1</b>
<b>Введение .....</b>	<b>4</b>
<b>1 Унифицированный процесс .....</b>	<b>6</b>
1.1 Определение унифицированного процесса.....	6
1.2 Краткая история развития UP .....	6
1.3 Аксиомы UP.....	7
1.4 Структура UP .....	8
<b>2 Постановка проблемы.....</b>	<b>11</b>
2.1 Анализ существующих методологий .....	11
2.2 Анализ существующих систем.....	15
2.3 Формальное описание системы адаптации Унифицированного процесса к требованиям конкретного проекта.....	17
<b>3. Разработка системы адаптации.....</b>	<b>33</b>
3.1 Определение требований.....	33
3.2 Анализ и проектирование системы .....	34
3.3 Реализация системы.....	39
3.4 Тестирование системы.....	46
3.5 Развертывание и внедрение системы.....	46
3.6 Описание реализованной системы адаптации .....	48
<b>Заключение.....</b>	<b>57</b>
<b>Глоссарий.....</b>	<b>58</b>
<b>Список использованных источников и литературы .....</b>	<b>60</b>
<b>Приложение А .....</b>	<b>62</b>

## **Введение**

**Актуальность исследования.** Каждый проект уникален и требует определённых подходов для решения тех или иных вопросов возникающих в процессе работы над ним, поэтому методологии расширяются и совершенствуются. Тема данной работы является актуальной, так как решает вопросы оптимизации и унификации процессов управления проектами с целью выпуска продукта в максимально сжатые сроки без потери качества.

Реализация системы адаптации позволит оптимизировать процессы планирования проекта и итераций, построить карту проекта с будущими процессами, ожидаемыми результатами и исполнителями. Сгенерированная информация позволит эффективно управлять проектом и командой, сделает жизненный цикл более прозрачным. Будущие пользователи данного продукта – это менеджеры проектов.

**Объект исследования:** методологии и системы адаптации управления проектами.

**Предмет исследования:** – унифицированный процесс и рациональный унифицированный процесс.

**Цель исследования:** – реализовать систему адаптации унифицированного процесса к требованиям конкретного проекта.

**Задачи исследования:**

- Формализовать унифицированный процесс.
- Спроектировать систему процесса адаптации.
- Реализовать систему процесса адаптации.

**Теоретической и методологической основой исследования явились:**

- Книга Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения.
- База знаний HTML Rational Software Corporation.

**Методы исследования:** В ходе исследования применялись теоретические и эмпирические методы. К первым относятся изучение и анализ

методологий на базе унифицированного процесса, ко вторым – знакомство с программным обеспечением аналогов.

**Теоретическая значимость исследования:**

- Проведен анализ методологии унифицированного процесса (UP).
- Проведен анализ методологий построенных на базе UP.
- Формализована и спроектирована система адаптации унифицированного процесса.

**Практическая значимость:**

- Реализована система адаптации унифицированного процесса к требованиям конкретного проекта.
- Результаты работы могут быть использованы в управлении проектами по разработке программного обеспечения.

**Структура исследовательской работы:**

**Во введении** обоснована актуальность данной работы, представлена тема исследования, цель, задачи, предмет, объект, охарактеризованы методы исследования, представлена методологическая основа и источники информации.

**В первой разделе** «Унифицированный процесс» представлено описание определения, история развития, аксиомы и структура изучаемой методологии.

**Во втором разделе** «Постановка проблемы» представлен анализ существующих методологий и систем созданных на базе унифицированного процесса, разработано формальное описание системы адаптации, спроектированы диаграммы активностей потоков работ каждого рабочего процесса для всех фаз унифицированного процесса.

**В третьем разделе** «Реализация системы адаптации» представлены этапы разработки системы: определение требований, анализ и проектирование, реализация, тестирование, развертывание системы. Представлено описание реализованной системы со скриншотами.

**В заключении**, подведены итоги исследования, сформулированы выводы по рассматриваемой теме.

## **1 Унифицированный процесс**

### **1.1 Определение унифицированного процесса**

Унифицированный процесс есть процесс разработки программного обеспечения. Процесс разработки программного обеспечения — это сумма различных видов деятельности, необходимых для преобразования требований пользователей в программную систему. Однако Унифицированный процесс — это больше чем единичный процесс, это обобщенный каркас процесса, который может быть специализирован для широкого круга программных систем, различных областей применения, уровней компетенции и размеров проекта.<sup>1</sup>

### **1.2 Краткая история развития UP**

Все началось в 1967 году с компании Эриксон. Система моделировалась при помощи набора связанных между собой блоков. Блоки из нижнего уровня собирались в подсистемы из верхнего уровня. Сегодня это называется «разработка, основанная на компонентах». Создателем данного метода является Ивар Джекобсон.

В 1987 году Ивар Джекобсон покинул компанию Эриксон и основал компанию Objectory AB. В течение нескольких лет он разрабатывал процесс названный Objectory (Фабрика объектов), используя специализированный стандарт объектного моделирования – SDL (Specification and Description Language). В результате стало очевидно, что варианты использования управляют разработкой и архитектура направляет разработчиков информируя заинтересованных лиц.

С 1988 по 1995 год вышла серия выпусков Objectory с 1.0 до 3.8. Во время спада в 1995 году компанию Objectory AB приобрела Rational Software Corporation.

---

<sup>1</sup> Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. Спб.: Питер, 2002.

В ноябре 1997 года SDL был вытеснен Унифицированным языком моделирования (UML).

К середине 1998 года Rational Objectory Process превратился в полномасштабный процесс. И уже в июне компания выпустила новую версию продукта, Rational Unified Process 5.0.

Унифицированный процесс компании Rational (Rational Unified Process, RUP) – это коммерческая версия UP от IBM, которая поглотила Rational Corporation в 2003 году.

Сегодня UP – это открытый обобщенный продукт (SEP – Software Engineering Process), а RUP коммерческий продукт расширяющий UP.

Как видно из названий произошла унификация процесса по многим параметрам, прежде всего это использование унифицированного языка моделирования и приведение к единой форме многих методологий.<sup>2</sup>

### **1.3 Аксиомы UP**

Унифицированный процесс является:

- Управляемым вариантами использования и означает, что варианты использования применяются в качестве первичных артефактов для определения желаемого поведения системы и для передачи этого поведения владельцам системы. Также это означает, что варианты использования являются первичными исходными данными для анализа, проектирования, реализации и тестирования системы.
- Архитектурно-ориентированным означает, что архитектура системы используется в качестве первичного артефакта для построения концепции, создания, управления и развития системы в ходе ее разработки.
- Итеративным и инкрементным означает что проект разбивается на небольшие итерации что позволяет реализовать функционал по частям.

---

<sup>2</sup> Арлоу Д., Нейштадт А. UML2 и унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. - СПб.: Символ-плюс, 2007.

- Управляемым рисками означает, что каждый новый релиз ориентирован на уменьшение наиболее существенных рисков для того, чтобы добиться успеха проекта.<sup>3</sup>

## **1.4 Структура UP**

Унифицированный процесс отвечает на вопросы – когда, кто, что и с помощью чего реализуется проект. Когда – это рабочие потоки, кто – это исполнители, что – это артефакты (результаты) и с помощью чего – это утилиты и программные продукты используемые в ходе работы над проектом.

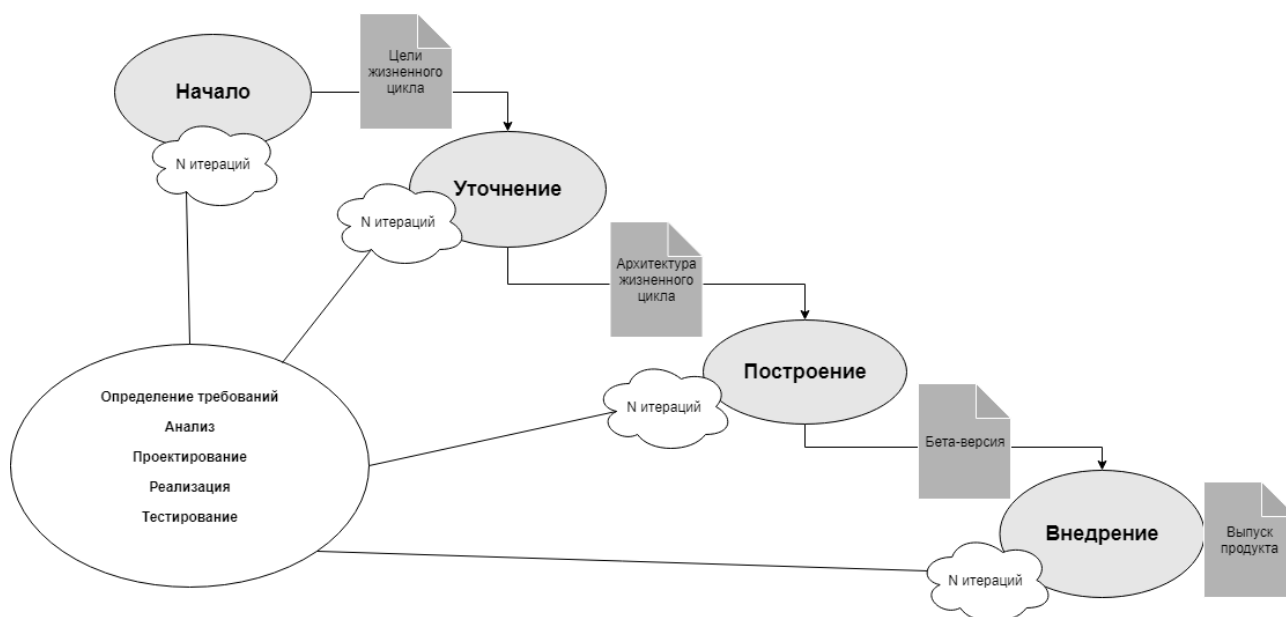
Жизненный цикл унифицированного проекта состоит из четырех фаз: «Начало», «Уточнение», «Построение» и «Внедрение». Каждая фаза имеет свои контрольные точки и завершается достижением определенного результата, т.е. созданием одного или нескольких артефактов, которые в свою очередь могут использоваться в последующих фазах.

Особенность унифицированного процесса в том что он ориентируется на цели, а не на создание артефактов. Каждая фаза завершается определенной контрольной точкой, состоящей из набора условий, которым надо удовлетворить, условия в свою очередь можно включать и выключать в зависимости от конкретных потребностей проекта. Далее подробно опишем цели фазы и ее контрольные точки. На рисунке 1.4.1 представлена структура UP.

---

<sup>3</sup> Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. С. 33-38.





*Рисунок 1.4.1 – Структура унифицированного процесса*

Фаза «Начало» инициирует проект. Цель данной фазы дать ход проекту. В рамках фазы «Начало» проводятся работы по экономическому обоснованию, определяются основные требования для создания предметной области с учетом мнений всех заинтересованных лиц, выявляются потенциальные риски. Основными исполнителями в данной фазе являются менеджер проектов и архитектор системы.

В фазе «Начало» упор делается на рабочие потоки, определение требований и анализ, допустима реализация прототипа будущей системы в качестве доказательства концепции проекта (proof of concept). Контрольной точкой фазы «Начало» являются цели жизненного цикла. Для реализации набора условий выхода из фазы, т.е. достижение целей контрольной точки, иногда требуется создание артефакта.

Для фазы «Уточнение» одной из ключевых целей является разработка базовой архитектуры системы.

Основное внимание отводится детализации предметной области системы и требований, создание стабильной архитектуры, реализация и тестирование базовой архитектуры для системы. Контрольная точка фазы – Архитектура жизненного цикла.

Цель фазы «Построение» - завершить определение требований, анализ и проектирование и развить базовую архитектуру в завершённую систему. Основной поток данной фазы – реализация, кроме того на передний план выходит тестирование. Контрольная точка – Бета версия системы.

И наконец фаза «Внедрение» отвечает за окончательное развертывание системы и дальнейшее сопровождение. К данной фазе относится устранение всех дефектов, обнаруженных при тестировании бета-версии системы. Цели данной фазы следующие: исправление дефектов, создание руководств пользователей, обучение пользователей, послепроектная ретроспектива.

Основное внимание отводится потокам реализации и тестирования. Если на данной стадии задействованы потоки определения требований и анализа значит с проектом что-то не так. Контрольная точка фазы «Внедрение» - Выпуск продукта.

В начале каждой фазы планируется и уточняется количество итераций. Каждая итерация характеризуется приращением результатов. В рамках итерации запускаются основные рабочие потоки. К основным рабочим потокам относятся: определение требований, анализ и проектирование, реализация, тестирование, развертывание (внедрение).<sup>4</sup>

---

<sup>4</sup> Арлоу Д., Нейштадт А. UML2 и унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. С. 59.

## **2 Постановка проблемы**

На сегодняшний день существуют различные методологии управления проектами. Одни методологии являются универсальными и могут применяться в разных областях, другие методологии работают только с конкретными типами проектов. С развитием рынка мобильных приложений выросло количество проектов по разработке программного обеспечения для мобильных платформ, это породило развитие новых методологий и подходов, ускоряющий выпуск продукта. Разработка программных продуктов проходит в условиях жесткой конкуренции, где наряду с актуальностью решаемых задач и качеством продукта, решающим критерием успешности проекта является скорость выпуска коммерческого релиза. Поэтому основной целью любой методологии является унификация процессов и эффективное управление проектами. Под эффективностью понимается совокупность различных параметров: достижение целей проекта, улучшения качества продукта, оптимизация процессов управления и т.д.

В наши дни популярными являются «облачные решения», не требующие установки и не зависящие от возможностей и системных настроек вашего компьютера и установленного программного обеспечения. На первое место выходят такие критерии как: интуитивно понятный интерфейс, простая и понятная логика, функциональность. Исходя из проведенного анализа, можно сделать вывод, что реализуемая система является востребованной и актуальной.

### **2.1 Анализ существующих методологий**

Унифицированный процесс лежит в основе ряда методологий по разработке программного обеспечения, одни из них расширяют UP, другие упрощают. Анализ методологий позволит понять различия, сильные и слабые стороны каждого подхода и будет полезен для разработки системы адаптации.

Ниже представлен список наиболее популярных методологий:

- Рациональный унифицированный процесс разработки программного обеспечения (RUP - Rational Unified Process) - это частный случай унифицированного процесса. RUP состоит из четырех фаз и пяти основных дисциплин (стадий). В рамках RUP разработка ведется в несколько итераций, где результатом каждой итерации является окончательная версия части системы с набором артефактов. RUP одна из популярных методологий UP. На рисунке 2.1.1 представлена нагрузка на каждую из фаз по дисциплинам (рабочим процессам).

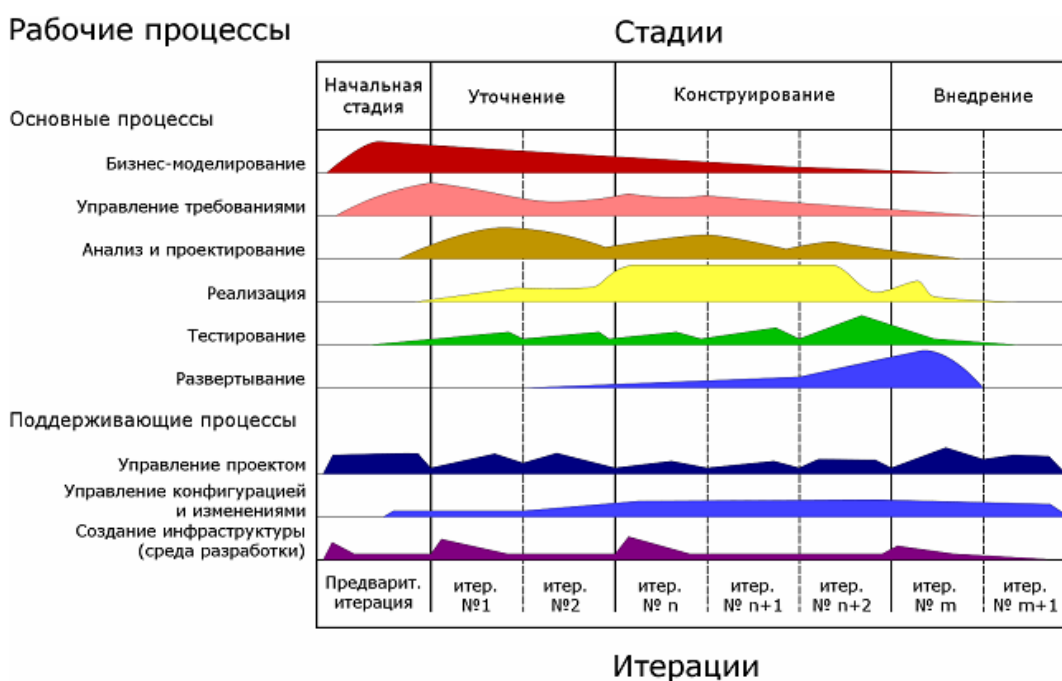


Рисунок 2.1.1 – Фазы и дисциплины рационального унифицированного процесса

Источник: [https://ru.wikipedia.org/wiki/Rational\\_Unified\\_Process](https://ru.wikipedia.org/wiki/Rational_Unified_Process)

- Основной унифицированный процесс (EssUP - Essential Unified Process) – был разработан Иваром Якобсоном, как усовершенствованный RUP. EssUP использует такие практики как, сценарии использования, итеративная разработка, разработка на основе архитектуры, командные практики и практики процессов которые заимствованы из RUP, CMMI и Agile.

- Корпоративный унифицированный процесс (EUP – Enterprise Unified Process), данная методология EUP является расширением известной методологии RUP (Rational Unified Process), которая предназначена для поддержки софтверных проектов, использующих объектно-ориентированную парадигму. Методология предусматривает использование жизненного цикла разработки. Под понятием (enterprise) надо понимать сложную организационную систему, процессы которой могут быть автоматизированы. К уже знакомым фазам добавлены новые фазы – «Промышленная эксплуатация» и «Снятие с эксплуатации». К наследуемым дисциплинам из RUP в EUP добавлена «Операция и поддержка», а также семь корпоративных дисциплин (рабочих процессов): бизнес-моделирование предприятия, управление портфелем, архитектура предприятия, стратегическое повторное использование, управление персоналом, администрация предприятия и улучшения программного процесса.

- Упорядоченная гибкая разработка (DAD - Disciplined Agile Delivery) задумана как согласованная сквозная стратегия практического применения быстрых методов подготовки программных продуктов. DAD представляет собой Agile подход, который ставит на первое место кадры и ориентацию на обучение. Жизненный цикл данной концепции опирается на снижение рисков и повышение ценности продукта, ориентирован на достижение целей, является масштабируемым, действует в контексте предприятия. DAD – считается золотой серединой, между RUP и Scrum.

- Открытый унифицированный процесс (OpenUP - Open Unified Process) – итеративно-инкрементный метод разработки ПО. Представляет собой легкий и гибкий вариант RUP. OpenUP содержит в себе те же четыре фазы, что и RUP. Данная методология делит проект на итерации. Разработчики OpenUP самостоятельно решают вопросы выполнения задач итераций и передачи результатов. Для этого они сначала определяют и детально

описывают требования, а затем решают хорошо детализированные задачи из списка работ.<sup>5</sup>

- Базовый унифицированный процесс (BUP - Basic Unified Process) – это упрощенная версия RUP, оптимизированная для небольших проектов с командой до шести человек и с периодом разработки до шести месяцев. BUP содержит следующие дисциплины (рабочие процессы): требования, архитектура, разработка, тестирование, управление проектом и управление изменениями. Дисциплины для каждой из четырёх фаз содержат следующие потоки работ: управление итерацией, управление требованиями, определение архитектуры, разработка решения, проверка сборки, оценка и планирование следующей итерации, управление требованиями. В BUP используются следующие роли: аналитик, архитектор, разработчик, тестировщик, менеджер проектов, любая роль.

