

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Инженерная школа информационных технологий и робототехники  
Направление подготовки 09.04.01 Информатика и вычислительная техника  
Отделение информационных технологий

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема работы
Автоматизация доставки, установки и обновления лицензированного программного обеспечения на базе веб-сервиса UTS Marketplace

УДК 004.415.2:004.774

Студент

Группа	ФИО	Подпись	Дата
8ВМ81	Ибетуллов Юрий Юзикович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель ВКР, доцент ОИТ ИШИТР	Цапко Ирина Валериевна	к.т.н.		

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Конотопский Владимир Юрьевич	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД	Горбенко Михаил Владимирович	к.т.н.		

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Кочегурова Елена Алексеевна	к.т.н.		

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Код результата	Результат обучения (выпускник должен быть готов)
	<i>Общепрофессиональные компетенции</i>
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
	<i>Профессиональные компетенции</i>
P5	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
P6	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
	<i>Общекультурные компетенции</i>
P8	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом
P9	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.

Код результата	Результат обучения (выпускник должен быть готов)
P10	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P11	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

**Министерство образования и науки Российской Федерации**  
 федеральное государственное автономное образовательное учреждение  
 высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники  
 Направление подготовки 09.04.01 Информатика и вычислительная техника  
 Отделение информационных технологий

УТВЕРЖДАЮ:  
 Руководитель ООП

\_\_\_\_\_  
 (Подпись)      (Дата)      (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

магистерской диссертации
--------------------------

Студенту:

Группа	ФИО
8BM81	Ибетуллов Юрий Юзикович

Тема работы:

<b>Автоматизация доставки, установки и обновления лицензированного программного обеспечения на базе веб-сервиса UTS Marketplace</b>	
Утверждена приказом директора (дата, номер)	02.06.2020 №154-53/с

Срок сдачи студентом выполненной работы:	01.06.2020
--	------------

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<p><b>Исходные данные к работе</b>  <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объектом исследования является информационная система для автоматизации процессов лицензирования и доставки программного обеспечения на устройства клиентов, его обновления и контроля за исполнением.                  Цель работы состоит в проектировании и разработке системы для автоматизации доставки программного обеспечения на устройства клиентов, его своевременного обновления и контроля.</p>
<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>  <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<p>Выявление цели и задач работы                  Анализ предметной области                  Разработка технического задания                  Проектирование системы                  Реализация информационной системы                  Финансовый менеджмент, ресурсоэффективность и ресурсосбережение                  Социальная ответственность                  Заключение по работе</p>

<b>Перечень графического материала</b> <i>(с точным указанием обязательных чертежей)</i>	Диаграмма компонентов системы Диаграмма развертывания Иллюстрации пользовательского интерфейса Презентация в формате *.pptx
<b>Консультанты по разделам выпускной квалификационной работы</b>	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Конотопский Владимир Юрьевич
Социальная ответственность	Горбенко Михаил Владимирович
<b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>	
<ul style="list-style-type: none"> <li>- 1.1 Автоматизация развертывания программного обеспечения</li> <li>- 1.2 Лицензирование программного обеспечения</li> <li>- 1.4 Обзор существующих аналогов</li> </ul>	

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	16.03.2020
---	------------

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко Ирина Валериевна	к.т.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ВМ81	Ибетуллов Юрий Юзикович		

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Инженерная школа информационных технологий и робототехники  
Направление подготовки 09.04.01 Информатика и вычислительная техника  
Уровень образования магистратура  
Отделение информационных технологий  
Период выполнения весенний

Форма представления работы:

магистерская диссертация (бакалаврская работа, дипломный проект/работа, магистерская диссертация)
--

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН  
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	01.06.2020
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Основная часть	75
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко Ирина Валериевна	к.т.н.		

**СОГЛАСОВАНО:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Кочегурова Елена Алексеевна	к.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ВМ81	Ибетуллов Юрий Юзикович

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение (НОЦ)</b>	<b>Информационных технологий</b>
Уровень образования	Магистратура	Направление/специальность	09.04.01 Информатика и вычислительная техника

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Использовать действующие ценники и договорные цены на потребленные материальные и информационные ресурсы, а также указанную в МУ величину тарифа на эл. энергию
2. Нормы и нормативы расходования ресурсов	—
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Действующие ставки единого социального налога и НДС, ставка дисконтирования = 0,1 (см. МУ)

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Оценка коммерческого и инновационного потенциала НТИ	Дать характеристику существующих и потенциальных потребителей (покупателей) результатов ВКР, ожидаемых масштабов их использования
2. Разработка устава научно-технического проекта	Разработать проект такого устава в случае, если для реализации результатов ВКР необходимо создание отдельной организации или отдельного структурного подразделения внутри существующей организации
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Построение плана-графика выполнения ВКР, составление соответствующей сметы затрат, расчет цены результата ВКР.
4. Определение ресурсной, финансовой, экономической эффективности	Оценка экономической эффективности использования результатов ВКР, характеристика других видов эффекта