- Гибкий унифицированный процесс (AUP - Agile Unified Process) – упрощенная версия унифицированного процесса, разработанная Скотом Эмблером. Данная методология объединяет в себе элементы Agile методологий и унифицированного процесса. В частности AUP предлагает использовать разработку через тестирование, когда сначала пишутся тесты и далее пишется код, применение Agile моделирования и гибкое управление изменениями. AUP содержит семь дисциплин (рабочих процессов): моделирование, реализация, тестирование, развертывание, управление конфигурациями, управление проектом, организация среды.

- Унифицированная методология компании Oracle поддерживает весь жизненный цикл информационных технологий на предприятии, включая поддержку внедрения продуктов и технологий Oracle. Данная методология универсальна и у нее нет явной привязки к конкретным продуктам и технологиям, что даёт возможность её адаптации под другие типы ИТ-проектов. Она включает набор готовых шаблонов, руководство и структуру

---

<sup>5</sup> Свободная энциклопедия «Википедия» [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/OpenUP> (дата обращения: 10.05.2019).

работ, а так же содержит программные инструменты для управления рисками, связанными с проектами. OUM состоит из трёх областей: управление программами, поддержка ИТ-архитектуры, реализация проектов. На рисунке 2.1.2 представлены фазы и процессы данной методологии.

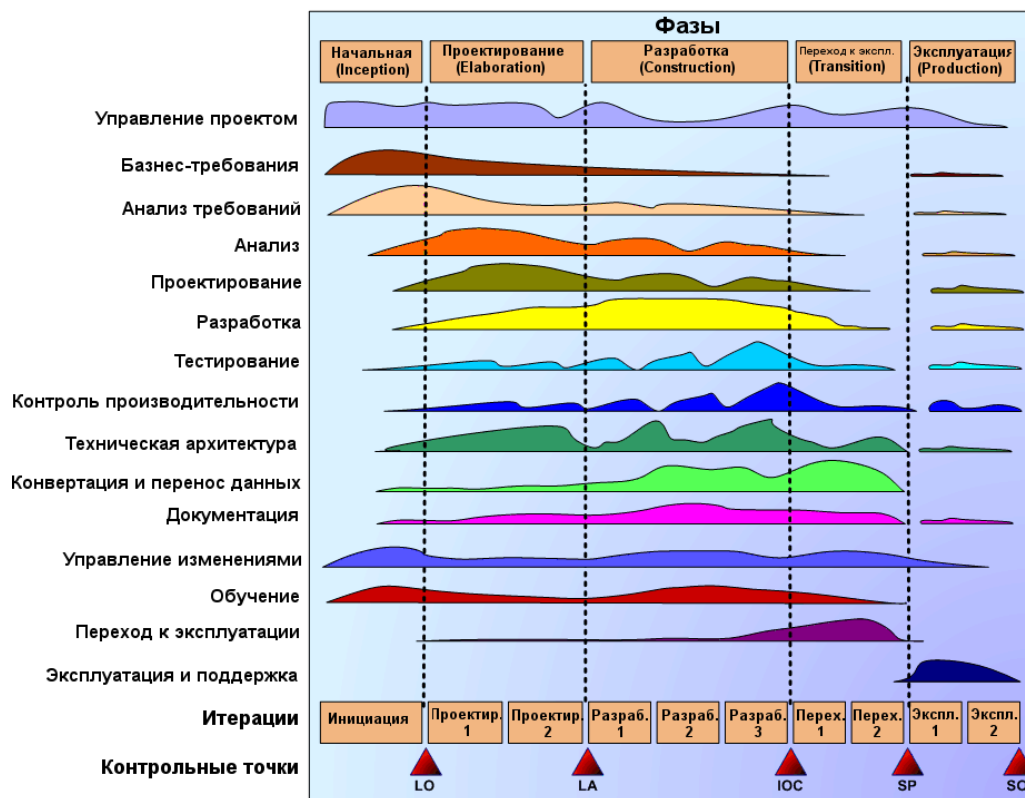


Рисунок 2.1.2 – Фазы и процессы унифицированной методологии Oracle

Источник: <https://stan1slav.blogspot.com/2011/11/oracle-unified-method.html>

## 2.2 Анализ существующих систем

Из приведенного выше списка видно, что существует достаточное количество методологий построенных на унифицированном процессе. Однако реализаций использующих данные методологии не так и много. Далее будут рассмотрены два решения, одно от компании IBM и второе от компании Eclipse.

RMS (Rational Method Composer) – это гибкая система управления процессами, которая включает инструмент создания процессов, а также обширную библиотеку процессов и практических рекомендаций. RMS позволяет создавать базы знаний процессов, которые можно просматривать,

управлять и публиковать. Эта база может включать как информацию предоставленную IBM, так и информацию из сторонних проектов, которую можно импортировать. Данный продукт разрабатывался как система управления содержанием, которое публикуется в виде HTML-файлов и предоставляет возможность управления процессной методологией менеджерам проектов посредством выбора, адаптации и быстрой сборки процессов для конкретных проектов разработки. RMC предоставляет предопределенный набор процессов для типичных проектных ситуаций, которые могут быть адаптированы под конкретные нужды. Он также предоставляет блоки построения процессов разработки, называемые процессами ключевых областей разработки, которые представляют собой лучшие практики разработки для конкретных дисциплин, технологий или стилей управления проектами. Эти строительные блоки формируют инструментарий, для быстрого создания процесса разработки основываясь на конкретных нуждах проекта. Кроме того в RMC можно настраивать специфические для организации процессы ключевых областей разработки. Наконец, процессы разработки созданные с помощью RMC могут быть опубликованы в виде вебсайтов. Безусловно инструмент очень мощный и функциональный, однако интуитивно сложен и перегружен. RMC – это десктоп приложение, которое требует установки, размер установочного пакета варьируется от 1.5 до 2.5 Гбайт.

Существует также бесплатное решение EPF Composer – это платформа с открытым исходным кодом, предназначенная для менеджеров проектов для разработки, адаптации и публикации методов и процессов для организаций и проектов. К сожалению EPF Composer не удалось скачать и проанализировать из-за серверной ошибки веб-ресурса на котором размещен данный продукт.

Популярный продукт Jira от компании Atlassian, позиционирующий себя инструментом управления проектом ориентированным на agile-команды, где разработка ведется короткими итерациями, является по факту системой



мониторинга задач и ошибок, и вопросы планирования и адаптации процессов к специфике проекта не решает.

### **2.3 Формальное описание системы адаптации Унифицированного процесса к требованиям конкретного проекта**

В процессе реализации формального описания системы адаптации за основу была принята база знаний RUP. Методология RUP детально проработана и содержит подробную информацию по фазам, рабочим процессам, потокам и по операциями внутри потоков. Кроме того для каждой деятельности потока описан список шагов для достижения результатов по данной деятельности.

Как видно RUP имеет некоторые отличия от UP в терминологии.

Таблица А.1 – Фазы унифицированного RUP

Название фазы	Названия рабочих процессов
Начало	Управление проектом Определение требований Анализ и проектирование Тестирование
Уточнение	Управление проектом Определение требований Анализ и проектирование Реализация Тестирование
Построение	Управление проектом Определение требований Анализ и проектирование Реализация Тестирование
Внедрение	Управление проектом Определение требований Анализ и проектирование Реализация Тестирование Внедрение и сопровождение

Чтобы в дальнейшем исключить разницу толкований в таблице А.1 представлены названия фаз и рабочих процессов, которые будут использованы далее в работе.

Из таблицы видно, что мы оставили фазы из UP и добавили рабочие процессы из RUP, ниже на рисунке 2.3.1 представлена получившаяся структура процесса адаптации, которая будет использована далее при проектировании и реализации системы адаптации.

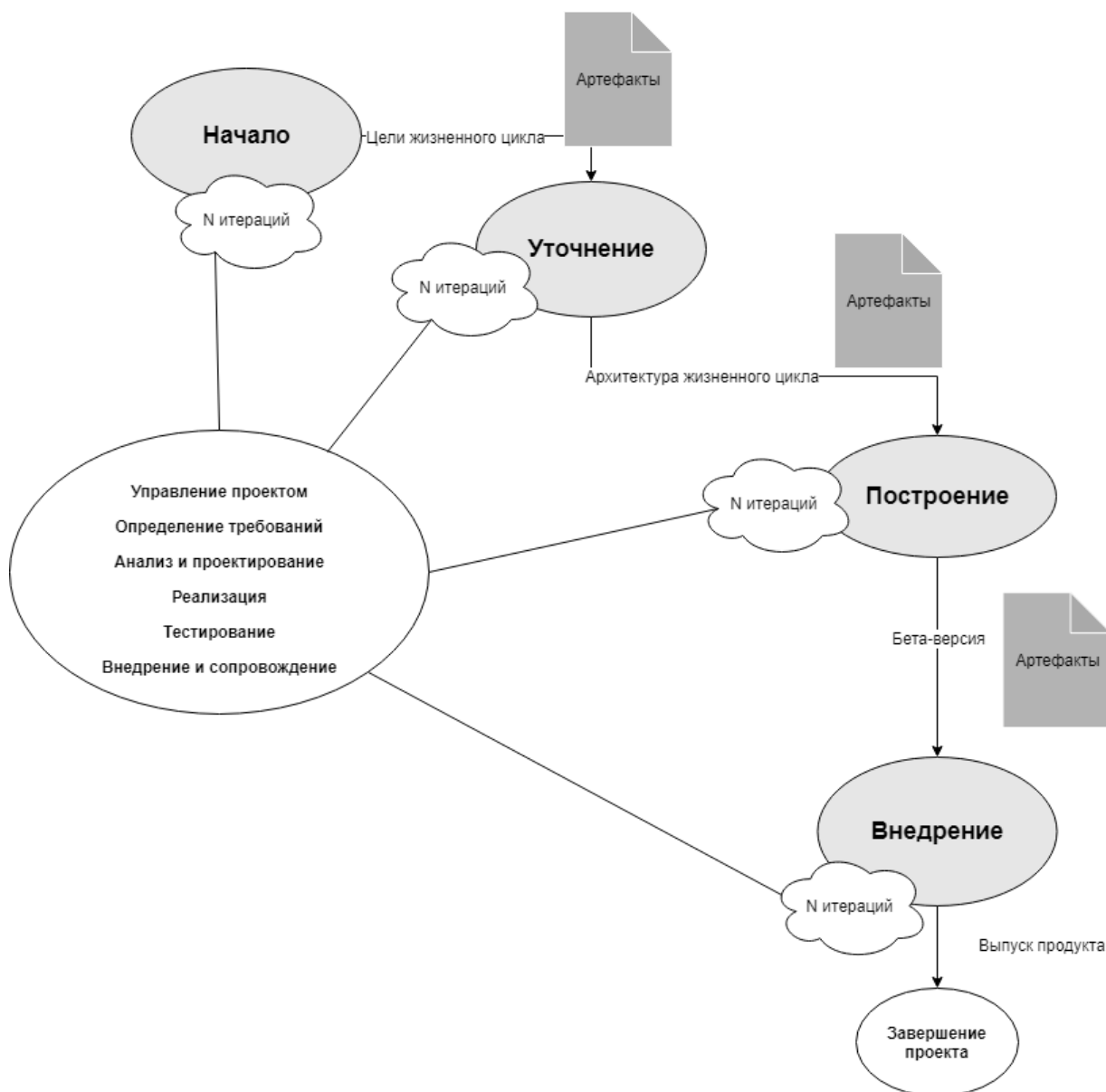


Рисунок 2.3.1 – Структура процесса адаптации

Как уже было сказано выше в данной работе будет использован вспомогательный рабочий процесс «Управление проектом».

Фрагмент таблицы А.4 – Потоки и деятельности работ

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Начало	Управление проектом	Разработать концепцию проекта	<ul style="list-style-type: none"> <li>- Идентифицировать и назначить риски</li> <li>- Разработать бизнес-кейс</li> <li>- Начать проект</li> </ul>
		Оценить масштаб и риск проекта	<ul style="list-style-type: none"> <li>- Идентифицировать и назначить риски</li> <li>- Разработать бизнес-кейс</li> </ul>
		Реализовать план разработки проекта	<ul style="list-style-type: none"> <li>- Разработать план метрик</li> <li>- Разработать план обеспечения качества</li> <li>- Разработать план по риск менеджменту</li> <li>- Определить организационную структуру и команду</li> <li>- Определить процессы мониторинга и контроля</li> <li>- Разработать план по принятию продукта</li> <li>- План фаз и итераций</li> <li>- Разработать план решения проблем</li> <li>- Составить план разработки программного обеспечения</li> </ul>

В таблице А.4 представлены в качестве примера несколько потоков вспомогательного рабочего процесса «Управление проектом» для фазы «Начало». Полностью таблица представлена в Приложении А.

Таблица 1 – Пример набора артефактов для фазы «Начало»

Деятельность	Входящие артефакты	Исходящие артефакты	Исполнитель
Идентифицировать и назначить риски	Видение продукта	Список рисков	Менеджер проектов
Разработать бизнес-кейс	Видение продукта, Список рисков	Бизнес-кейс	Менеджер проектов
Начать проект	Видение продукта, Список рисков, Бизнес-кейс	План разработки План итерации	Менеджер проектов
Идентифицировать и назначить риски	Видение продукта, Список рисков	Список рисков	Менеджер проектов
Разработать план метрик	-	План метрик	Менеджер проектов
Разработать план обеспечения качества	-	План обеспечения качества	Менеджер проектов

Продолжение таблицы 1 – Пример набора артефактов для фазы «Начало»

Деятельность	Входящие артефакты	Исходящие артефакты	Исполнитель
Разработать план по риск менеджменту	-	План управления рисками	Менеджер проектов
Определить организационную структуру и команду	Видение продукта Список рисков Бизнес-кейс	План разработки	Менеджер проектов
Определить процессы мониторинга и контроля	План управления рисками	План метрик План разработки	Менеджер проектов
Разработать план по принятию продукта	-	План по принятию продукта	Менеджер проектов
План фаз и итераций	План разработки	План разработки	Менеджер проектов
Разработать план решения проблем	-	План решения проблем	Менеджер проектов
Составить план разработки программного обеспечения	-	План разработки	Менеджер проектов

Продолжение таблицы 1 – Пример набора артефактов для фазы «Начало»

Деятельность	Входящие артефакты	Исходящие артефакты	Исполнитель
Разработать план итерации	Видение продукта Список рисков Бизнес-кейс План разработки	План итерации	Менеджер проектов
Разработать бизнес-кейс	Видение продукта Список рисков	Бизнес-кейс	Менеджер проектов
Выделить команду	План итерации План разработки	План итерации План разработки	Менеджер проектов
Начать итерацию	План итерации План разработки	Рабочее задание	Менеджер проектов
Оценить итерацию	Видение продукта Список ошибок План метрик Бизнес-кейс Тест-план Оценка состояния План итерации План разработки	Оценка итерации Запрос изменений	Менеджер проектов Тестировщик
Составить график работ и назначить исполнителей	Запрос изменений План разработки План итерации	Рабочее задание	Менеджер проектов
Мониторить статус проекта	Список ошибок План управления рисками План метрик	Список рисков	Менеджер проектов
Готовить отчет по статусу	Список ошибок	Оценка состояния	Менеджер проектов

Продолжение таблицы 1 – Пример набора артефактов для фазы «Начало»

Деятельность	Входящие артефакты	Исходящие артефакты	Исполнитель
Обработать исключения и проблемы	Оценка состояния План решения проблем Список ошибок План итерации План разработки	Рабочее задание Запрос изменений Список ошибок План итерации План разработки	Менеджер проектов
Идентифицировать и оценить риски	Список рисков Видение продукта	Список рисков	Менеджер проектов
Разработать бизнес-кейс	Список рисков	Бизнес-кейс	Менеджер проектов
Разработать план итерации	Список рисков Видение продукта План разработки	План итерации	Менеджер проектов
Разработать бизнес-кейс	Список рисков Видение продукта	Бизнес-кейс	Менеджер проектов
Разработать план метрик	-	План метрик	Менеджер проектов
Разработать план обеспечения качества	-	План обеспечения качества	Менеджер проектов
Разработать план по риск менеджменту	-	План управления рисками	Менеджер проектов
Определить организационную структуру и команду	Видение продукта Список рисков Бизнес-кейс	План разработки	Менеджер проектов

Окончание таблицы 1 – Пример набора артефактов для фазы «Начало»

Деятельность	Входящие артефакты	Исходящие артефакты	Исполнитель
Определить процессы мониторинга и контроля	План управления рисками	План метрик План разработки	Менеджер проектов
Разработать план по принятию продукта	-	План по принятию продукта	Менеджер проектов
План фаз и итераций	План разработки	План разработки	Менеджер проектов
Разработать план решения проблем	-	План решения проблем	Менеджер проектов
Составить план разработки программного обеспечения	-	План разработки	Менеджер проектов

В таблице 1 представлен пример табличной структуры для фазы «Начало» рабочего процесса «Управление проектами» показывающий какой набор деятельностей предусмотрен для потока и какие входящие и исходящие артефактами связаны с данной деятельностью ней. Для каждой деятельности указан в том числе и исполнитель.

Для лучшего представления списка потоков работ были разработаны диаграммы для каждого рабочего процесса по всем фазам унифицированного процесса.

На рисунках 2.3.2, 2.3.3, 2.3.4, 2.3.5, 2.3.6, 2.3.7 представлены диаграммы активностей потоков работ для следующих рабочих процессов унифицированного процесса:

- Рабочий процесс «Управление проектами».
- Рабочий процесс «Определение требований».



- Рабочий процесс «Анализ и проектирование».
- Рабочий процесс «Реализация».
- Рабочий процесс «Тестирование».
- Рабочий процесс «Развертывание».

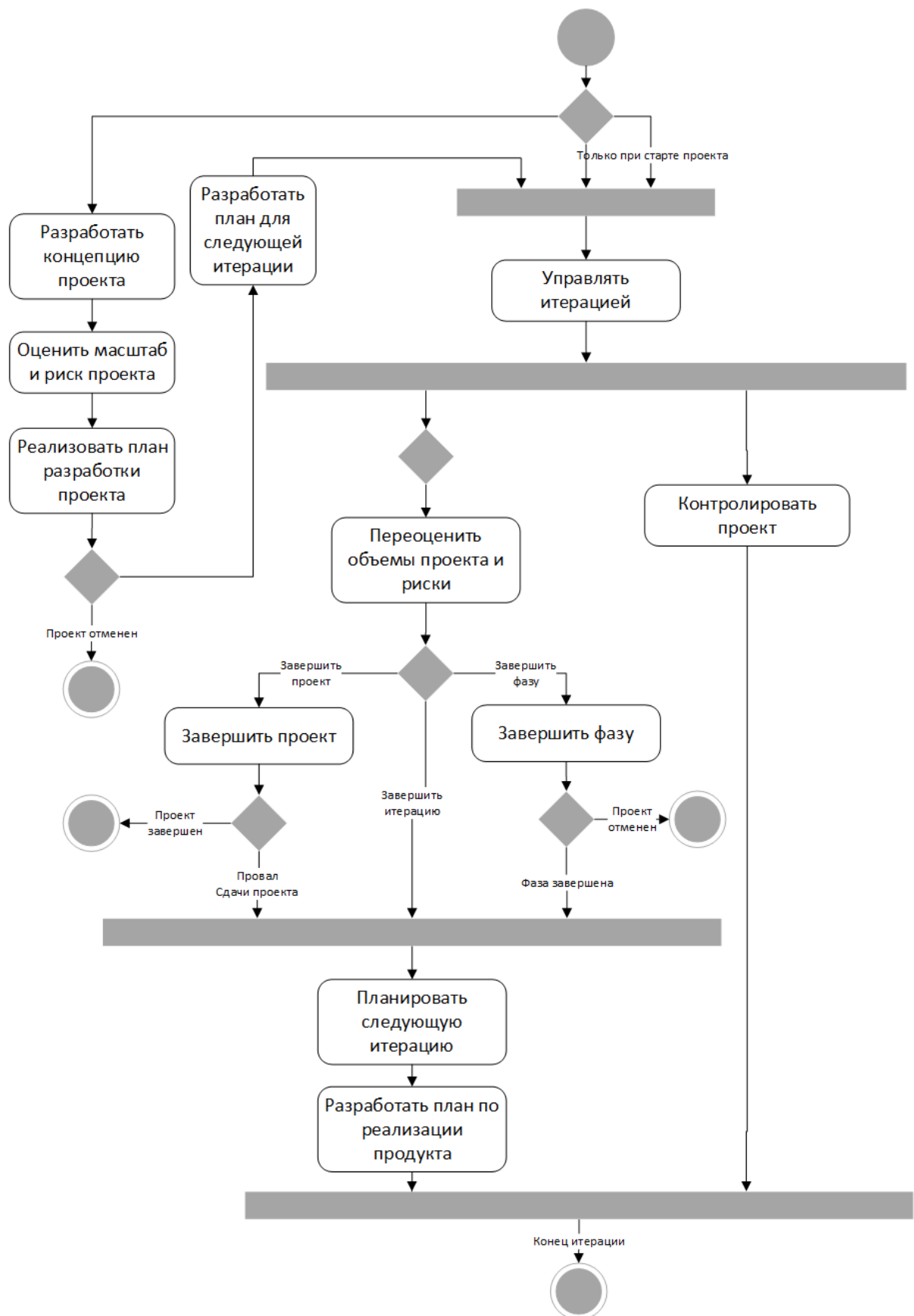


Рисунок 2.3.2 – Поток работ вспомогательного рабочего процесса «Управление проектами» для всех фаз унифицированного процесса

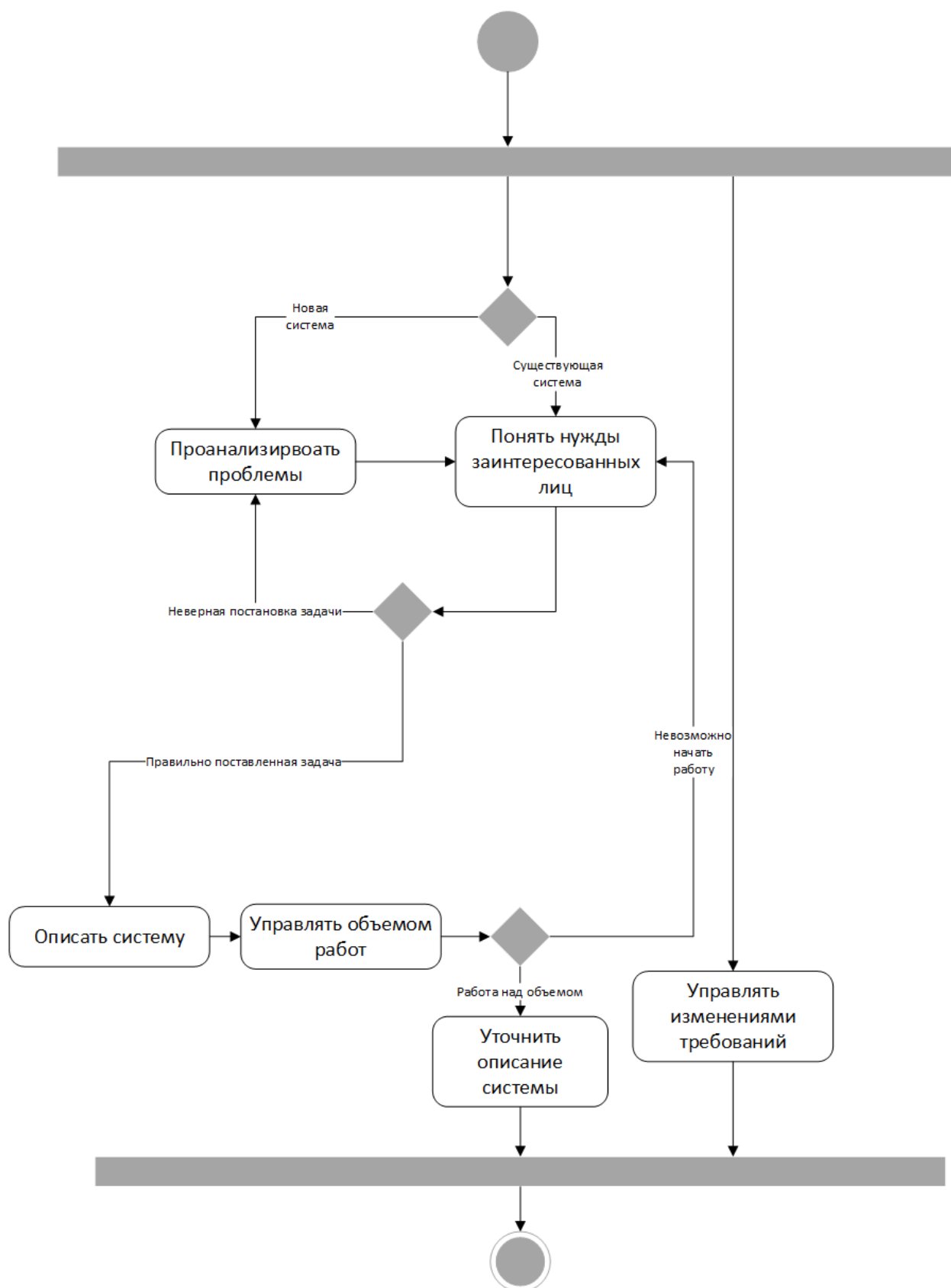


Рисунок 2.3.3 – Поток работы рабочего процесса «Определение требований» для всех фаз унифицированного процесса

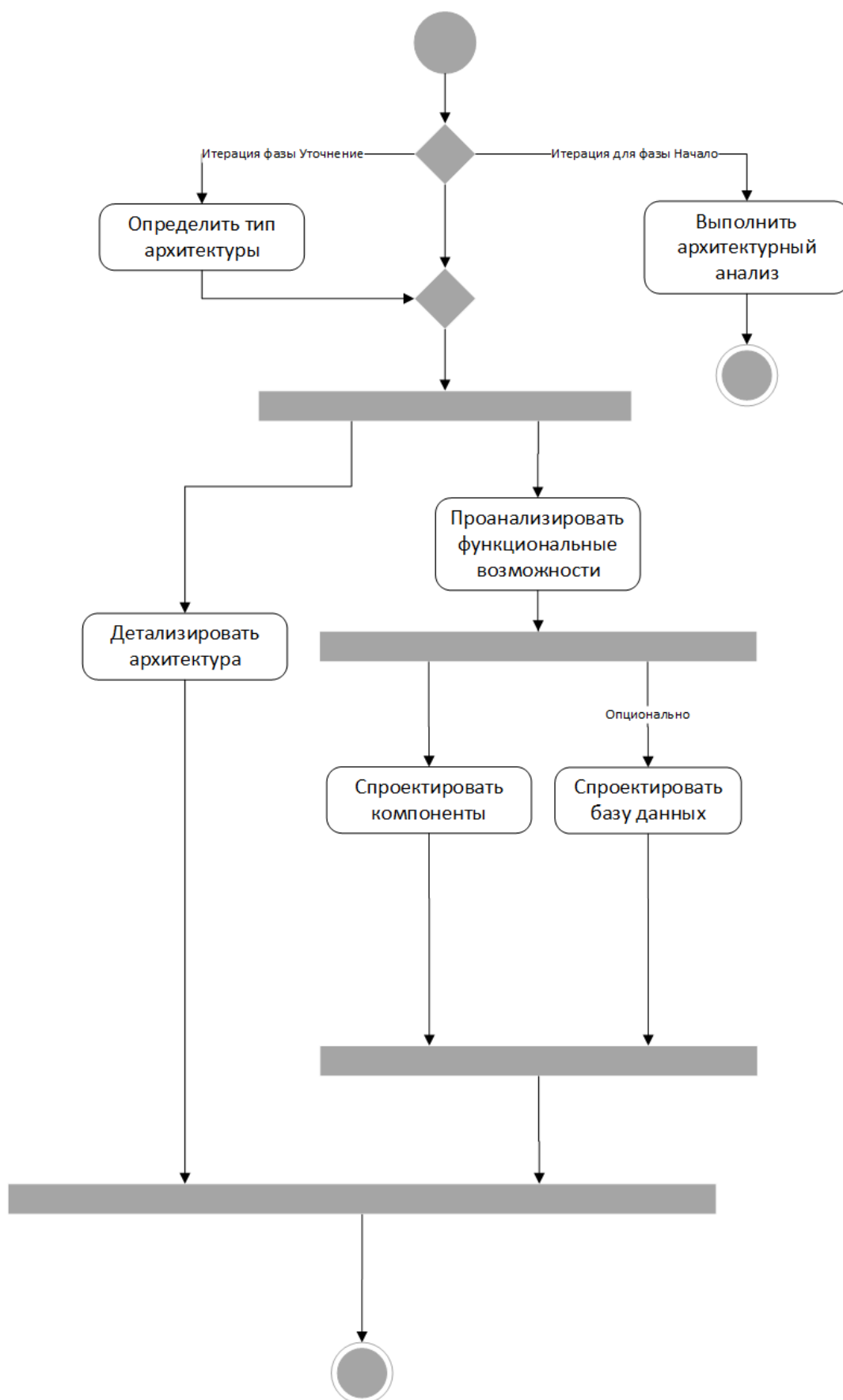
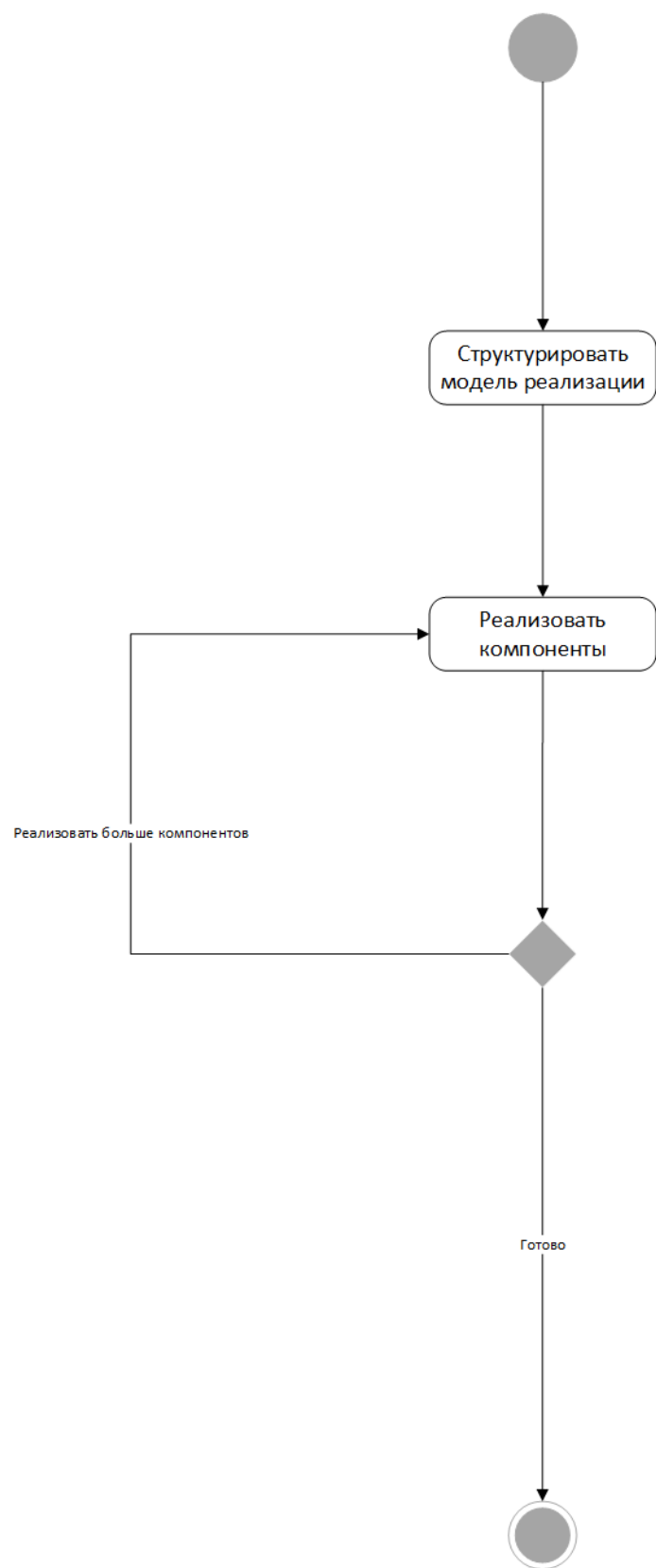
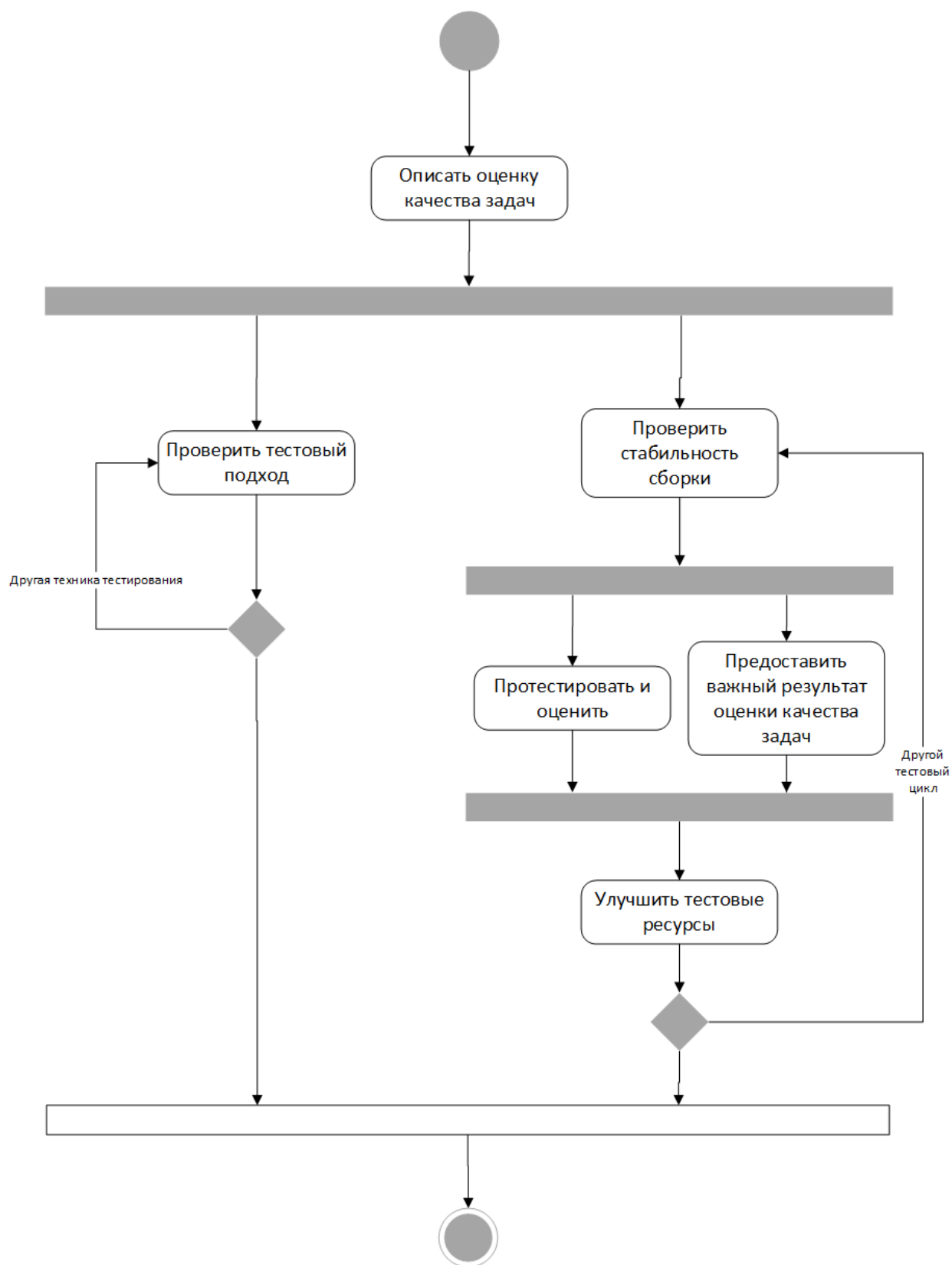


Рисунок 2.3.4 – Поток работ рабочего процесса «Анализ и проектирование» для всех фаз унифицированного процесса



*Рисунок 2.3.5 – Поток работ рабочего процесса «Реализация» для всех фаз унифицированного процесса*



*Рисунок 2.3.6 – Поток работ рабочего процесса «Тестирование» для всех фаз унифицированного процесса*

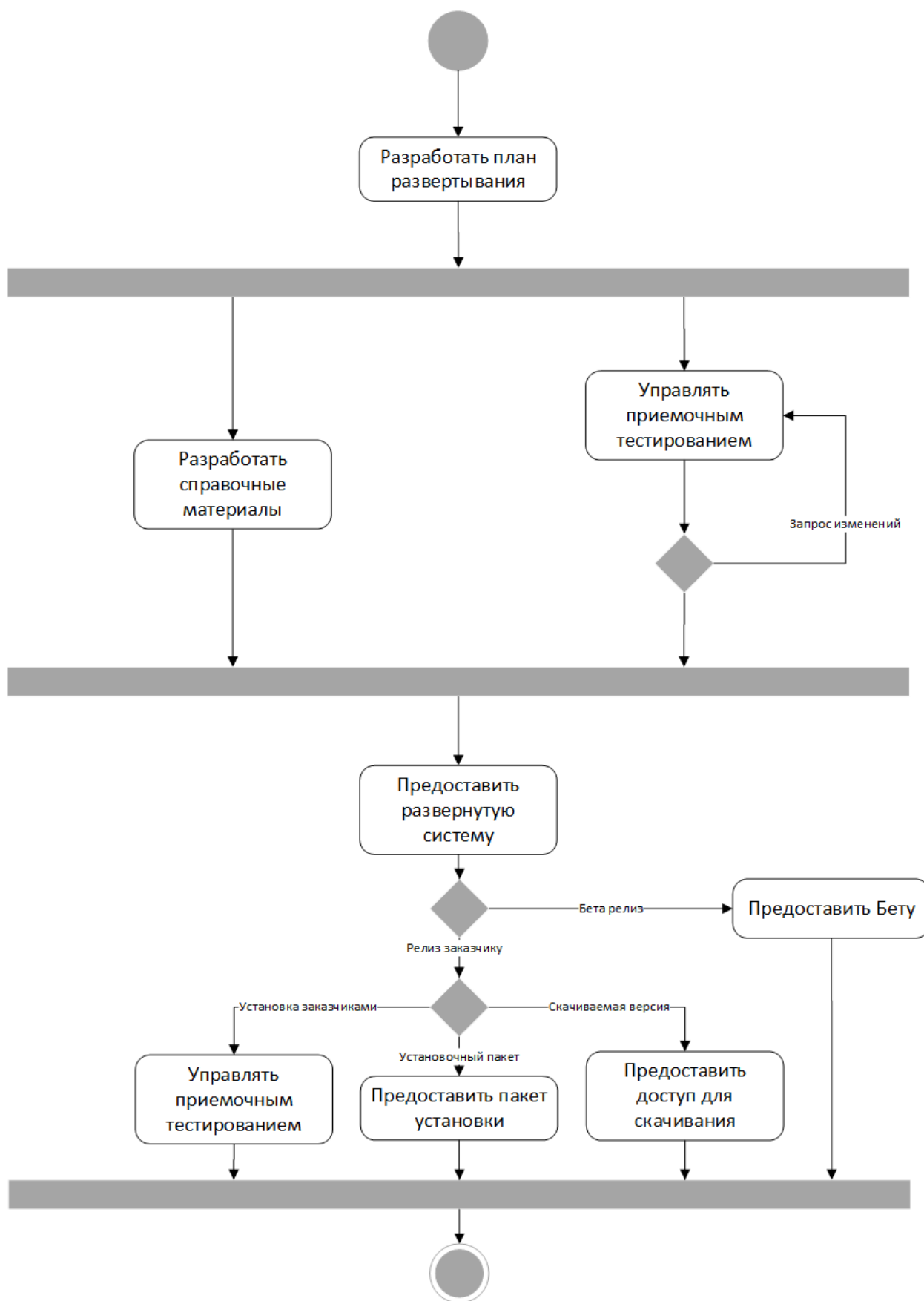


Рисунок 2.3.7 – Поток работы рабочего процесса «Внедрение и сопровождение» для всех фаз унифицированного процесса

Схема формального описания системы адаптации представлена на диаграмме ниже, см. рисунок 2.3.8. Схема учитывает не только новые проекты, но и существующие проекты с возможностью запуска новой итерации для одной и той же фазы, пока не будет удовлетворен набор условий выхода из фазы.