**Перечень графического материала (с точным указанием обязательных чертежей):**

1. «Портрет» потребителя результатов НТИ
2. Сегментирование рынка
3. Оценка конкурентоспособности технических решений
4. Диаграмма FAST
5. Матрица SWOT
6. График проведения и бюджет НТИ - выполнить
7. Оценка ресурсной, финансовой и экономической эффективности НТИ - выполнить
8. Потенциальные риски

Дата выдачи задания для раздела по линейному графику	25.02.2020
--	------------

Задание выдал консультант:

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент ОСГН	Конотопский Владимир Юрьевич	к.э.н.		25.02.2020 г.

Задание принял к исполнению студент:

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8BM81	Ибетуллов Юрий Юзикович		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа 8ВМ81	ФИО Ибетуллов Юрий Юзикович
-----------------	--------------------------------

Школа	ИШИТР	Отделение (НОЦ)	Информационных технологий
Уровень образования	Магистратура	Направление/специальность	09.04.01 Информатика и вычислительная техника

Тема ВКР:

Автоматизация доставки, установки и обновления лицензированного программного обеспечения на базе веб-сервиса UTS Marketplace

**Исходные данные к разделу «Социальная ответственность»:**

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является информационная система для автоматизации процессов лицензирования и доставки программного обеспечения на устройства клиентов, его обновления и контроля за исполнением.
--	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p><b>1. Правовые и организационные вопросы обеспечения безопасности:</b></p> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	<ol style="list-style-type: none"> <li>1. Специальные правовые нормы трудового законодательства;</li> <li>2. Организационные мероприятия при компоновке рабочей зоны.</li> </ol>
<p><b>2. Производственная безопасность</b></p> <p>2.1. Анализ выявленных вредных и опасных факторов</p> <p>2.2. Обоснование мероприятий по снижению воздействия</p>	<p><b>Вредные факторы:</b> 1. Отклонения показателей микроклимата; 2. Недостаточная освещенность рабочей зоны; 3. Превышение уровня шума; 4. Повышенный уровень электромагнитных излучений;</p> <p><b>Опасные факторы:</b> 1. Электрический ток; 2. Опасность возникновения пожара.</p>
<p><b>3. Экологическая безопасность</b></p>	<p>Воздействие объекта на атмосферу, гидросферу не происходит. В работе проведён анализ воздействия на литосферу (образование отходов при поломке оборудования).</p>
<p><b>4. Безопасность в чрезвычайных ситуациях:</b></p>	<p>Перечень возможных ЧС. Мероприятия по профилактике и недопущению пожаров.</p>

Дата выдачи задания для раздела по линейному графику	25.02.2020 г.
--	---------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД	Горбенко Михаил Владимирович	к.т.н.		25.02.2020

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ81	Ибетуллов Юрий Юзикович		

## РЕФЕРАТ

Выпускная квалификационная работа содержит 130 стр., 36 рис., 39 источников, 17 таблиц.

Ключевые слова: веб-сервис, лаунчер, программное обеспечение, распространение, обновление, лицензия.

Объектом исследования является методы доставки программного, своевременного обновления и защиты программного обеспечения.

Цель работы состоит в проектировании и разработке системы для автоматизации доставки программного обеспечения на устройства клиентов, его своевременного обновления и контроля.

В процессе работы проводились изучение анализ существующих аналогов разрабатываемой системы, выявление функциональных требований к системе, проектирование и разработка обозначенной информационной системы.

В результате была разработана и внедрена система для автоматизации доставки программного обеспечения на устройства клиентов, его своевременного обновления и контроля.

Область применения: частные предприятия, занимающиеся разработкой программного обеспечения.

Экономическая эффективность/значимость работы заключается в повышении эффективности исполнения рутинных процессов установки программного обеспечения на устройства клиентов и обновления версия программного обеспечения путем сокращения непроизводительных и дублирующих операций, операций, выполняемых «вручную», оптимизации информационного взаимодействия участников процесса. Повышение качества принятия управленческих решений за счет оперативности представления, полноты, достоверности и удобства форматов отображения информации.

В будущем планируется дальнейшая разработка и совершенствование разработанной системы с целью реализации функционала необходимого пользователям системы и возможности предоставления разработанной системы как средства развёртывания приложений другим компаниям.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

В данной работе применены описанные ниже сокращения с соответствующими определениями.