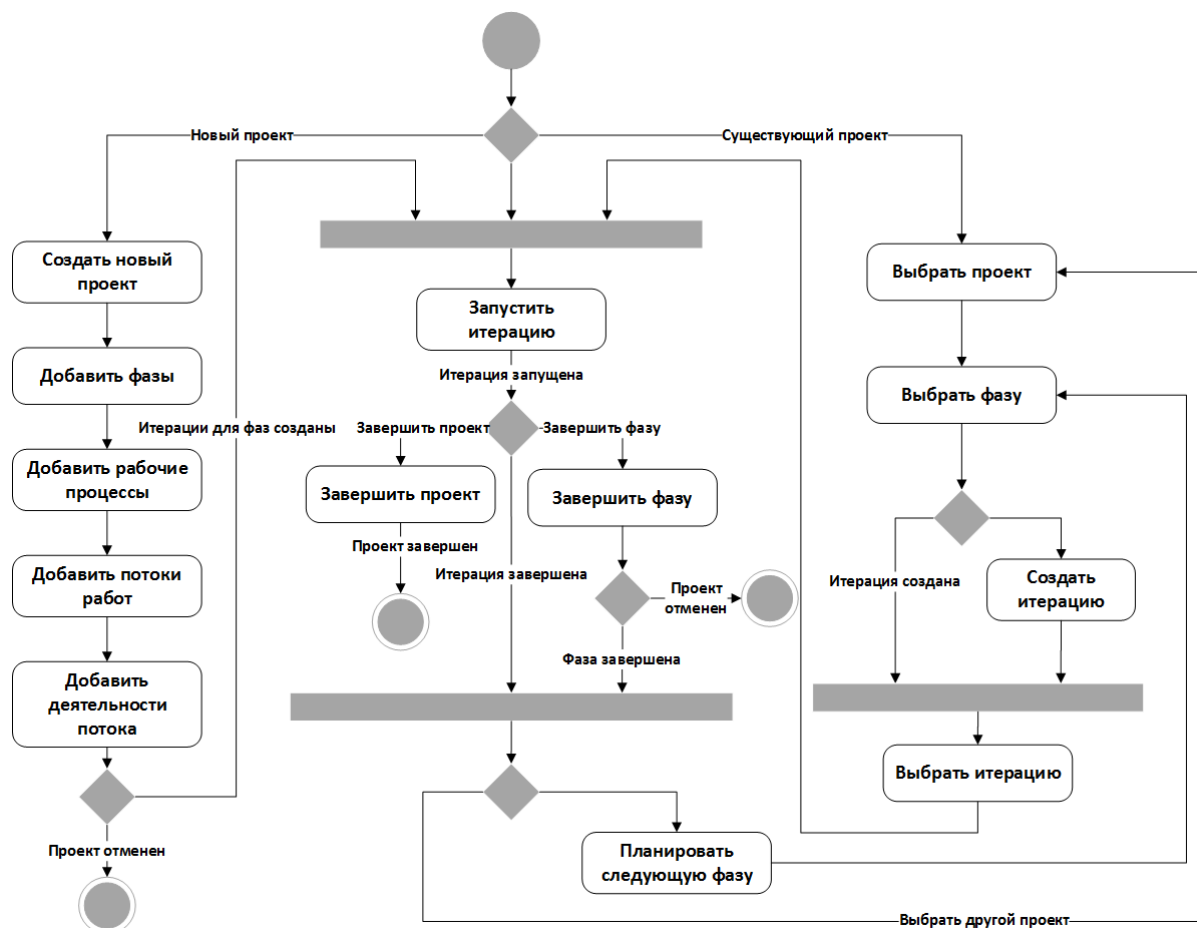


Рисунок 2.3.8 – Схема формального описания системы адаптации UP к требованиям конкретного проекта

Под адаптацией понимается выбор необходимых фаз, рабочих процессов и списка деятельностей для проекта любого типа и сложности. Процессы настраиваются на уровне фаз проекта и управляются итерациями.

Применение всех процессов и потоков работ скорее всего приведет к усложнению процессов, что однозначно негативно повлияет на проект. В результате адаптации UP под нужды проекта менеджер проектов получает настроенный экземпляр процессов, своего рода план со списком последовательных операций.



### **3. Разработка системы адаптации**

#### **3.1 Определение требований**

Как было уже сказано выше, в качестве базовой методологии для реализации системы адаптации, мы используем унифицированный процесс и базу знаний RUP.

Сбор и обработка требований к системе адаптации накапливались в процессе анализа методологий UP и RUP, а также в ходе работы над реальными производственными проектами. В результате были сформулированы следующие функциональные требования к системе:

- Пользователь должен иметь возможность создать и настроить проект.
- Пользователь должен иметь возможность создать итерацию для созданного проекта.
- Пользователь должен иметь возможность менять состояние итерации.
- Пользователь должен иметь возможность просмотреть детали проекта.
- Пользователь должен иметь возможность поиска проектов по дате и по названию.
- Система должна создать итерации для выбранных фаз нового проекта автоматически.
- Система должна обеспечить экспорт данных в файл формата CSV.
- Система должна сгенерировать схему работ (список деятельности) по созданному проекту.

При создании проекта пользователь должен иметь возможность указать название проекта, дату начала, дату окончания, краткое описание, выбрать фазы, рабочие процессы, потоки.

Под возможностью «настроить проект» понимается указание деятельности и добавление команды.

При создании проекта система создает итерации для каждой фазы автоматически и назначает статус «Не начата».

В рамках функциональной возможности «просмотреть детали проекта» понимается просмотр созданных фаз и создание новых итераций, список команды и артефактов проекта. Кроме того должен быть доступен статус состояния фазы, итерации и артефакта. Фаза и итерация имеют три статуса: не начата, в процессе, завершена.

Выбор CSV формата обусловлен простотой организации хранения данных, что дает возможность реализовать функции экспорта сравнительно быстро. Кроме того файл формата CSV можно импортировать в приложение Microsoft Project.

Основное требование к нотации для сгенерированной схемы проекта адаптации – она должна быть интуитивно понятной. Области фаз и рабочих процессов должны быть визуально выделены, должна быть определена последовательность потоков и деятельности, с указанием входящих и исходящих артефактов.

Также были определены нефункциональные требования к системе:

- Система должна быть предназначена для менеджеров проектов.
- Система не должна требовать установки конечным пользователем.
- Система должна хранить данные локально.

### **3.2 Анализ и проектирование системы**

Единственный актер системы – это менеджер проектов, ему доступен весь описанный выше функционал, для данного актера были формализованы функциональные возможности системы и описаны с помощью диаграммы вариантов использования. Диаграмма представлена на рисунке 3.2.1.

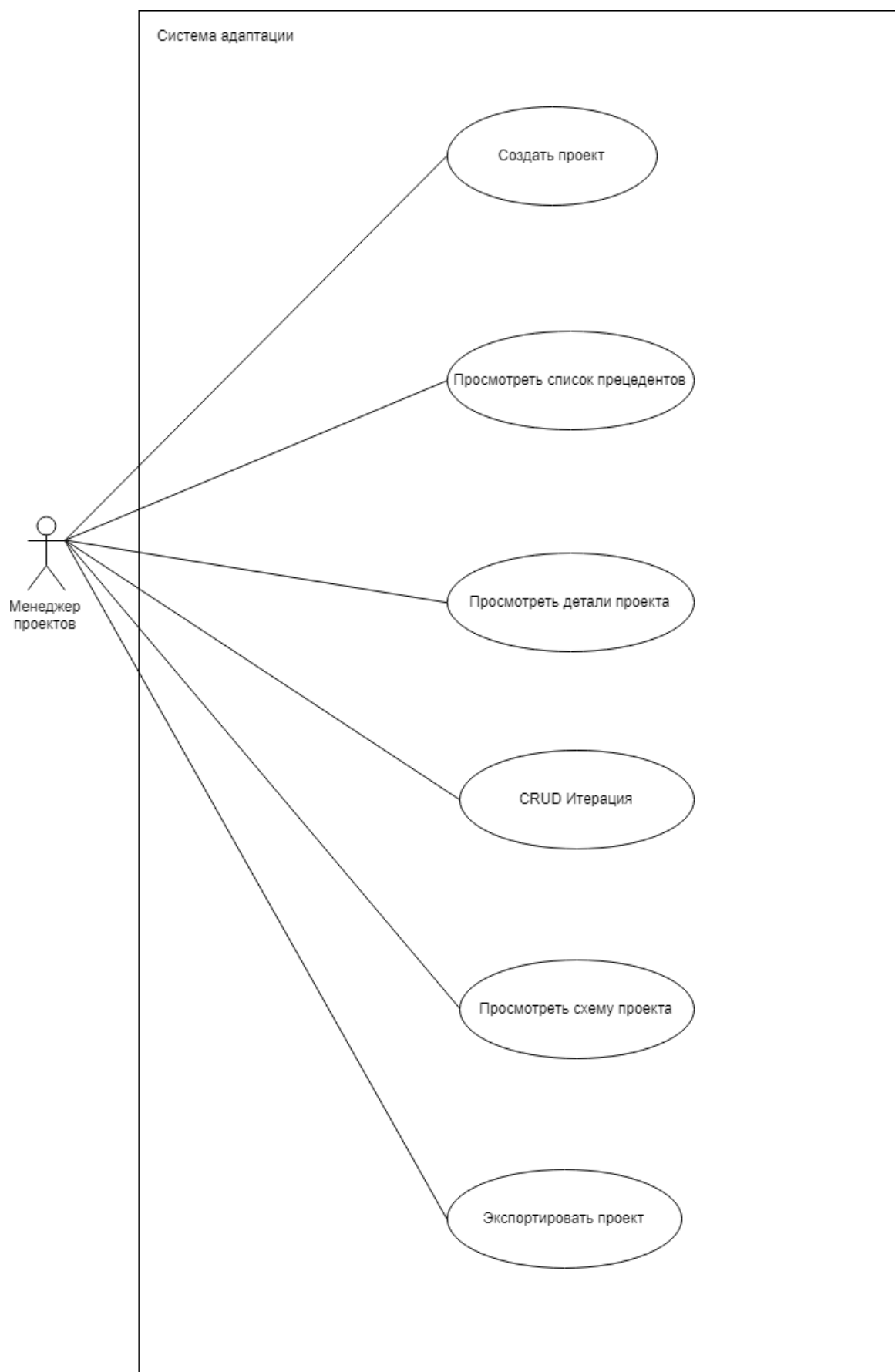


Рисунок 3.2.1 – Диаграмма варианта использования актера «Менеджер проектов»

На основе сформулированных требований к системе было принято решение реализовать клиент-серверное веб-приложение. Проектирование эскизов интерфейса (UX) было выполнено с помощью онлайн сервиса [app.moqups.com](http://app.moqups.com). Всего было подготовлено девятнадцать страниц эскизов интерфейса для системы адаптации. На рисунке 3.2.2 представлен эскиз интерфейса для вкладки «Детали проекта».

The sketch shows a web browser window titled 'Mozilla' with the address bar displaying 'http://moqups.com'. The page has a navigation bar with tabs: 'Детали проекта' (selected), 'Фазы проекта', 'Итерации', 'Команда', and 'Артефакты'. A link 'Вернуться к списку проектов' is in the top right. The main content area is divided into two columns. The left column contains: 'Название проекта' with the value 'Сайт визитка'; 'Дата начала проекта' (4/22/2012) and 'Дата окончания проекта' (4/22/2012) with calendar icons; a 'Статус проекта' dropdown menu with a callout box listing 'Статусы: в разработке, выполнен, отменен'; a 'Начало' dropdown menu with a callout box listing 'Начало, уточнение, конструирование, переход'; and a list of checkboxes for project phases: 'Формирование требований', 'Анализ требований', 'Проектирование', 'Реализация', 'Тестирование', and 'Внедрение и сопровождение'. A callout box for 'Проектирование' says 'Для каждой фазы, свои процессы'. The right column contains 'Описание проекта' with the text 'Проект реализуется для ювелира'. A 'Сохранить' button is at the bottom right.

*Рисунок 3.2.2- Эскиз интерфейса «Детали проекта»*

Для хранения структуры унифицированного процесса был выбран JSON формат. На рисунке 3.2.3 представлен фрагмент структуры UP в формате JSON.

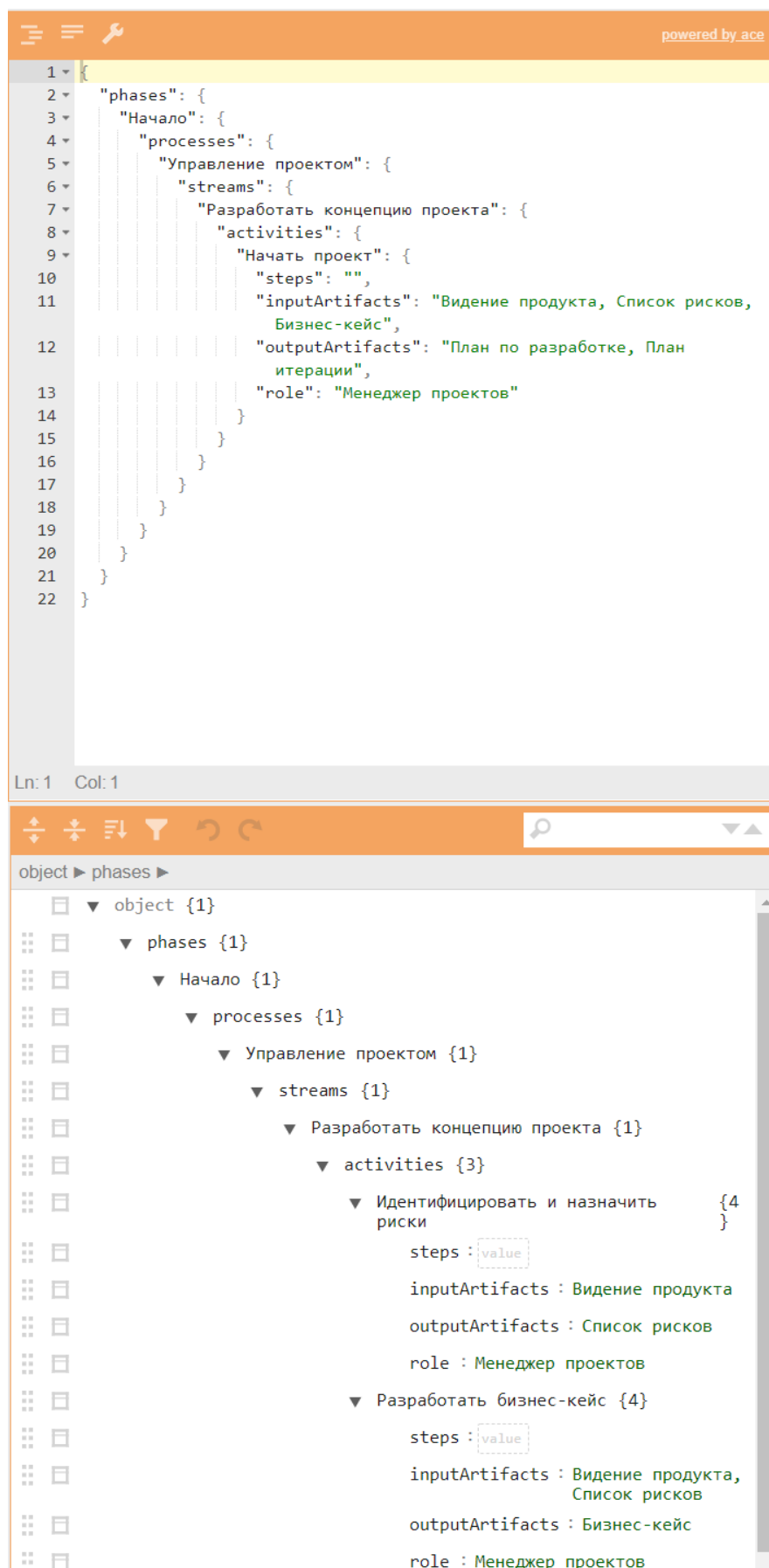


Рисунок 3.2.3 – Фрагмент структуры унифицированного процесса в формате JSON

Данные записаны в виде вложенных объектов, внутри объектов хранится набор пар ключ/значение. Например, объект «Начать проект» содержит набор ключей и соответствующие значения к ним:

- *steps* – набор условия для прохождения контрольной точки фазы;
- *inputArtifacts* – входящие артефакты, необходимые ресурсы для осуществления деятельности и создания новых артефактов;
- *outputArtifacts* – исходящие артефакты, поставляемые результаты;
- *role* – исполнитель, сотрудник ответственный за деятельность и ее результаты.

Для работы по форматированию JSON формата был применен веб-инструмент JSON Editor Online.