**ПО:** программное обеспечение;

**БП:** бизнес-процесс;

**ИС:** информационная система;

**СУБД:** система управления базами данных;

**БД:** база данных;

**UI:** user interface, пользовательский интерфейс;

В данной работе применены описанные ниже термины с соответствующими определениями:

**API:** application programming interface, описание способов, которыми одна программа может взаимодействовать с другой программой;

**DLC:** downloadable content, форма распространения цифрового медиа контента в виде загружаемого контента;

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	14
1 Анализ предметной области .....	16
1.1 Автоматизация развертывания программного обеспечения .....	16
1.2 Лицензирование программного обеспечения .....	20
1.3 Анализ автоматизируемого бизнес-процесса.....	26
1.4 Обзор существующих аналогов.....	36
1.5 Выводы по разделу .....	40
2 Проектирование системы .....	42
2.1 Функциональные требования .....	42
2.2 Варианты использования.....	45
2.3 Компонентное архитектурное представление .....	56
2.4 Архитектурное представление развёртывания .....	58
2.5 Проектирование пользовательского интерфейса .....	59
3 Реализация системы и анализ результатов .....	63
3.1 Методы и средства реализации .....	63
3.1.1 Язык программирования .....	63
3.1.2 Фреймворки .....	63
3.1.3 Система управления базами данных .....	64
3.2 Серверная часть.....	65
3.2.1 Сервис s-device .....	65
3.2.2 Сервис s-license .....	66
3.3 Клиентская часть.....	68
3.3.1 Страницы в UTS Marketplace.....	68
3.3.2 Утилита UTS Packer.....	71
3.3.3 Клиентское приложение UTS Launcher.....	72
4 Финансовый менеджмент.....	82
4.1 Организация и планирование работ .....	82
4.1.1 Продолжительность этапов работ .....	83
4.2 Расчет сметы затрат на выполнение проекта .....	87
4.2.1 Расчет затрат на материалы .....	87
4.2.2 Расчет заработной платы.....	88
4.2.3 Расчет затрат на социальный налог.....	89
4.2.4 Расчет затрат на электроэнергию .....	89

4.2.5	Расчет амортизационных расходов .....	90
4.2.6	Расчет прочих расходов.....	91
4.2.7	Расчет общей себестоимости разработки .....	91
4.2.8	Расчет прибыли .....	91
4.2.9	Расчет НДС .....	92
4.2.10	Цена разработки ОКР (НИР).....	92
4.3	Оценка экономической эффективности проекта .....	92
5	Социальная ответственность .....	94
5.1	Введение.....	94
5.2	Правовые и организационные вопросы обеспечения безопасности .....	95
5.3	Производственная безопасность .....	96
5.3.1	Анализ вредных и опасных производственных факторов.....	97
5.3.2	Обоснование мероприятий по снижению воздействия.....	107
5.4	Экологическая безопасность.....	108
5.5	Безопасность в чрезвычайных случаях.....	109
5.5.1	Анализ вероятных ЧС, которые могут возникнуть на рабочем месте ....	109
5.5.2	Мероприятия по предотвращению ЧС .....	109
5.6	Выводы по разделу .....	110
	ЗАКЛЮЧЕНИЕ .....	112
	CONCLUSION .....	114
	Список использованных источников .....	116
	Приложение А .....	120

## ВВЕДЕНИЕ

Сегодня, в век бурного развития цифровых технологий, всё больше компаний занимается созданием программного обеспечения. Одним из широко развивающихся направлений является создание программно-аппаратных систем, таких как терминалы самообслуживания, интерактивное игровое оборудование. Одной из таких компаний является ООО «Универсальные терминал системы».

При эксплуатации подобных комплексов основными трудностями являются процессы установки программного обеспечения и его своевременного обновления, контроля над выполнением запущенного программного обеспечения. В большинстве случаев работа специализированных приложений предполагает то, что пользователь не сможет прекратить его работу и выйти в операционную систему.

Отсюда вытекает потребность компании в автоматизации приведенных процессов. Создание и использование подобной системы позволит упростить работу отделов технической поддержки и работы с клиентами.

Цель магистерской работы и задачи состоят в проектировании и разработке системы для автоматизации доставки программного обеспечения на устройства клиентов, его своевременного обновления и контроля. Разработка системы велась для ООО «Универсальные терминал системы» на базе собственной системы цифрового распространения программного обеспечения UTS Marketplace.

Для достижения поставленной цели требуется решить следующие задачи:

- изучение и анализ предметной области, обзор возможных аналогов системы;
- выявление требований к системе;
- проектирование компонентов системы;
- проектирование пользовательского интерфейса;
- реализация намеченных функциональных компонентов системы,
- разработка пользовательского интерфейса системы;

– интеграция разработанного функционала в систему цифрового распространения программного обеспечения UTS Marketplace.

Объектом исследования является автоматизация процессов установки, обновления, запуска и контроля выполнения программного обеспечения. Предметом является автоматизированная система доставки, установки и обновления лицензированного программного обеспечения.

Практическая новизна разрабатываемой системы заключается в объединении воедино функций установки, автоматического фоновое обновления программного обеспечения, запуска специализированных приложений в режиме терминала с ограничением доступа пользователя к функциям операционной системы и контроля выполнения запущенных приложений.

Разработанная система позволит автоматизировать и значительно сократить расходы на процессы подготовки оборудования перед отправкой клиенту, обновления программного обеспечения устройствах клиентов. Позволит сократить расходы на защиту программного обеспечения от копирования.

Реализованная система опробована и применяется в компании ООО «Универсальные терминал системы».

# 1 Анализ предметной области

## 1.1 Автоматизация развертывания программного обеспечения

Развёртывание программного обеспечения — это все действия, которые делают программную систему готовой к использованию. Данный процесс является частью жизненного цикла программного обеспечения [1].

Процесс развертывания приложений часто рассматривается в виде некоей неприятной, но обязательной процедуры, предшествующей выпуску программного продукта. При этом принципы разработки приложений могут и должны применяться в процессе развертывания так же, как и на других этапах создания системы. Ручное развертывание представляет собой повторяющуюся из раза в раз и подверженную ошибкам процедуру. Однако она может быть автоматизирована с целью ускорения выпуска системы и предотвращения ошибок аналогично тому, как автоматизация процесса сборки ускоряет и повышает надежность процесса разработки [2].

К действиям, связанным с развертыванием программного обеспечения, относят:

- **Выпуск (release)** - следует из завершенного процесса разработки и иногда классифицируется как часть процесса разработки, а не процесса развертывания. Он включает в себя все операции по подготовке системы к сборке и передаче в компьютерную систему (системы), на которой она будет запущена в производство. Поэтому иногда это включает определение ресурсов, необходимых для работы системы с приемлемой производительностью, и планирование и/или документирование последующих действий процесса развертывания;

- **Установка и активация** - для простых систем установка программного обеспечения заключается в создании некоторой формы команды, ярлыка, сценария или службы для запуска программного обеспечения (вручную или автоматически). Для сложных систем процесс установки может включать в себя настройку системы - зачастую, задавая конечным пользователям вопросы о

предполагаемом использовании; Активация же заключается в запуске исполняемого компонента программного обеспечения в первый раз.

– **Деактивация** является обратной активацией и означает отключение любых уже выполняющихся компонентов системы. Деактивация часто требуется для выполнения других действий по развертыванию, например, может потребоваться деактивация системы программного обеспечения перед выполнением обновления. Практика удаления редко используемых или устаревших систем из службы часто называется удалением приложений или списанием приложений.

– **Обновление** - процесс обновления направлен на замену более ранней версии всей или некоторой части системы программного обеспечения на более новую версию. Обычно заключается в деактивации с последующей установкой новой версии программного обеспечения;

– **Отслеживание версий** – осуществляется системами отслеживания версий, которые помогают пользователю находить и устанавливать обновления для программного обеспечения;

– **Удаление** является процессом, обратным установке. Заключается в удалении программного обеспечения, которое больше не требуется. Может также включать некоторую реконфигурацию других программных систем для удаления зависимостей удаленной системы.

Современные методы разработки программного обеспечения подразумевают его развертывание в нескольких средах (окружениях). В основном выделяют рабочее (production) и тестовое окружения, хотя для каждого программного продукта могут быть свои окружения, например, окружение разработки (development), отладочное (debug), предрелизное (pre-production), инсценировочное (staging) и прочие, зависящие от конкретного процесса разработки, используемого в команде.

Под рабочим или «боевым» окружением понимается среда, в которой программное обеспечение эксплуатируется реальным пользователем с использованием актуальных данных. К таким средам предъявляется особое

требование – в них должна находиться актуальная и стабильная версия программного продукта, прошедшая через различные стадии тестирования.

Тестовое окружение является противоположностью рабочего окружения. Это окружение необходимо для проверки работоспособности версии программного обеспечения. У рядовых пользователей отсутствует доступ к такому рода окружениям.

Существуют различия между поставкой приложения в рабочую среду и его развертыванием в тестовых средах, не в последнюю очередь связанные с чувством ответственности человека, осуществляющего поставку. Однако с технической точки зрения эти различия должны быть инкапсулированы в наборе конфигурационных файлов. При развертывании в рабочей среде должны выполняться те же процессы, что и при любом другом развертывании [3].

Важной особенностью развертывания программного обеспечения в рабочих средах является возможность отката последней опубликованной версии продукта. Зачастую при выпуске новых версий продуктов случаются ошибки, которые могут привести к неработоспособности программного обеспечения, в виду чего возникнет недовольство клиентов и, возможно, финансовые потери. В общем случае, оптимальной стратегией отката является возврат в рабочую среду предыдущей версии программы, или последней стабильно работающей версии.