В качестве базовых полей для экспорта были определены следующие данные:

- Название итерации с вложенными рабочими процессами.
- Название потока.
- Название деятельности с соблюдением иерархии.
- Дата начала проекта.
- Дата окончания проекта.
- Длительность в днях.
- Входящие артефакты.
- Исходящие артефакты.

### 3.3 Реализация системы

Для начала реализации были подготовлены следующие необходимые ресурсы:

- UX – эскизы интерфейса.
- Список функциональных и нефункциональных требований.
- Структура процесса адаптации.
- Выбраны инструменты разработки.

Для работы с исходным кодом был использован GitHub - веб-сервис для хостинга поддерживающий контроль версий.

Дизайн системы был реализован с помощью каскадных таблиц стилей (CSS) на основе эскизов интерфейса (UX).

Для локального хранения данных был выбран Web Storage – API ориентированный на обеспечение простого способа хранения данных пользователя на стороне клиента. Технология очень мощная, быстрая и безопасная. Web Storage предоставляет два типа объекта для хранения данных: **Local Storage** и **Session Storage**. Первый тип хранит данные даже после закрытия браузера. Второй тип хранит данные только пока открыта сессия. Для каждого домена браузер создает свой объект Local Storage, соответственно редактировать или просматривать его можно только с этого домена. Для системы адаптации был использован Local Storage. Объем Local Storage составляет 5Mb, что является более чем достаточным для нужд будущего веб-приложения.

Перед тем как окончательно выбрать технологию Web Storage, с помощью онлайн сервиса «Can I use» был проанализирован список версий браузеров, которые умеют с ней работать. На рисунке 3.3.1 представлен довольно внушительный список версий браузеров, которые поддерживают работу с Local Storage.

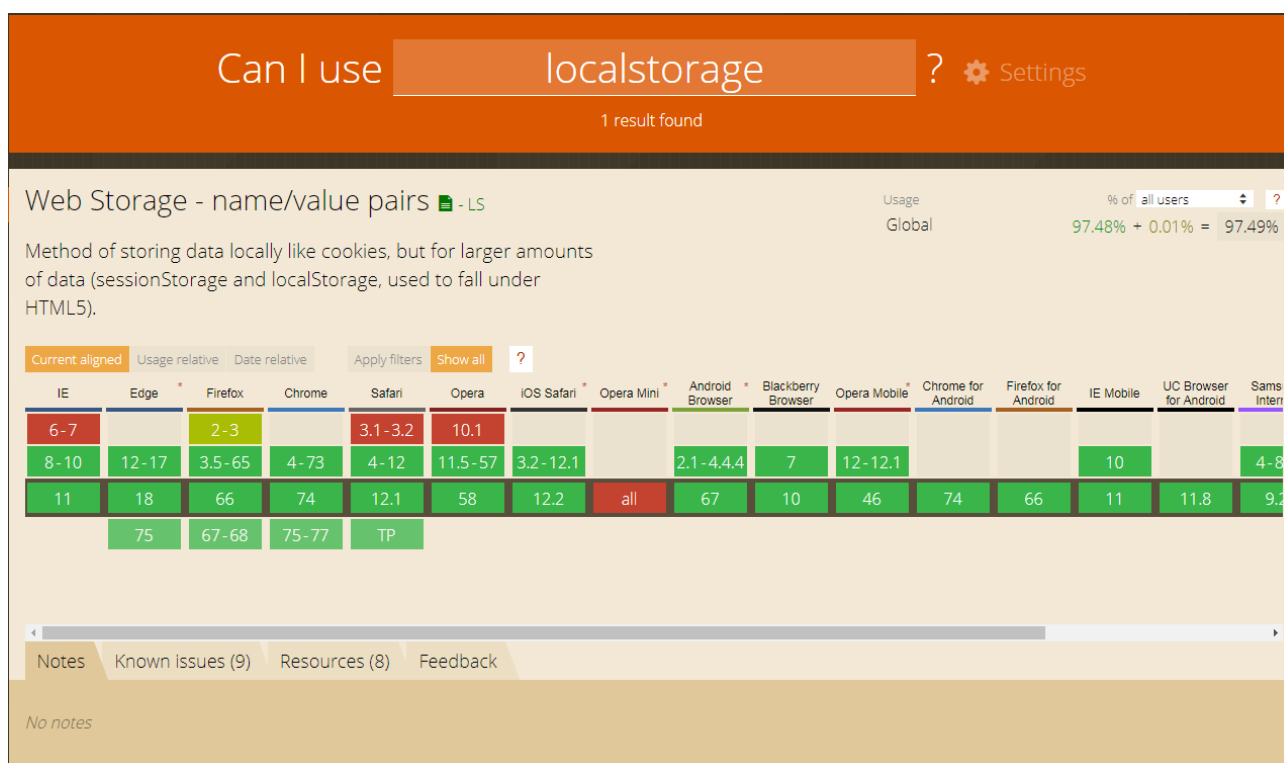


Рисунок 3.3.1 – Список версий браузеров поддерживающих WebStorage

В качестве интегрированной среды разработки был выбран WebStorm от компании JetBrains. Для сборки веб-приложения был использован статический модульный сборщик WebPack. Для локальной разработки был использован Node.js, который позволяет разрабатывать и отлаживать веб-приложение без использования браузера.

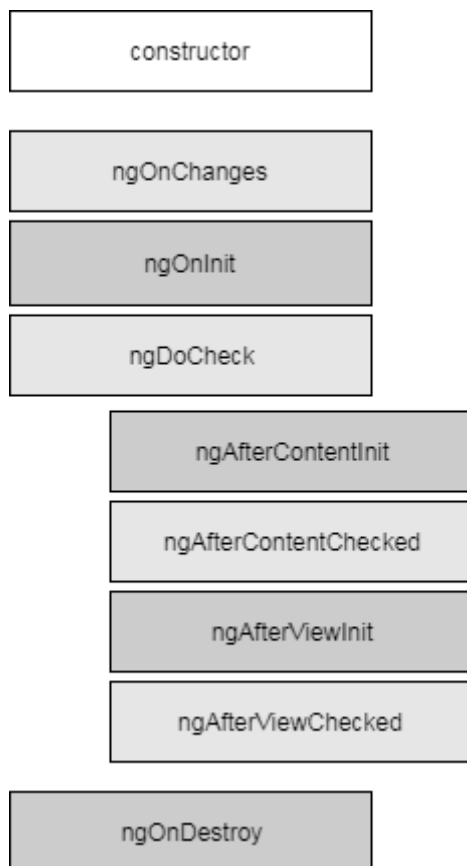
В качестве потенциальных фреймворков для будущей веб-системы были рассмотрены и проанализированы Angular, React, Redux. Анализ и оценка инструментов осуществлялась по следующим ключевым критериям: популярность, документированность, регулярные обновления версий. Фреймворк Angular был выбран как наиболее подходящий, по данным критериям, инструмент. Это разработка от компании Google, которая является на сегодняшний день очень популярной и хорошо документированной платформой и, что немаловажно, регулярно обновляемой. Фреймворк Angular поддерживает решения SPA (Single Page Application), т.е. ориентирован на поддержку одностраничных приложений и является наследником AngularJS. Ключевой особенностью Angular является использование языка программирования TypeScript.



На архитектуру выбранного фреймворка Angular стоит обратить особое внимание, так как она диктует правила и навязывает свои принципы при разработке веб-приложений.

Angular приложение состоит из нескольких модулей. Корневой модуль приложения называется AppModule, он обязателен и представлен классом. Корневой модуль используется в качестве точки входа. Внутри модуля объявляются содержащиеся в нем компоненты. Компонент отвечает за отображение представления на экране. Шаблон (Template) представляет собой кусок разметки HTML с вставками кода Angular. Шаблон – это представление, которое увидит пользователь при работе с приложением.

После создания компонента, Angular вызывает у этого компонента ряд методов, которые представляют различные этапы жизненного цикла, наглядно представлено на рисунке 3.3.2:



*Рисунок 3.3.2 – Жизненный цикл компонента Angular*

- **ngOnChanges**: вызывается до метода `ngOnInit()` при начальной установке свойств, которые связаны механизмом привязки, а также при любой их переустановки или изменении их значений. Данный метод в качестве параметра принимает объект класса `SimpleChanges`, который содержит предыдущие и текущие значения свойства.

- **ngOnInit**: вызывается один раз после установки свойств компонента, которые участвуют в привязке, и выполняет инициализацию компонента.

- **ngDoCheck**: вызывается при каждой проверке изменений свойств компонента сразу после методов `ngOnChanges` и `ngOnInit`.

- **ngAfterContentInit**: вызывается один раз после метода `ngDoCheck()` после вставки содержимого в представление компонента кода HTML.

- **ngAfterContentChecked**: вызывается фреймворком Angular при проверке изменений содержимого, которое добавляется в представление компонента. Вызывается после метода `ngAfterContentInit()` и после каждого последующего вызова метода `ngDoCheck()`.

- **ngAfterViewInit**: вызывается фреймворком Angular после инициализации представления компонента, а также представлений дочерних компонентов. Вызывается только один раз, сразу после первого вызова метода `ngAfterContentChecked()`.

- **ngAfterViewChecked**: вызывается фреймворком Angular после проверки на изменения в представлении компонента, а также проверки представлений дочерних компонентов. Вызывается после первого вызова метода `ngAfterViewInit()` и после каждого последующего вызова `ngAfterContentChecked()`.

- **ngOnDestroy**: вызывается перед тем, как фреймворк Angular удалит компонент.

Angular поддерживает механизмы привязки, таким образом различные части шаблона могут быть привязаны к некоторым значениям, определенным в компоненте. Каждый компонент имеет свой шаблон. Метаданные в Angular используются для оформления класса, задавая его ожидаемое поведение.

За работу с данными в Angular отвечают сервисы. Сервисы не работают напрямую с представлениями, т.е. с HTML разметкой. В задачи сервисов входит: предоставление данных приложению, предоставление канала взаимодействия между отдельными компонентами приложения и инкапсулирование бизнес-логики. Допустимо использование одного сервиса в другом, для этого необходимо применить к классу сервиса декоратор `Injectable`. Декораторы были предложены в качестве стандарта в ECMAScript 7 для изменения классов и свойств на этапе разработки.

Маршрутизация позволяет сопоставить запросы к приложению с определенными ресурсами внутри приложения. Для работы с маршрутизацией необходимо объявить базовый адрес приложения.

Работа с форматированием отображаемых значений осуществляется с помощью инструмента `pipes`, перечислим основные из них:

- `CurrencyPipe`: форматирование валюты.
- `PercentPipe`: форматирование процентов.
- `UpperCasePipe`: перевод строки в верхний регистр.
- `LowerCasePipe`: перевод строки в нижний регистр.
- `DatePipe`: форматирование даты.
- `DecimalPipe`: задание формата числа.
- `SlicePipe`: обрезание строки

На рисунке 3.3.3 представлена архитектура системы адаптации. Далее мы рассмотрим каждый модуль и его структуру по отдельности. Для проектирования архитектуры приложения была выбрана свободная нотация и использован онлайн сервис создания диаграмм и схем `draw.io`.

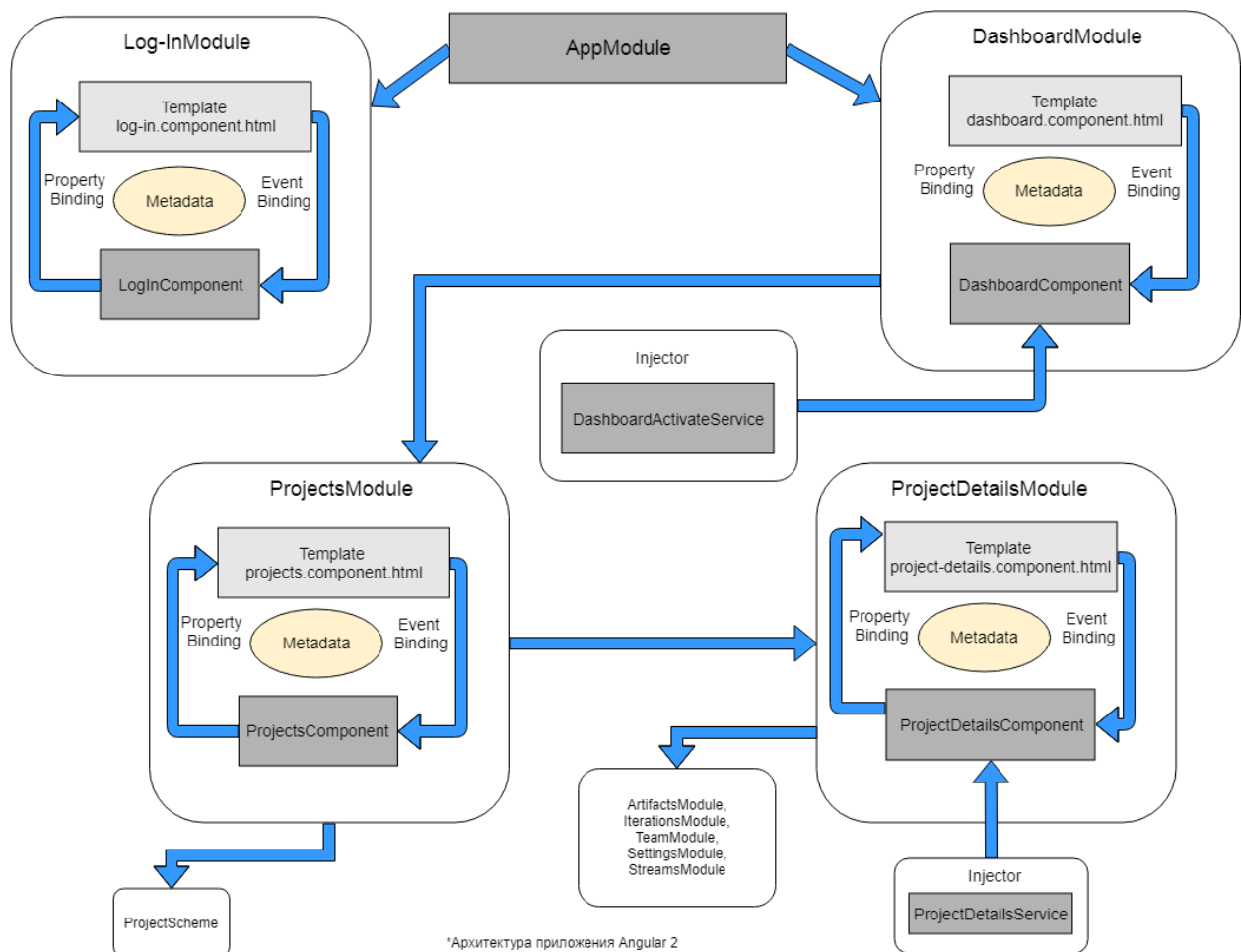


Рисунок 3.3.3 – Архитектура системы адаптации

**LogInModule** – модуль отвечает за авторизацию в приложении, содержит свой шаблон представления и компонент **LogInComponent**.

**DashboardModule** – модуль отвечает за работу с главной страницей веб-приложения, где отображается список проектов, содержит шаблон представления и компонент **DashboardComponent** и сервис работы с данными **DashboardActivateService**, который может вызывать другие сервисы.

**ProjectsModule** – модуль отвечает за работу со списком проектов, содержит компонент **ProjectsComponent** и шаблон представления.

**ProjectSchemeModule** – модуль отвечает за генерацию схемы адаптации проекта, содержит шаблон представления и компонент **ProjectSchemeComponent**.

**ProjectDetailsModule** – модуль отвечает за работу с деталями проекта и содержит компонент **ProjectDetailsComponent**, сервис **ProjectDetailsServer** и шаблон представления.

**ArtifactsModule** – модуль отвечает за работу с артефактами, содержит шаблон представления и компонент ArtifactsComponent.

**IterationsModule** – модуль отвечает за работу с итерациями, содержит шаблон представления и компонент IterationsComponent.

**TeamModule** – модуль отвечает за работу с данными команды, содержит шаблон представления и компонент TeamComponent.

**SettingsModule** – модуль отвечает за работу с настройками проекта, содержит компонент SettingsComponent и шаблон представления.

**StreamsModule** – модуль отвечает за работу с деятельностью рабочего процесса, содержит шаблон представления и компонент StreamsComponent.

Схема взаимодействия клиента и системы адаптации представлена на рисунке 3.3.4.

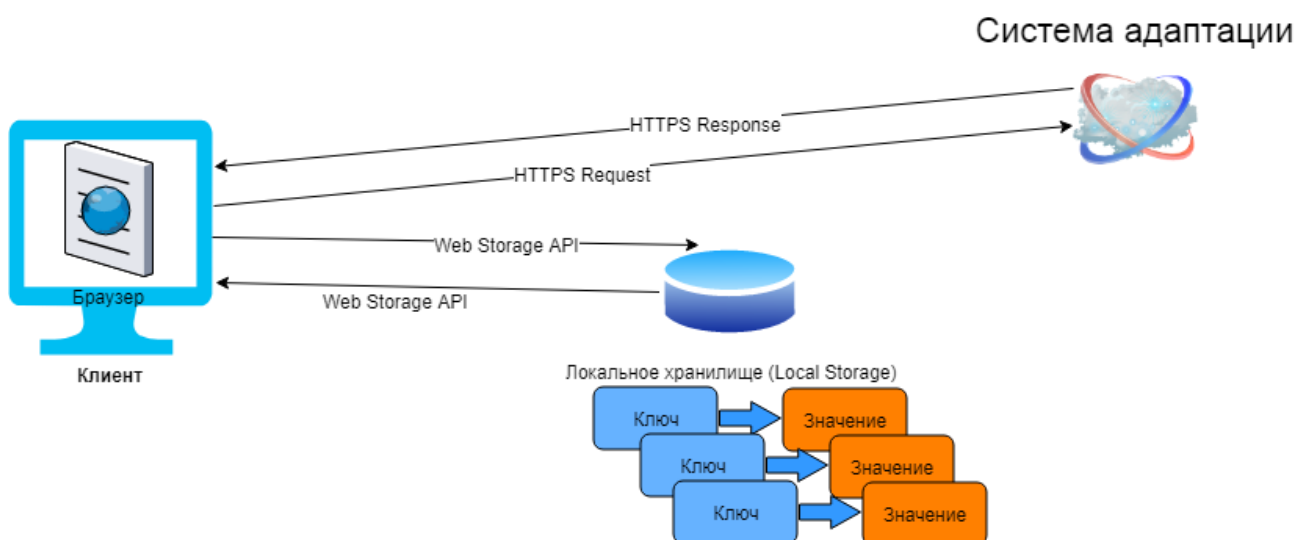


Рисунок 3.3.4 – Взаимодействие клиента и системы адаптации

### 3.4 Тестирование системы

План по тестированию системы был разработан в начале стадии реализации. Согласно плану были проведены следующие работы по тестированию:

- При выдаче на тестирование реализованного функционального требования, проводилось функциональное тестирование по методу «белого ящика» и тестирование удобства пользователя.
- При выдаче на тестирование версии с исправлениями не критичных дефектов, проводилось дымовое тестирование.
- При выдаче на тестирование версии с исправлениями критичных дефектов проводилось регрессионное тестирование.

Отчеты по тестированию формировались и хранились в Google Drive.