Все эти процессы - развертывание в тестовых и рабочих средах и откат развертывания - должны быть частью реализации конвейера развертывания. Необходимо всегда иметь список сборок, доступных для развертывания в каждой из этих сред. Кроме того, автоматический процесс развертывания должен запускаться путем щелчка на кнопке «*Развернуть*» и выбора версии приложения и целевой среды. Фактически, процедура развертывания должна быть единственным доступным способом конфигурирования сред, включая конфигурации операционной системы и приложений сторонних поставщиков. Только тогда можно будет точно выяснить, какая версия приложения и в какой среде развернута, кто выполнил развертывание и какие изменения были внесены в приложение и среду с момента предыдущего развертывания [3].

При этом развертывание новой версии приложения в рабочей среде, которая контролируется вами, и поставка новой версии, устанавливаемой пользователями самостоятельно на своих компьютерах, - существенно разные процессы. В последнем случае необходимо решить ряд дополнительных задач:

- управление процессом обновления;
- перенос двоичных кодов, данных и конфигураций;
- тестирование процесса обновления;
- получение от пользователей отчетов о неполадках.

Существует несколько способов управления обновлениями.

1. Приложение проверяет, появилась ли новая версия. Если да, приложение отображает сообщение для пользователя и предлагает загрузить и установить последнюю версию. Данный способ легче всего реализовать, но наиболее тяжело использовать. Пользователи не хотят сидеть и смотреть на индикатор прогресса.

2. Можно загрузить новую версию в фоновом режиме и отобразить приглашение установить ее. В данном случае приложение периодически проверяет, появилась ли новая версия, и загружает ее, ничего не говоря об этом пользователю. Когда загрузка успешно завершена, приложение предлагает пользователю обновить систему до последней версии.

3. Загрузка новой версии выполняется в фоновом режиме. При следующем запуске приложения оно автоматически обновляется. После этого система может сообщить пользователю об обновлении и попросить перезапустить приложение.

Современная практика показывает, что наиболее эффективным способом обновления приложения является третий вариант. Это связано с тем, что зачастую пользователи откладывают обновления программы, потому что боятся потерять важные данные, накопленные за время работы с ней. Пользователи редко понимают, зачем нужны обновления и в каких случаях их можно отложить на потом. В большинстве случаев их устраивает то, что программа работает без ошибок и не видят смысла обновлять её. Также это связано с тем что, запрашивая

у пользователя подтверждение на установку обновлений разработчики перекладывают на пользователя ответственность за последствия установки новой версии программы.

Правильное решение заключается в том, чтобы сделать процесс обновления надежным и выполнять его автоматически, без согласия пользователей. Если процесс обновления терпит крах, приложение должно автоматически вернуться к прежней версии, сообщив команде разработки о неуспешной попытке обновления. Команда устраняет проблему и развертывает новую версию, которая, хочется надеяться, корректно обновит приложение. Все это должно происходить без участия пользователей. Они могут даже ничего не знать об этом. Единственной веской причиной вовлечения пользователей в процесс обновления может быть только необходимость скорректировать что-либо с их участием.

## **1.2 Лицензирование программного обеспечения**

Любая компания стремится защитить своё программное обеспечение от его неправомерного использования. Как продукт интеллектуальной собственности, программное обеспечение имеет защиту. Распространяется эта защита на законность копирования, изменения программного продукта и оформляется в виде лицензии. Получение лицензии означает согласие со всеми пунктами использования, прописанными в договоре. Любое несанкционированное (без согласования с его автором) использование программного обеспечения – безусловно, деяние противоправное.

Основной документ, определяющий права и обязанности пользователя ПО – это лицензионное соглашение (license agreement). Оно прилагается к приобретенному продукту в бумажном или электронном виде. Именно это соглашение определяет правила использования данного экземпляра продукта.

### **Типы лицензий**

Программное обеспечение делятся на две большие группы [4]:

– свободного использования: бесплатная и открытая лицензия. К бесплатным лицензиям относятся Freeware, GPL, Adware, Postcardware, Donationware, Nagware/Begware, а к открытым относятся Open Source программы с открытым кодом которые можно модифицировать;

– несвободного (коммерческая лицензия) - главная цель таких программ получение прибыли, код программ закрыт;

Кроме того, существует программное обеспечение с условно-бесплатным типом распространения, его можно отнести к обеим группам. Такое программное обеспечение позволяет использовать бесплатно, но некоторыми ограничениями функционала, которое можно снять, заплатив за лицензию. К такому типу лицензий относятся ShareWare, TrialWare, Demoware.

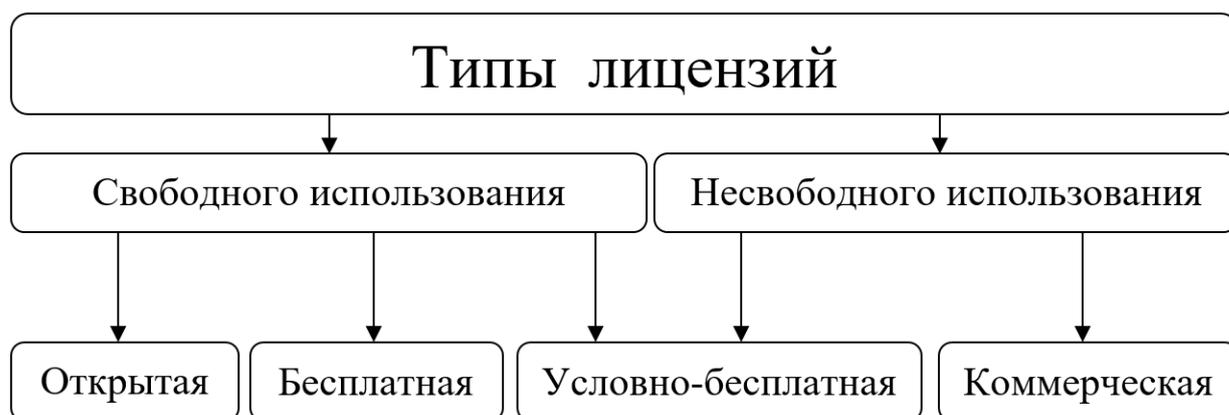


Рисунок 1.1 – Типы лицензий

В российском законодательстве возможные способы использования произведения закреплены в ст. 1270 ГК РФ и включают в себя следующие правомочия (с учетом их применимости к компьютерным программам) [5]:

1. Воспроизведение компьютерной программы, т.е. изготовление одного и более экземпляров, в том числе путем записи компьютерной программы в память компьютера;

2. Распространение компьютерной программы путем продажи или иного отчуждения ее оригинала или экземпляров;

3. Импорт экземпляров компьютерной программы для целей их распространения;

4. Прокат экземпляров компьютерной программы. Владелец исключительного права на компьютерную программу обладает данным правомочием, только если компьютерная программа является основным объектом проката. Если же программное обеспечение является неотъемлемой частью технического устройства, которое и выступает объектом проката (бытовая техника, средства связи и т.п.), то данное правомочие исключено из сферы исключительного права и использование такого программного обеспечения (firmware) не требует заключения отдельного лицензионного соглашения с правообладателем [6];

5. Публичное исполнение компьютерной программы, т.е. ее представление с помощью технических средств в месте, открытом для свободного посещения, или в месте, где присутствует значительное число лиц, не принадлежащих к обычному кругу семьи;

6. Переработка компьютерной программы, под которой понимается любое ее изменение, в том числе перевод такой программы или такой базы данных с одного языка на другой, за исключением адаптации, т.е. внесения изменений, осуществляемых исключительно в целях ее функционирования на конкретных технических средствах пользователя или под управлением конкретных программ пользователя;

7. Доведение компьютерной программы до всеобщего сведения таким образом, что любое лицо может получить доступ к компьютерной программе из любого места и в любое время по собственному выбору (например, размещение ее в сети Интернет).

Различные схемы лицензирования программного обеспечения - это защита своих авторских прав разработчиком. К каждому отдельному виду программного продукта могут быть применены различные типы лицензирования.

В зависимости от объема предоставляемой лицензии в коммерческой практике выделяется множество различных типов лицензий, в частности можно выделить:

1. Однопользовательская лицензия (single-use license). Предоставляет право установки и использования программного продукта одним пользователем на одном устройстве;

2. Лицензия, допускающая неограниченное использование программного продукта в рамках организации лицензиата (enterprise license);

3. OEM-лицензии (original equipment manufacturer) – это тип лицензий, который предоставляется, в основном, производителями компьютеров, которые продают компьютеры с предустановленным программным обеспечением. Основная отличительная особенность OEM-лицензий это то, что они «привязаны» к компьютеру, на который были первоначально установлены, и по условиям лицензии не могут быть перенесены на заменяющий компьютер или на любой другой персональный компьютер;

4. «Коробочная» лицензия (Full Package Product license). Такой тип лицензии относится к способу передачи лицензионного соглашения пользователю и в основном применяется при розничной торговле. При таком способе лицензионное соглашение печатается непосредственно на коробке с программным продуктом;

В зависимости от модели оплаты лицензии, выделяют:

1. Бессрочная лицензия, подразумевает обладание «вечным» правом пользования программным продуктом или, применимо для российского законодательства, на весь срок действия исключительного права [6];