### 3.5 Развертывание и внедрение системы

Система адаптации развернута на личном веб-ресурсе [www.trushnikova/up](http://www.trushnikova/up) с помощью интерфейса командной строки Angular CLI, для обновления сервера используется команда **ng build –prod**.

Angular CLI позволяет создавать проекты, добавлять новые файлы в проект, выполняет работы по тестированию, сборке и развертыванию.

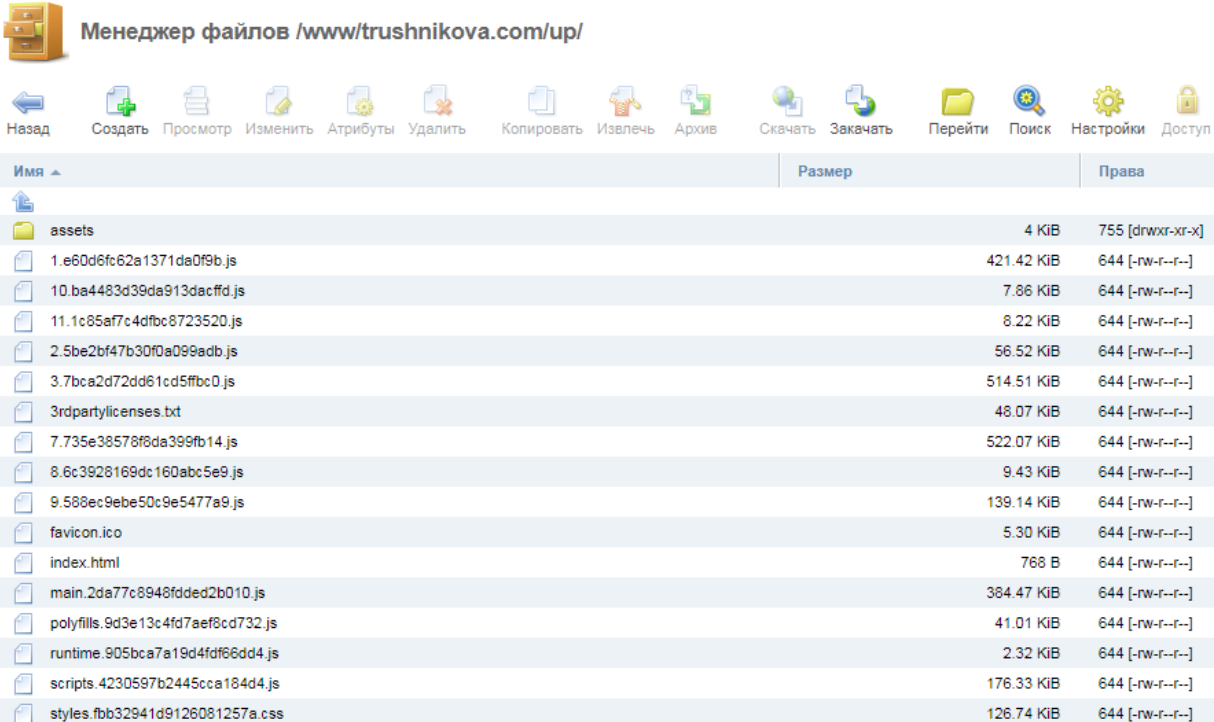
На рисунке 3.4.1 представлена структура папок исходного кода на GitHub. Для работы с GitHub использован интерпретатор командной строки для Windows.

..	
common	add logic for iteration filters
dashboard	delete title to phases
log-in	add some texts
app.component.html	create project
app.component.scss	add content for projects page
app.component.spec.ts	create project
app.component.ts	create project
app.module.ts	add logic for project form
app.routing.ts	add fake authorization
rxjs-extensions.ts	create project

Рисунок 3.4.1 – Файлы сборки проекта

На рисунке 3.4.2 представлен список файлов сборки размещенных на веб-ресурсе.

Доступ к системе адаптации на сегодняшний день является открыты, так как работа с данными ведется с использованием локального хранилища то нет необходимости ограничивать доступ авторизацией.



Менеджер файлов /www/trushnikova.com/up/

Имя	Размер	Права
assets	4 KiB	755 [drwxr-xr-x]
1.e60d6fc62a1371da0f9b.js	421.42 KiB	644 [-rw-r--r--]
10.ba4483d39da913dacfdd.js	7.86 KiB	644 [-rw-r--r--]
11.1c85af7c4dfbc8723520.js	8.22 KiB	644 [-rw-r--r--]
2.5be2bf47b30f0a099adb.js	56.52 KiB	644 [-rw-r--r--]
3.7bca2d72dd61cd5ffbc0.js	514.51 KiB	644 [-rw-r--r--]
3rdpartylicenses.txt	48.07 KiB	644 [-rw-r--r--]
7.735e38578f8da399fb14.js	522.07 KiB	644 [-rw-r--r--]
8.6c3928169dc160abc5e9.js	9.43 KiB	644 [-rw-r--r--]
9.588ec9ebe50c9e5477a9.js	139.14 KiB	644 [-rw-r--r--]
favicon.ico	5.30 KiB	644 [-rw-r--r--]
index.html	768 B	644 [-rw-r--r--]
main.2da77c8948fdded2b010.js	384.47 KiB	644 [-rw-r--r--]
polyfills.9d3e13c4fd7aef8cd732.js	41.01 KiB	644 [-rw-r--r--]
runtime.905bca7a19d4fdf68dd4.js	2.32 KiB	644 [-rw-r--r--]
scripts.4230597b2445cca184d4.js	176.33 KiB	644 [-rw-r--r--]
styles.fbb32941d9126081257a.css	126.74 KiB	644 [-rw-r--r--]

Рисунок 3.4.2 – Файлы сборки проекта

### **План по внедрению системы адаптации:**

- Реализация поддержки локализации на английском языке.
- Презентация системы адаптации ориентировочно назначена на конец июня 2019 года, которая пройдет в Китае.
- Доработка функционала контрольных точек выхода из фазы.
- Пилотный запуск системы адаптации на действующем проекте в августе 2019 года.

### **3.6 Описание реализованной системы адаптации**

В результате реализованы все заявленные функциональные требования к системе адаптации.

На рисунке 3.6.1 представлен скриншот реализованной системы со списком созданных проектов. Дополнительно реализован фильтр по названию проекта и по дате проекта. Фильтр по дате проекта работает по датам начала и окончания проекта включительно. На данной странице пользователю, доступны следующие функциональные возможности:

- Создание проекта.
- Просмотр списка проектов.
- Поиск проекта по фильтрам.
- Просмотр деталей проекта.
- Экспорт в файл формата CSV.
- Просмотр сгенерированной схемы по выбранному проекту.
- Выход из системы.



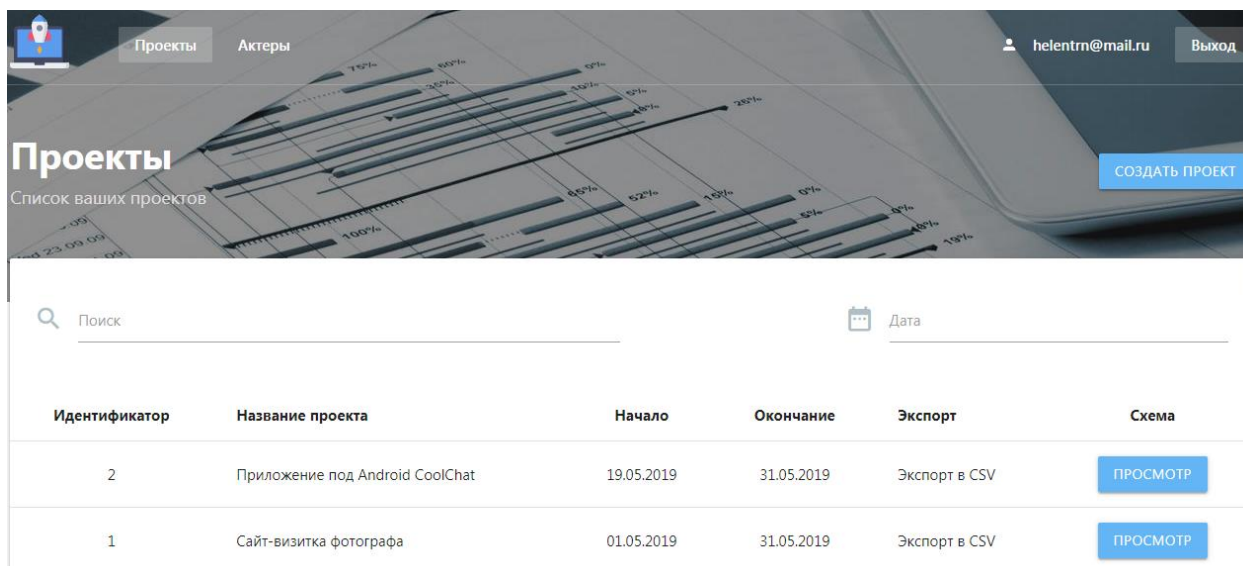


Рисунок 3.6.1 – Список созданных проектов

Функциональная возможность «Удалить проект» не планировалась изначально, так как существует возможность очистить Local Storage используя в браузере режим разработки, который активируется нажатием F12, далее выбирается домен и очищается, кроме того можно посмотреть по доменному имени как выглядят данные в Local Storage

```
[{name: "Test 17-05 21-30", startDate: "May 17, 2019", endDate: "May 24, 2019",...},...]
  1. 0: {name: "Test 17-05 21-30", startDate: "May 17, 2019", endDate: "May 24, 2019",...}
    1. description: "Тестирование продукта"
    2. endDate: "May 24, 2019"
    3. id: 2
    4. iterations: [{name: "-", startDate: "2019-05-16T17:00:00.000Z", endDate: "2019-05-30T17:00:00.000Z",...},...]
    5. name: "Test 17-05 21-30"
    6. phases: {values: ["Начало", "Уточнение"],...}
    7. startDate: "May 17, 2019"
    8. team: [{name: "Елена", role: "Менеджер проектов", isActive: false}]
  2. 1: {name: "Реализация системы адаптации унифицированного процесса к требованиям конкретного проекта",...}
    1. description: "Реализовать систему адаптации унифицированного процесса к требованиям конкретного проекта"
    2. endDate: "May 31, 2019"
    3. id: 1
    4. iterations: [{name: "-", startDate: "2019-05-12T17:00:00.000Z", endDate: "2019-05-26T17:00:00.000Z",...},...]
    5. name: "Реализация системы адаптации унифицированного процесса к требованиям конкретного проекта"
    6. phases: {values: ["Начало"],...}
    7. startDate: "May 13, 2019"
    8. team: []
```

Пример сгенерированной схемы адаптации по проекту представлен на Рисунке 3.6.2 На схеме указана последовательность потоков работ, деятельность потока и набор входящих и исходящих артефактов.

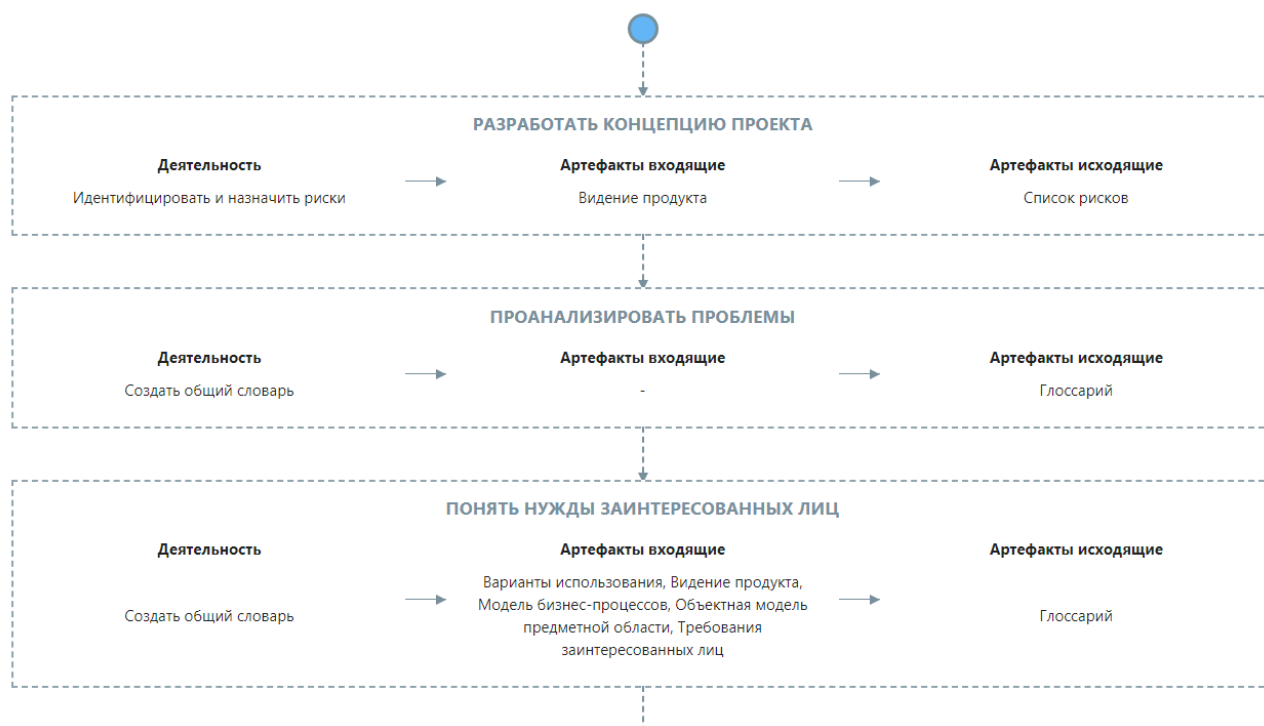


Рисунок 3.6.2 – Сгенерированная схема адаптации

Пример экспортированного файла из системы адаптации изображен на рисунке 3.6.3. При экспорте соблюдена иерархия вложенности, для этого система адаптации генерирует уровень для итерации, рабочего процесса, потоков и деятельности.

	1	2	3	4	5	6	7	8	9	1
1	Название	Уровень	Начало	Окончани	Длительн	Названия	Входящий	Исходящий	артефакт	
2	Итерация	1	Ср 10.01.1	Ср 15.05.1	351 дней					
3	Управлен	2	Ср 10.01.1	Пн 13.05.1	349 дней					
4	Разработа	3	Ср 10.01.1	Пн 13.05.1	349 дней					
5	Идентифи	4	Пн 13.05.1	Пн 13.05.1	1 день		Артефакт	Артефакт 3		
6	Разработа	4	Ср 10.01.1	Чт 11.01.1	2 дней		Артефакт	Артефакт 3		
7	Начать пр	4	Ср 10.01.1	Чт 11.01.1	2 дней		Артефакт	Артефакт 3		
8	Опреде	2	Вт 14.05.1	Ср 15.05.1	2 дней					
9	Проанали	3	Вт 14.05.1	Вт 14.05.1	0 дней		Артефакт	Артефакт 3		
10	Понять ну	3	Вт 14.05.1	Ср 15.05.1	2 дней		Артефакт	Артефакт 3		
11										
12										

Рисунок 3.6.3 – Экспортированный файл формата CSV

Для осуществления импорта данных из файла формата CSV необходимо настроить схему сопоставления данных в приложении Microsoft Project. На рисунке 3.6.4 представлена схема сопоставления данных. Схема создается один раз и хранится в Microsoft Project. Схема сопоставления для импорта экспортированного файла из системы адаптации должна обновляться только в случае изменений в реализации функции экспорта системы адаптации.

**Мастер импорта - сопоставление задач**

Сопоставление данных задач

Имя исходной таблицы:  
Таблица\_задач1

Проверьте или измените сопоставление данных.

Из: поле текстового файла	В: поле Microsoft Project	Тип данных
Название	Название	Текст
Уровень_структуры	Уровень структуры	Текст
Начало	Начало	Текст
Окончание	Окончание	Текст
Длительность	Длительность	Текст
Названия ресурсов	Названия ресурсов	Текст

Добавить все    Очистить все    Добавить строку    Удалить строку

Образец

Текстовый файл: Название		Уровень_структуры	Начало
Проект:	Название	Уровень структуры	Начало
Образец:	Итерация 1 - Фаза Начало	1	Ср 10.01.1
	Управление проектом	2	Ср 10.01.1

<    >

Справка    < Назад    Далее >    Готово    Отмена

*Рисунок 3.6.4 – Сопоставление данных*

Импортированный файл в приложении Microsoft Project представлен на рисунке 3.6.5 ниже, и представляет из себя диаграмму Ганта, где указаны в качестве задач: итерации, рабочие процессы, потоки, деятельности с соблюдением вложенности, для каждой задачи определена длительность, дата начала и дата окончания задачи, входящие и исходящие артефакты. Далее полученный файл редактируется, заполняется и настраивается менеджером проектов.

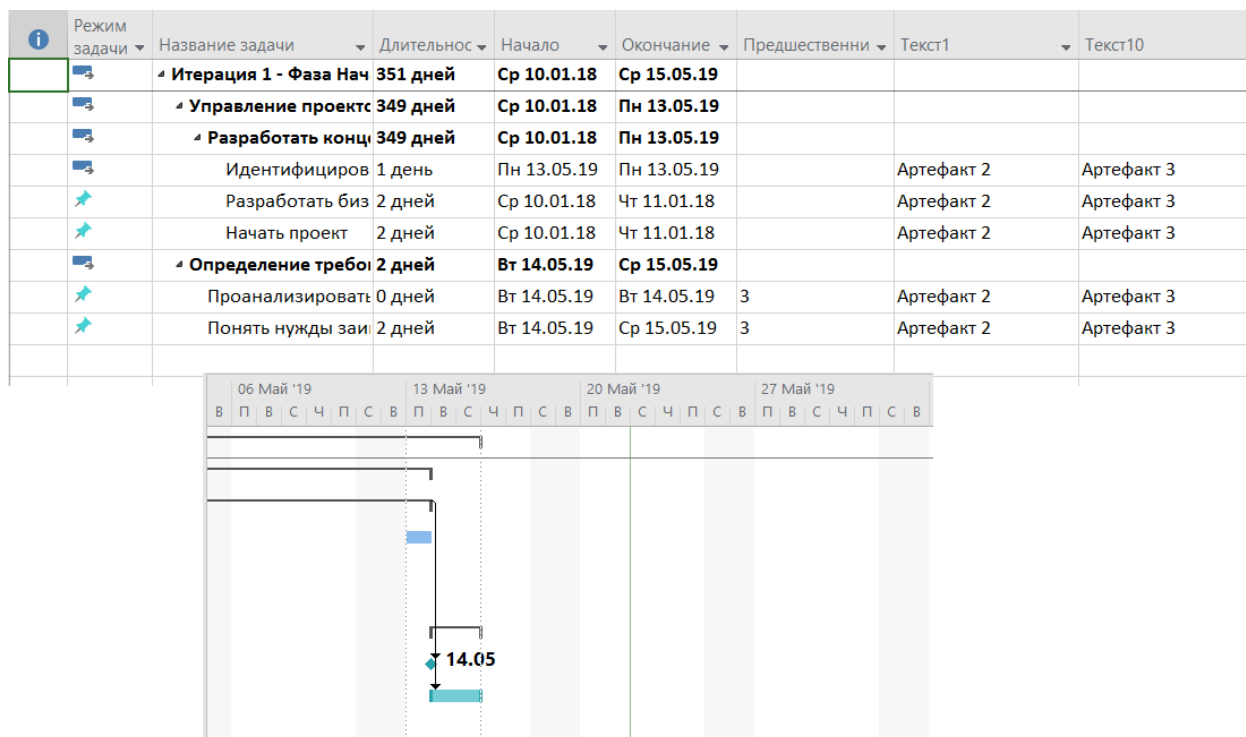


Рисунок 3.6.5 – Импортированный файл формата CSV в приложении Microsoft Project

На рисунке 3.6.6 представлена вкладка «Детали проекта» с общей информацией по проекту, выбранными фазами и процессами. Возможен переход по вкладкам и возврат в список проектов.

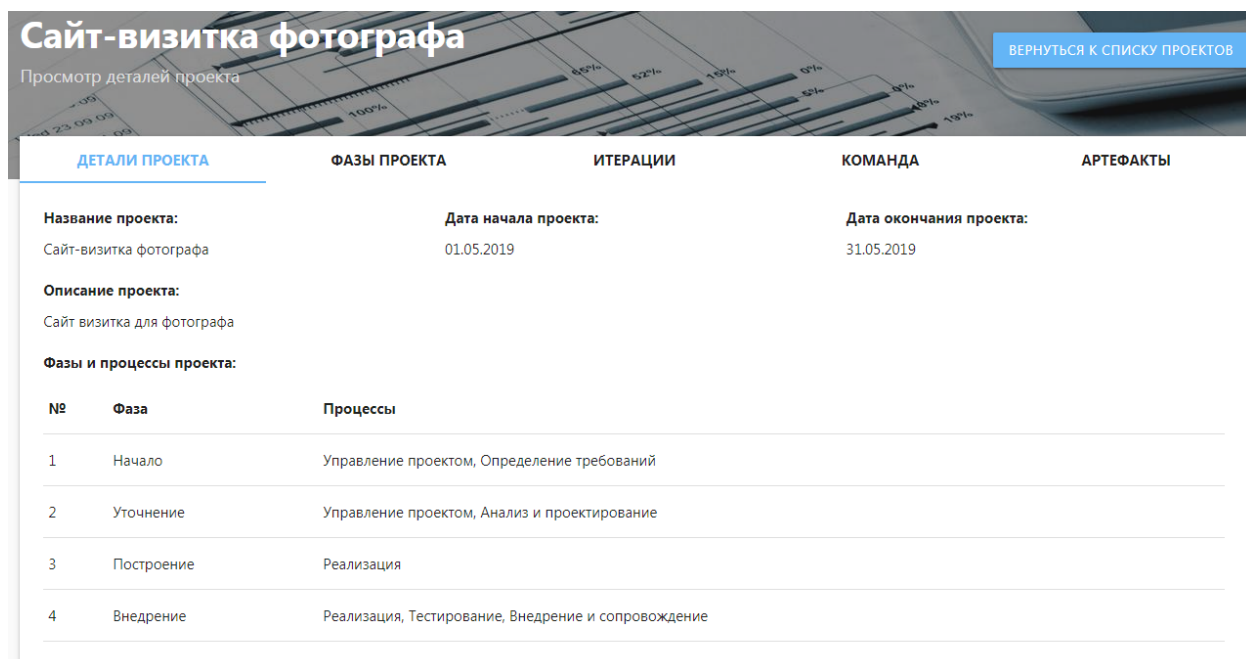


Рисунок 3.6.6 – Детали проекта

На рисунке 3.6.7 представлена вкладка «Фазы проекта» и информация о статусе и количестве итераций по каждой фазе.

ДЕТАЛИ ПРОЕКТА		ФАЗЫ ПРОЕКТА	ИТЕРАЦИИ	КОМАНДА	АРТЕФАКТЫ
№	Фаза	Кол-во итераций		Статус	
1	Начало	1		Не начата	
2	Уточнение	1		Не начата	
3	Построение	1		Не начата	
4	Внедрение	1		Не начата	

*Рисунок 3.6.7 – Фазы проекта*

При создании проекта автоматически создаются итерации для всех выбранных фаз. Если нужны дополнительные итерации пользователь может воспользоваться функцией «Создать итерацию».

На рисунке 3.6.8 показана вкладка «Итерации» представляющая собой список итераций со статусами и временными рамками. Возможные действия: просмотр потоков по рабочему процессу, изменение статуса и удаление итерации. Для итераций действует проверка условий, нельзя начать следующую итерацию не завершив предыдущую, итерацию можно удалить только если она в статусе «Не начата».

Сайт-визитка фотографа

Просмотр деталей проекта

ВЕРНУТЬСЯ К СПИСКУ ПРОЕКТОВ

ДЕТАЛИ ПРОЕКТА

ФАЗЫ ПРОЕКТА

ИТЕРАЦИИ

КОМАНДА

АРТЕФАКТЫ

Фазы

Статус

Дата

ОЧИСТИТЬ ФИЛЬТР

СОЗДАТЬ ИТЕРАЦИЮ

№	Фаза	Итерация	Процессы	Дата начала	Дата завершения	Актеры	Артефакты	Статус	Действия
1	Начало	Итерация - Начало	Управление проектом, Определение требований	15.05.2019	29.05.2019	2	4	Не начата	
2	Уточнение	Итерация - Уточнение	Управление проектом, Анализ и проектирование	29.05.2019	12.06.2019	2	9	Не начата	
3	Построение	Итерация - Построение	Реализация	12.06.2019	26.06.2019	1	5	Не начата	
4	Внедрение	Итерация - Внедрение	Реализация, Тестирование, Внедрение и сопровождение	26.06.2019	10.07.2019	3	20	Не начата	

Рисунок 3.6.8 – Итерации проекта

Для проекта реализована функция добавление команды, скриншот представлен на рисунке 3.6.9. Список ролей (исполнителей) предустановлен в системе адаптации. Участника команды можно удалить или сделать неактивным. Из данной вкладки возможен переход в предыдущие вкладки и на вкладку «Артефакты».

Сайт-визитка фотографа

Просмотр деталей проекта

ВЕРНУТЬСЯ К СПИСКУ ПРОЕКТОВ

ДЕТАЛИ ПРОЕКТА

ФАЗЫ ПРОЕКТА

ИТЕРАЦИИ

КОМАНДА

АРТЕФАКТЫ

ДОБАВИТЬ В КОМАНДУ

№	ФИО	Исполнитель	Активность	Действия
1	Трушников Елен	Менеджер проектов		

Рисунок 3.6.9 – Команда проекта

Пользователь имеет возможность ознакомиться со списком входящих и исходящих артефактов, исполнителями, скриншот реализованной системы представлен на рисунке 3.6.10 – Артефакты проекта. Данную функцию можно использовать при создании матрицы ответственности.



ДЕТАЛИ ПРОЕКТА		ФАЗЫ ПРОЕКТА	ИТЕРАЦИИ	КОМАНДА		АРТЕФАКТЫ	
№	Входящий артефакт		Исходящий артефакт	Актер	ФИО	Итерация	Состояние
2	Варианты использования, Видение продукта, Модель бизнес-процессов, Объектная модель предметной области, Требования заинтересованных лиц		Глоссарий	Системный аналитик, Заинтересованное лицо		-	В ожидании
3	-		Глоссарий	Системный аналитик		-	В ожидании
3	Видение продукта		Список рисков	Менеджер проектов	Трушникова Елена	-	В ожидании
3	Глоссарий, Видение продукта, Варианты использования, Дополнительные требования, Стандарты требований к интерфейсу, Документ по архитектуре ПО		Документ по архитектуре ПО	Архитектор программного обеспечения		-	В ожидании

Рисунок 3.6.10 – Артефакты проекта

При создании проекта пользователь может исключить из проекта деятельность вместе с артефактами, наглядно показано на рисунке 3.6.11. На данной вкладке также доступен фильтр по фазе, рабочим процессам и потокам рабочего процесса, так как список деятельности может содержать большое количество строк.

### Создание проекта

Настройка нового проекта

[ВЕРНУТЬСЯ К СПИСКУ ПРОЕКТОВ](#)

ДЕТАЛИ ПРОЕКТА		НАСТРОЙКА ПРОЕКТА		ПОТОКИ И ДЕЯТЕЛЬНОСТИ		ИТЕРАЦИИ	КОМАНДА	
Фазы		Рабочие процессы		Поток работ процесса			<a href="#">ОЧИСТИТЬ ФИЛЬТР</a>	
№	Фаза	Рабочий процесс	Название деятельности	Шаги	Артефакты входящие	Артефакты исходящие	Роль	Выбрать
2	Начало	Определение требований	Создать общий словарь		Варианты использования, Видение продукта, Модель бизнес-процессов, Объектная модель предметной области, Требования заинтересованных лиц	Глоссарий	Системный аналитик, Заинтересованное лицо	✓
3	Начало	Определение требований	Создать общий словарь		-	Глоссарий	Системный аналитик	✓
3	Начало	Управление проектом	Идентифицировать и назначить риски		Видение продукта	Список рисков	Менеджер проектов	✓
3	Уточнение	Реализация	Структурировать модель реализации		Дополнительные требования, Модель данных	Модель реализации	Архитектор программного обеспечения	✓
4	Уточнение	Управление проектом	Выделить персонал		План разработки	План разработки	Менеджер проектов	✓

[ОТМЕНА](#)
[ДАЛЕЕ](#)

Рисунок 3.6.11 – Потоки деятельности

## **Выводы по разделу.**

Реализованное веб-приложение адаптации UP уже сейчас предоставляет достаточный объем функциональных возможностей для менеджеров проектов, который способен покрыть и решить ряд вопросов являющихся наиболее насущными в управлении проектами, а именно:

- Акцентирование внимания всех участников проекта на контрольных точках фаз и набора условий выхода из фазы. Понимание целей проекта это половина успеха. Большинство задержек в релизах возникают из-за того, что команда разработки неверно представляет себе цели проекта или итерации.
- Карта проекта (схема потоков работ) для каждой фазы, формирует видение по проекту и картину в целом.
- Построение диаграммы Ганта по настроенному процессу. Большинство компаний вручную переносят задачи из систем управления проектами и создают диаграмму Ганта с использованием сторонних сервисов.
- Статусы итераций и фаз отражают где и в какой момент мы находимся в проекте, так называемые заинтересованные лица проекта могут отслеживать статус проекта по итерации и в целом.



## **Заключение**

Результатом данной работы стало успешное завершение проекта. В рамках данной работы решены все поставленные задачи:

- Формализовано описание процесса адаптации унифицированного процесса к требованиям конкретного проекта.
- Спроектирована система процесса адаптации.
- Реализована система процесса адаптации.

Цель исследования «Реализовать систему адаптации унифицированного процесса к требованиям конкретного проекта» достигнута. Система позволяет решить круг вопросов и задач в сфере управления проектами по разработке программного обеспечения такие как унификация процессов, понимание содержания каждого этапа разработки, сроков и планов.

Кроме того, сформулированы рекомендации по дальнейшему развитию проекта, одной из ключевых рекомендаций является локализация приложения, перевод системы на английский язык, это обусловлено тем что данная система будет внедряться в международной компании, где основным языком общения – английский язык.

Далее после реализации локализации одним из следующих шагов планируется презентация системы адаптации проектным менеджерам и продуктовым менеджерам и последующий пилотный запуск на одном из рабочих проектов.

Система адаптации должна помочь проектной группе унифицировать процессы управления проектами и добиться максимальной эффективности в одном из критичных процессов – поставка релизов в срок и качественно.

## Глоссарий

Унифицированный процесс (UP – Unified Process) – это методология. Она указывает на исполнителей (актеров), действия (деятельности) и артефакты, которые необходимо использовать, осуществить или создать для моделирования программной системы.

Фаза – стадия жизненного цикла, которая завершается достижением конкретного результата (созданием артефактов) используемого далее в качестве управления последующими работами или завершающего реализацию проекта.

Рабочий процесс – определяет какие виды работ должны быть выполнены и какой результат должен быть получен.

Поток – это последовательность видов работ (деятельность) выполняемых в рамках рабочего процесса

Деятельность – это работа выполняемая в рамках потока рабочего процесса.

Артефакт – это результат деятельности рабочего процесса.

CSV (Comma-Separated Values – значения разделенные запятыми) – текстовый формат, предназначенный для представления табличных данных.

Итерация – каждая фаза унифицированного процесса разделяется на итерации. Итерация – это полный цикл разработки, вырабатывающий промежуточный продукт.

Актер – это роль, исполняемые сущностями, непосредственно взаимодействующими с системой.

Унифицированный язык моделирования (Unified Modeling Language, UML) – это универсальный язык визуального моделирования систем.

Жизненный цикл проекта (англ. Project Life Cycle) — последовательность фаз проекта, задаваемая исходя из потребностей управления проектом.

Диагра́мма Га́нта (англ. Gantt chart, также ленточная диаграмма, график Ганта) — это популярный тип столбчатых диаграмм(гистограмм), который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов. Используется в приложениях по управлению проектами.

UX – (user experience, пользовательский опыт) система проектируется на основе потребностей пользователей и результатов тестирований.

UI – (user interface, пользовательский интерфейс) взаимодействие человека с сайтом. Дизайн пользовательского интерфейса (UI) – это дизайн кнопок, полей ввода, форм заявки – всех точек взаимодействия пользователя с сайтом.

Agile – гибкая методология разработки ПО, ориентированная на итерационный подход, динамическое формирование требований и их реализацию в результате постоянного взаимодействия с заинтересованными лицами.

## Список использованных источников и литературы

1. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. Спб.: Питер, 2002. – 496 с.
2. Кратчен Ф. Введение в Rational Unified Process.-М.: ДИАЛОГ МИФИ, 2000.-304 с. 5.Леффингуэл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению.-М.: Издательский дом «Вильямс», 2002.-288 с.
3. Арлоу Д., Нейштадт А. UML2 и унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. - СПб.: Символ-плюс, 2007. - 624 с.
4. Библиографическое описание документа : общие требования и правила составления : рекомендации [Электронный ресурс] URL: [http://www.lib.tsu.ru/win/produkcija/metodichka/spis\\_lit.html](http://www.lib.tsu.ru/win/produkcija/metodichka/spis_lit.html) (дата обращения: 22.05.2019).
5. Rational Software Corporation documentation [Электронный ресурс] URL: <https://sce.uhcl.edu/helm/rationalunifiedprocess/> (дата обращения: 15.01.2018).
6. Проблемы автоматизации структурно-параметрического синтеза [Электронный ресурс] URL: <http://www.structuralist.narod.ru/dictionary/rup.htm> (дата обращения: 10.05.2019).
7. Свободная энциклопедия «Википедия» [Электронный ресурс] URL: <https://ru.wikipedia.org> (дата обращения: 10.05.2019).
8. Издательство «Открытые системы» [Электронный ресурс] URL: <https://www.osp.ru/os/2012/04/13015775/> (дата обращения: 12.05.2019).
9. Методология Oracle Unified Method [Электронный ресурс] URL: <https://stan1slav.blogspot.com/2011/11/oracle-unified-method.html> (дата обращения: 06.05.2018).

10. Tproger. Издание о разработке [Электронный ресурс] URL: <https://tproger.ru/articles/> (дата обращения: 18.05.2018).

11. Gitlab User Guide [Электронный ресурс] // Gitlab [сайт]. 2019. URL: <https://docs.gitlab.com/> (дата обращения: 06.12. 2018).

## Приложение А

Таблица А.1 – Фазы унифицированного RUP

Название фазы	Названия рабочих процессов
Начало	Управление проектом Определение требований Анализ и проектирование Тестирование
Уточнение	Управление проектом Определение требований Анализ и проектирование Реализация Тестирование
Построение	Управление проектом Определение требований Анализ и проектирование Реализация Тестирование
Переход	Управление проектом Определение требований Анализ и проектирование Реализация Тестирование Внедрение и сопровождение

Таблица А.2 - Рабочие процессы и потоки RUP

Название Фазы	Название рабочего процесса	Потоки работ
Начало	Управление проектом	Разработать концепцию проекта Оценить масштаб и риск проекта Реализовать план разработки проекта Разработать план итерации Управление итерацией Мониторинг и контроль проекта Переоценить объемы проекта и риски План для следующей итерации Усовершенствовать план разработки проекта

## Продолжение Приложения А

Продолжение таблицы А.2

Название Фазы	Название рабочего процесса	Потоки работ
Начало	Определение требований	Проанализировать проблемы Понять нужды заинтересованных лиц Описать систему Управлять объемом работ Уточнить описание системы Управлять изменениями требований
	Анализ и проектирование	Выполнить архитектурный анализ
	Тестирование	Описать оценку качества задач проекта
Уточнение	Управление проектом	Управлять итерацией Контролировать и управлять проектом Переоценить объем проекта и риски Подготовить план для следующей итерации Детализировать план разработки
	Определение требований	Проанализировать проблемы Понять нужды заинтересованных лиц Описать систему Управлять объемом работ Уточнить описание системы Управлять изменениями требований
	Анализ и проектирование	Определить тип архитектуры Проанализировать функциональные возможности Спроектировать компоненты Спроектировать базу данных Детализировать архитектуру

## Продолжение Приложения А

Продолжение таблицы А.2

Название Фазы	Название рабочего процесса	Потоки работ
Уточнение	Реализация	Структурировать модель реализации Реализовать компоненты
	Тестирование	Описать оценку качества задач Проверить тестовый подход Протестировать и оценить Представить важный результат оценки качества задач Улучшить тестовые ресурсы
Построение	Управление проектом	Управлять итерацией Контролировать и управлять проектом Переоценить объем проекта и риски Подготовить план для следующей итерации Детализировать План разработки
	Определение требований	Управлять изменениями требований
	Анализ и проектирование	Спроектировать компоненты Спроектировать базу данных Детализировать архитектуру
	Реализация	Реализовать компоненты
	Тестирование	Описать оценку качества задач Проверить тестовый подход Проверить стабильность сборки Протестировать и оценить Представить важный результат оценки качества задач Улучшить тестовые ресурсы
Переход	Управление проектом	Управлять итерацией Контролировать и управлять проектом, Завершить проект



## Продолжение Приложения А

Окончание таблицы А.2

Название Фазы	Название рабочего процесса	Потоки работ
Переход	Определение требований	Управлять изменениями требований
	Анализ и проектирование	Детализировать архитектуру
	Реализация	Реализовать компоненты
	Тестирование	Описать оценку качества задач Проверить тестовый подход Протестировать и оценить Представить важный результат оценки качества задач Улучшить тестовые ресурсы
	Внедрение и сопровождение	План развертывания Разработать вспомогательные материалы Управлять тестом приемки Предоставить развернутую систему Управлять тестом приемки Предоставить установочный пакет Предоставить доступ