2. Временная лицензия, подразумевает обладание правом пользования, ограниченным установленным временным промежутком (shareware), предоставляя пользователю полноценный доступ к продукту на определённый срок, по истечении которого программа перестаёт запускаться или утрачивает часть функционала;

3. Лицензия «по подписке» (subscription license) предусматривает внесение платежей в установленные временные промежутки, зачастую ежемесячно или ежегодно. Такая модель подписки позволяет разбить стоимость

лицензии на большой временной отрезок, что позволяет получить требуемые функции программного продукта при минимальных начальных тратах;

4. Рекламно-оплачиваемые лицензии (adware) [7] – вид распространения программного обеспечения, при котором пользователю принудительно показывается реклама. Такой принцип подразумевает, что за пользование продуктом платит не пользователь, а рекламодатель, которому предоставляется площадка для размещения рекламы.

### **Контроль за исполнением лицензии**

Современные методы лицензирования программного обеспечения, помимо юридических аспектов включают, как правило, и технические механизмы защиты.

Традиционно, проблема лицензирования техническими средствами решается с помощью специальных программ - администраторов лицензий (license manager), задачей которых является осуществление контроля лицензий (чаще количества пользователей или запущенных приложений) и ограничение доступа нелегальных пользователей. Такой контроль обычно реализуется поиском ключа (идентификатора) пользователя в списке легальных пользователей программного продукта [8].

Другим подходом является мониторинг сетевых приложений. Многие системы (например, Norton Administrator for Network производства Symantec Corp.) осуществляют учет и отслеживание запускаемых приложений, контроль легальности всех пользователей системы и отображение общей картины эксплуатации программного обеспечения на мониторе администратора сети (пакет Hermes производства Microsoft).

Недостатком таких систем (с точки зрения разработчиков программного обеспечения) является то, что контроль за правомерностью пользования программой возлагается на администраторов сетей и зачастую такой способ не защищен от обмана ввиду человеческого фактора.

Современные системы управления лицензированием сетевого программного обеспечения с целью контроля и ограничения числа рабочих станций, одновременно работающих с защищенным приложением [9]:

- сравнивают количество одновременно запущенных копий с допустимым количеством копий согласно лицензионному договору;

- встраивают функции учета в приложения. Идея заключается в том, что при загрузке лицензируемое приложение посылает сообщение по сети. Все копии приложения реагируют на него, посылая ответ рабочей станции, с которой производится загрузка. Рабочая станция производит подсчет ответов и сравнение с общим количеством лицензий;

- контролируют соблюдение лицензий с помощью создания на сервере специальной таблицы лицензий. Каждое запускаемое приложение обращается к таблице. Если лимит лицензий исчерпан, приложение либо не запускается, либо посылает сообщение о необходимости приобретения дополнительной лицензии (Licensing Services for NetWare производства фирмы Novell и другие).

### 1.3 Анализ автоматизируемого бизнес-процесса

Перед оптимизацией или автоматизацией любого процесса требуется его непосредственное изучение, что позволит выявить проблемные места, на которые следует обратить внимание. В этом разделе для повышения наглядности и упрощения анализа бизнес-процесса использовано моделирование бизнес-процесса.

Моделирование бизнес-процессов (БП) – это построение и изучение бизнес-процессов какой-либо организационной единицы (организации или ее отдела) [10]. Является самостоятельной дисциплиной, с помощью которой можно описать предметную область с целью ее анализа и изменения, поиска моментов, которые можно оптимизировать. Кроме того, моделирование БП сегодня часто входит в состав процесса разработки программного обеспечения (ПО). Так, моделирование БП используется для определения автоматизируемых процессов и операций, а также для определения требований к информационным системам.

Моделирование БП практически реализуется в специализированных графических языках, называемых нотациями. Как правило, нотации состоят из блок-схем и правил представления их компонентов для описания исследуемого объекта. В данной работе была использована нотация BPMN (Business Process Modeling Notation). Другими популярными нотациями моделирования БП являются IDEF 0, DFD (data flow diagram – диаграмма потоков работ), EPC (Event-Driven Process Chain – событийная цепочка процессов), VACD (Value added chain diagram – цепочка добавленной стоимости), UML (Unified Modeling Language – унифицированный язык моделирования).

В настоящей работе рассматривается сразу несколько бизнес-процессов с целью их автоматизации. Далее подробно раскрыты бизнес-процессы с присущими им особенностями для организации заказчика выполнения работы – компании ООО «Универсальные Терминал Системы» (УТС), которая является производственной компанией, основным направлением деятельности которой является проектирование и производство сенсорных информационных киосков

и разработка информационно-сервисных приложений для них. Ещё одним направлением деятельности компании является создание интерактивного обучающего и развлекательного оборудования с элементами дополненной реальности.