## Продолжение Приложения А

Таблица А.3 - Потоки и деятельности работ

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Начало	Управление проектом	Разработать концепцию проекта	<ul style="list-style-type: none"> <li>- Идентифицировать и назначить риски</li> <li>- Разработать бизнес-кейс</li> <li>- Начать проект</li> </ul>
		Оценить масштаб и риск проекта	<ul style="list-style-type: none"> <li>- Идентифицировать и назначить риски</li> <li>- Разработать бизнес-кейс</li> </ul>
		Реализовать план разработки проекта	<ul style="list-style-type: none"> <li>- Разработать план метрик</li> <li>- Разработать план обеспечения качества</li> <li>- Разработать план по риск менеджменту</li> <li>- Определить организационную структуру и команду</li> <li>- Определить процессы мониторинга и контроля</li> <li>- Разработать план по принятию продукта</li> <li>- План фаз и итераций</li> <li>- Разработать план решения проблем</li> <li>- Составить план разработки программного обеспечения</li> </ul>
		Разработать план итерации	<ul style="list-style-type: none"> <li>- Разработать план итерации</li> <li>- Разработать бизнес-кейс</li> </ul>
		Управление итерацией	<ul style="list-style-type: none"> <li>- Выделить команду</li> <li>- Начать итерацию</li> <li>- Оценить итерацию</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Начало	Управление проектом	Мониторинг и контроль проекта	<ul style="list-style-type: none"> <li>- Составить график работ и назначить исполнителей</li> <li>- Мониторить статус проекта</li> <li>- Готовить отчет по статусу</li> <li>- Обработать исключения и проблемы</li> </ul>
		Переоценить объемы проекта и риски	<ul style="list-style-type: none"> <li>- Идентифицировать и оценить риски</li> <li>- Разработать бизнес-кейс</li> </ul>
		План для следующей итерации	<ul style="list-style-type: none"> <li>- Разработать план итерации</li> <li>- Разработать бизнес-кейс</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Начало	Управление проектом	Усовершенствовать план разработки проекта	<ul style="list-style-type: none"> <li>- Разработать план метрик</li> <li>- Разработать план обеспечения качества</li> <li>- Разработать план по риск менеджменту</li> <li>- Определить организационную структуру и команду</li> <li>- Определить процессы мониторинга и контроля</li> <li>- Разработать план по принятию продукта</li> <li>- План фаз и итераций</li> <li>- Разработать план решения проблем</li> <li>- Составить план разработки программного обеспечения</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Начало	Определение требований	Проанализировать проблемы	<ul style="list-style-type: none"> <li>- Создать общий словарь</li> <li>- Разработать план управления требованиями</li> <li>- Найти актеров и варианты использования</li> <li>- Разработать видение</li> </ul>
		Понять нужды заинтересованных лиц	<ul style="list-style-type: none"> <li>- Создать общий словарь</li> <li>- Собрать требования заинтересованных лиц</li> <li>- Разработать видение</li> <li>- Найти актеров и варианты использования</li> <li>- Управлять зависимостями</li> </ul>
		Описать систему	<ul style="list-style-type: none"> <li>- Разработать видение</li> <li>- Создать общий словарь</li> <li>- Найти актеров и варианты использования</li> <li>- Управлять зависимостями</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Начало	Анализ и проектирование	Управлять объемом работ	- Приоритезировать варианты использования - Разработать видение - Управлять зависимостями
		Уточнить описание системы	- Детализировать варианты использования - Детализировать требования к разработке - Построить модель пользовательского интерфейса - Создать пользовательский интерфейс
		Управлять изменениями требований	- Структурировать модель вариантов использования - Управлять зависимостями
		Выполнить архитектурный анализ	- Провести архитектурный анализ - Разработать архитектурное подтверждение концепции - Оценить устойчивость к использованию архитектурное подтверждение концепции

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Начало	Тестирование	Определить оценку миссии	<ul style="list-style-type: none"> <li>- Определить тестовые мотиваторы</li> <li>- Понять миссию и цели</li> <li>- Определить цели тестирования</li> <li>- Описать оценку и учет требований</li> <li>- Определить идеи тестирования</li> <li>- Описать подход к тестированию</li> </ul>
Уточнение	Управление проектом	Управлять итерацией	<ul style="list-style-type: none"> <li>- Выделить персонал</li> <li>- Начать итерацию</li> <li>- Оценить итерацию</li> </ul>
		Контролировать и управлять проектом	<ul style="list-style-type: none"> <li>- Составить график работ и назначить исполнителей</li> <li>- Мониторить статус проекта</li> <li>- Готовить отчет по статусу</li> <li>- Обработать исключения и проблемы</li> </ul>
		Переоценить объем проекта и риски	<ul style="list-style-type: none"> <li>- Идентифицировать и оценить риски</li> <li>- Разработать бизнес-кейс</li> </ul>
		Подготовить план для следующей итерации	<ul style="list-style-type: none"> <li>- Разработать план итерации</li> <li>- Разработать бизнес-кейс</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Уточнение	Управление проектом	Детализировать план разработки	<ul style="list-style-type: none"> <li>- Разработать план метрик</li> <li>- Разработать план обеспечения качества</li> <li>- Разработать план по риск менеджменту</li> <li>- Определить организационную структуру и команду</li> <li>- Определить процессы мониторинга и контроля</li> <li>- Разработать план по принятию продукта</li> <li>- План фаз и итераций</li> <li>- Разработать план решения проблем</li> <li>- Составить план разработки программного обеспечения</li> </ul>
	Определение требований	Проанализировать проблемы	<ul style="list-style-type: none"> <li>- Создать общий словарь</li> <li>- Разработать план управления требованиями</li> <li>- Найти актеров и варианты использования</li> <li>- Разработать видение</li> </ul>



## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Уточнение	Определение требований	Понять нужды заинтересованных лиц	<ul style="list-style-type: none"> <li>- Создать общий словарь</li> <li>- Собрать требования заинтересованных лиц</li> <li>- Разработать видение</li> <li>- Найти актеров и варианты использования</li> <li>- Управлять зависимостями</li> </ul>
		Описать систему	<ul style="list-style-type: none"> <li>- Разработать видение</li> <li>- Создать общий словарь</li> <li>- Найти актеров и варианты использования</li> <li>- Управлять зависимостями</li> </ul>
		Управлять объемом работ	<ul style="list-style-type: none"> <li>- Приоритезировать варианты использования</li> <li>- Разработать видение</li> <li>- Управлять зависимостями</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Уточнение	Определение требований	Уточнить описание системы	- Детализировать варианты использования - Детализировать требования к разработке - Построить модель пользовательского интерфейса - Создать пользовательский интерфейс
		Управлять изменениями требований	- Структурировать модель варианты использования - Управлять зависимостями
	Анализ и проектирование	Определить тип архитектуры	- Провести архитектурный анализ - Проанализировать варианты использования
		Проанализировать функциональные возможности	- Проанализировать варианты использования - Определить элементы дизайна
		Разработать системную архитектуру компонентов	- Разработать варианты использования - Спроектировать классы

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Уточнение	Анализ и проектирование	Спроектировать базу данных	- Разработать схему БД - Спроектировать классы
	Реализация	Структурировать модель реализации	Структурировать модель реализации
		Реализовать компоненты	- Реализовать компоненты - Проверить код - Исправить дефекты - Реализовать Unit tests
	Тестирование	Описать оценку качества задач	- Определить тестовые мотиваторы - Понять миссию и цели - Определить цели тестирования - Описать оценку и учет требований - Определить идеи тестирования - Описать подход к тестированию

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Уточнение	Тестирование	Проверить тестовый подход	<ul style="list-style-type: none"> <li>- Определить настройки для тестового окружения</li> <li>- Определить механизмы тестирования</li> <li>- Определить тестируемые элементы</li> <li>- Определить детали тестирования</li> <li>- Реализовать набор тестов</li> <li>- Реализовать значимые тесты</li> <li>- Поддерживать нужды тестирования</li> </ul>
		Протестировать и оценить	<ul style="list-style-type: none"> <li>- Реализовать значимые тесты</li> <li>- Реализовать набор тестов</li> <li>- Выполнить набор тестов</li> <li>- Проанализировать провалы в тестировании</li> <li>- Структурировать реализацию тестов</li> <li>- Определить тестируемые элементы</li> <li>- Идентифицировать задачи тестирования</li> <li>- Определить детали тестирования</li> <li>- Проверить изменения в сборке</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Уточнение	Тестирование	Представить важный результат оценки качества задач	<ul style="list-style-type: none"> <li>- Оценить и минимизировать трудозатраты в тестировании</li> <li>- Оценить и поддерживать качество</li> <li>- Определить результаты тестирования</li> <li>- Идентифицировать задачи тестирования</li> <li>- Реализовать значимые тесты</li> <li>- Реализовать набор тестов</li> <li>- Выполнить набор тестов</li> <li>- Проанализировать провалы в тестировании</li> </ul>
		Улучшить тестовые ресурсы	<ul style="list-style-type: none"> <li>- Разработать руководства по тестированию</li> <li>- Определить тестируемые элементы</li> <li>- Структурировать реализацию тестов</li> <li>- Идентифицировать задачи тестирования</li> <li>- Определить детали тестирования</li> <li>- Описать оценку и учет требований</li> <li>- Реализовать значимые тесты</li> <li>- Реализовать набор тестов</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Построение	Управление проектом	Управлять итерацией	<ul style="list-style-type: none"> <li>- Выделить персонал</li> <li>- Начать итерацию</li> <li>- Оценить итерацию</li> </ul>
Построение	Управление проектом	Контролировать и управлять проектом	<ul style="list-style-type: none"> <li>- Составить график работ и назначить исполнителей</li> <li>- Мониторить статус проекта</li> <li>- Готовить отчет по статусу</li> <li>- Обработать исключения и проблемы</li> </ul>
		Переоценить объем проекта и риски	<ul style="list-style-type: none"> <li>- Идентифицировать и оценить риски</li> <li>- Разработать бизнес-кейс</li> </ul>
		Подготовить план для следующей итерации	<ul style="list-style-type: none"> <li>- Разработать план итерации</li> <li>- Разработать бизнес-кейс</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Построение	Управление проектом	Детализировать план разработки	<ul style="list-style-type: none"> <li>- Разработать план метрик</li> <li>- Разработать план обеспечения качества</li> <li>- Разработать план по риск менеджменту</li> <li>- Определить организационную структуру и команду</li> <li>- Определить процессы мониторинга и контроля</li> <li>- Разработать план по принятию продукта</li> <li>- План фаз и итераций</li> <li>- Разработать план решения проблем</li> <li>- Составить план разработки программного обеспечения</li> </ul>
	Определение требований	Управлять изменениями требований	<ul style="list-style-type: none"> <li>- Структурировать варианты использования</li> <li>- Управлять зависимостями</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Построение	Анализ и проектирование	Спроектировать компоненты	- Спроектировать варианты использования - Спроектировать классы
		Спроектировать базу данных	- Спроектировать классы - Спроектировать базу данных
		Детализировать архитектуру	- Определить механизмы структуры проекта - Определить элементы структуры проекта - Включить существующие элементы архитектуры - Описать архитектуру среды выполнения - Описать распределение системы



## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Построение	Реализация	Реализовать компоненты	<ul style="list-style-type: none"> <li>- Реализовать компоненты</li> <li>- Проверить код</li> <li>- Исправить дефекты</li> <li>- Реализовать Unit tests</li> </ul>
	Тестирование	Описать оценку качества задач	<ul style="list-style-type: none"> <li>- Определить тестовые мотиваторы</li> <li>- Понять миссию и цели</li> <li>- Определить цели тестирования</li> <li>- Описать оценку и учет требований</li> <li>- Определить идеи тестирования</li> <li>- Описать подход к тестированию</li> </ul>
		Проверить тестовый подход	<ul style="list-style-type: none"> <li>- Определить настройки для тестового окружения</li> <li>- Определить тестируемые элементы</li> <li>- Реализовать набор тестов</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Построение	Тестирование	Проверить стабильность сборки	<ul style="list-style-type: none"> <li>- Реализовать тесты</li> <li>- Выполнить тесты</li> <li>- Проанализировать провалы тестирования</li> <li>- Определить детали тестирования</li> <li>- Определить результаты тестирования</li> <li>- Оценить и поддерживать качество</li> </ul>
		Протестировать и оценить	<ul style="list-style-type: none"> <li>- Реализовать значимые тесты</li> <li>- Реализовать набор тестов</li> <li>- Выполнить набор тестов</li> <li>- Проанализировать провалы в тестировании</li> <li>- Структурировать реализацию тестов</li> <li>- Определить тестируемые элементы</li> <li>- Идентифицировать задачи тестирования</li> <li>- Определить детали тестирования</li> <li>- Проверить изменения в сборке</li> <li>- Определить результаты тестирования</li> <li>- Оценить и минимизировать трудозатраты в тестировании</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Построение	Тестирование	Представить важный результат оценки качества задач	<ul style="list-style-type: none"> <li>- Оценить и минимизировать трудозатраты в тестировании</li> <li>- Оценить и поддерживать качество</li> <li>- Определить результаты тестирования</li> <li>- Идентифицировать задачи тестирования</li> <li>- Реализовать значимые тесты</li> <li>- Реализовать набор тестов</li> <li>- Выполнить набор тестов</li> <li>- Проанализировать провалы в тестировании</li> </ul>
		Улучшить тестовые ресурсы	<ul style="list-style-type: none"> <li>- Разработать руководства по тестированию</li> <li>- Определить тестируемые элементы</li> <li>- Структурировать реализацию тестов</li> <li>- Идентифицировать задачи тестирования</li> <li>- Определить детали тестирования</li> <li>- Описать оценку и учет требований</li> <li>- Реализовать значимые тесты</li> <li>- Реализовать набор тестов</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Переход	Управление проектом	Управлять итерацией	- Выделить персонал - Начать итерацию - Оценить итерацию
		Контролировать и управлять проектом	- Составить график работ и назначить исполнителей - Мониторить статус проекта - Готовить отчет по статусу - Обработать исключения и проблемы
		Завершить проект	Подготовить завершение проекта
	Определение требований	Управлять изменениями требований	- Структурировать варианты использования - Управлять зависимостями
	Анализ и проектирование	Детализировать архитектуру	- Определить механизмы структуры проекта - Определить элементы структуры проекта - Включить существующие элементы архитектуры - Описать архитектуру среды выполнения - Описать распределение системы

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Переход	Реализация	Реализовать компоненты	<ul style="list-style-type: none"> <li>- Реализовать компоненты</li> <li>- Проверить код</li> <li>- Исправить дефекты</li> <li>- Реализовать Unit tests</li> </ul>
	Тестирование	Описать оценку качества задач	<ul style="list-style-type: none"> <li>- Определить тестовые мотиваторы</li> <li>- Понять миссию и цели</li> <li>- Определить цели тестирования</li> <li>- Описать оценку и учет требований</li> <li>- Определить идеи тестирования</li> <li>- Описать подход к тестированию</li> </ul>
		Проверить стабильность сборки	<ul style="list-style-type: none"> <li>- Реализовать тесты</li> <li>- Выполнить тесты</li> <li>- Проанализировать провалы тестирования</li> <li>- Определить детали тестирования</li> <li>- Определить результаты тестирования</li> <li>- Оценить и поддерживать качество</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Переход	Тестирование	Протестировать и оценить	<ul style="list-style-type: none"> <li>- Реализовать значимые тесты</li> <li>- Реализовать набор тестов</li> <li>- Выполнить набор тестов</li> <li>- Проанализировать провалы в тестировании</li> <li>- Структурировать реализацию тестов</li> <li>- Определить тестируемые элементы</li> <li>- Идентифицировать задачи тестирования</li> <li>- Определить детали тестирования</li> <li>- Проверить изменения в сборке</li> <li>- Определить результаты тестирования</li> <li>- Оценить и минимизировать трудозатраты в тестировании</li> </ul>

## Продолжение Приложения А

Продолжение таблицы А.3

Название Фазы	Название рабочего процесса	Название потока работ	Название деятельности потока
Переход	Тестирование	Представить важный результат оценки качества задач	<ul style="list-style-type: none"> <li>- Оценить и минимизировать трудозатраты в тестировании</li> <li>- Оценить и поддерживать качество</li> <li>- Определить результаты тестирования</li> <li>- Идентифицировать задачи тестирования</li> <li>- Реализовать значимые тесты</li> <li>- Реализовать набор тестов</li> <li>- Выполнить набор тестов</li> <li>- Проанализировать провалы в тестировании</li> </ul>
		Улучшить тестовые ресурсы	<ul style="list-style-type: none"> <li>- Разработать руководства по тестированию</li> <li>- Определить тестируемые элементы</li> <li>- Идентифицировать задачи тестирования</li> <li>- Определить детали тестирования</li> <li>- Описать оценку и учет требований</li> <li>- Реализовать значимые тесты</li> <li>- Реализовать набор тестов</li> </ul>

## Продолжение Приложения А

Окончание таблицы А.3

Переход	Внедрение и сопровождение	План развертывания	- Разработать план развертывания - Определить спецификацию продукта
		Разработать вспомогательные материалы	- Разработать учебные материалы - Разработать справочные материалы
		Управлять тестом приемки	Управлять приемочным тестом
		Предоставить развернутую систему	- Написать примечание к версии - Разработать артефакты установки
		Управлять тестом приемки	Управлять приемочным тестом
		Предоставить установочный пакет	- Выпустить в производство - Проверить выпущенный продукт - Создать иллюстрацию продукта
		Предоставить доступ	Предоставить доступ для скачивания продукта



## Продолжение Приложения А

Таблица А.4 - Роли

#	Наименование исполнителя	Тип исполнителя
1	Системный аналитик	Аналитик
3	Спецификатор требований	Аналитик
5	Дизайнер	Аналитик
6	Рецензент кода	Разработчик
7	Архитектор системы	Разработчик
8	Тестирующий	Тестирующий
9	Менеджер проектов	Менеджер
10	Менеджер развертывания	Менеджер
11	Заинтересованное лицо	Дополнительная роль
12	Системный администратор	Дополнительная роль
13	Технический писатель	Дополнительная роль

# Отчет о проверке на заимствования №1



Автор: Т Elena [helentrn@gmail.com](mailto:helentrn@gmail.com) / ID: 6727528

Проверяющий: Т Elena ([helentrn@gmail.com](mailto:helentrn@gmail.com)) / ID: 6727528)




Отчет предоставлен сервисом «Антиплагиат»- <http://users.antiplagiat.ru>

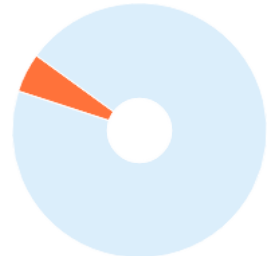
## ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 13  
Начало загрузки: 04.06.2019 07:29:29  
Длительность загрузки: 00:00:02  
Имя исходного файла: Реализация системы адаптации унифицированного процесса к требованиям конкретного проекта\_Трушников Елен  
Размер текста: 2741 кБ  
Символов в тексте: 78990  
Слов в тексте: 8864  
Число предложений: 481

## ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)  
Начало проверки: 04.06.2019 07:29:31  
Длительность проверки: 00:00:02  
Комментарии: не указано  
Модули поиска: Модуль поиска Интернет

ЗАИМСТВОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
5,22% 	0% 	94,78% 



Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.  
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.

Заимствования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Ссылка	Актуален на	Модуль поиска
[01]	1,21%	Rational Method Composer. Система IBM Rational	<a href="http://bourabai.kz">http://bourabai.kz</a>	25 Мар 2016	Модуль поиска Интернет
[02]	1,15%	не указано	<a href="http://inf.tsu.ru">http://inf.tsu.ru</a>	раньше 2011	Модуль поиска Интернет
[03]	0%	<a href="https://istina.msu.ru/download/3246593/1fXmY2:gHcp8THbO1U4fMq4Hqd...">https://istina.msu.ru/download/3246593/1fXmY2:gHcp8THbO1U4fMq4Hqd...</a>	<a href="https://istina.msu.ru">https://istina.msu.ru</a>	15 Сен 2018	Модуль поиска Интернет

Еще источников: 9

Еще заимствований: 2,86%