На данный момент в компании УТС насчитывается более 10 активно развивающихся программно-аппаратных интерактивных продуктов, среди которых интерактивный аэрохоккей IceHook [11], интерактивная песочница iSandBox [12], интерактивный пол Floorium и др. продукты, которые уже зарекомендовали себя как на образовательном, так и на развлекательном рынке. Помимо собственных разработок, компания занимается заказной разработкой программного обеспечения и на данный момент в компании насчитывается более 25 различных заказных разработок.

Для упрощения и автоматизации учёта программного обеспечения, его версии и продаж в компании была разработана информационная система UTS Marketplace, доступная по адресу <https://market.unitsys.ru/>. Данная информационная система позволила организовать единую базу программных продуктов и клиентов компании, автоматизировала процессы продажи продуктов новым клиентам и предоставила клиентам единую точку взаимодействия с компанией и доступ к приобретенным продуктам через личный кабинет клиента.

Кроме описанных выше процессов продажи продуктов, одним из немаловажных процессов является доставка клиенту приобретённого им продукта. Одной из особенностей продуктов компании является то, что все они представляют собой программно-аппаратные комплексы - терминалы самообслуживания, состоящие из специального корпуса, персонального компьютера и сенсоров ввода, в качестве которых могут выступать как сенсорные панели, так и различные пространственные датчики глубины (Kinect 2.0, Orbecc Astra и т.п.). В качестве аппаратной начинки терминала выступают различные программные продукты компании, функционирующие на базе операционных систем семейств Windows и Linux. Наибольшая часть клиентов

компания приобретает продукт в составе аппаратного комплекса и предустановленного на него программного продукта, но также существуют клиенты, приобретающие только программный продукт для уже существующего или изготовленного собственными силами аппаратного комплекса.

В настоящее время в компании задачами предустановки и настройки программных продуктов на устройство клиента занимаются отделы клиентского сервиса и технической поддержки. Но с каждым годом кол-во клиентов и программного обеспечения расчёт. Кроме первичной установки программного обеспечения на устройства клиента, которое осуществляется отделом клиентского сервиса, также необходимо его обновлять и осуществлять техническую поддержку. Со временем обновление программного обеспечения на постоянно растущей базе клиентов становится очень трудоёмким и дорогостоящим процессом, т.к. отдел технической поддержки ограничен в ресурсах и может работать только с малой группой клиентов. Из-за этого становится невозможным своевременное обновление программных продуктов на устройствах клиентов, что приводит к недовольствам и оттоку клиентов или дополнительным растратам. Ведь для обеспечения должного процесса установки и обновления программного обеспечения на устройствах клиентов требуется постоянное увеличение штата сотрудников технической поддержки. Поэтому в компании возникла потребность в автоматизации процессов обновления программного обеспечения на конечных устройствах клиентов.

Кроме этого, компания имеет клиентов по всему свету, что усложняет процесс обновления программного обеспечения на устройствах клиентов. Процесс обновления программного обеспечения осуществляется в ручном режиме с такими средствами удалённого управления компьютером, как Remote Desktop Protocol (RDP) [13] и TeamViewer [14]. Обновление программного обеспечения таким образом приводит к блокировке работы устройства на время его обновления. Также при таком подходе частой проблемой является отсутствие стабильного высокоскоростного интернета у конечного клиента, из-за чего случаются обрывы сеансов связи с устройством.

При всём этом на устройстве должно в безотказном режиме работать установленное на нём программное обеспечение (iSandBox, Floorium, MySelfie или др.), так как разрабатываемые в компании продукты по своей сути являются терминалами самообслуживания. В связи с этим требуется контролировать его исполнение (для перезапуска ПО в случае возникновения ошибок) и ограничивать доступ пользователя к системе, предоставляя доступ только к запущенной в режиме терминала программе. В компании для этих целей существует собственное решение – программная оболочка UTS Shell. Она заменяет системную оболочку и позволяет запускать указанное программное обеспечение в режиме терминала, блокируя доступ к остальным функциям системы. Основной проблемой существующего решения является отсутствие кроссплатформенности ввиду реализации на языке программирования C# с устаревшей кодовой базой. При разработке системы обновления программного обеспечения требуется учитывать описанный выше функционал, так как для обновления (а значит и контроля за исполнением программного обеспечения) требуется иметь доступ к запущенному процессу, чтобы можно было иметь возможность его остановить для обновления и перезапустить после.

Разрабатываемое компанией программное обеспечение также требует защиты - необходимо лицензировать, то есть ограничивать возможности запуска одной копии на разных устройствах. Для этих целей используется платная система защиты от несанкционированного копирования StarForce, разработанная российской компанией Protection Technology [15]. Её использование достаточно дорого, неудобно и, к тому же, она трудно интегрируется в экосистему UTS Marketplace. В связи с этим появилась необходимость разработки собственного функционала защиты от несанкционированного копирования программного обеспечения, который осуществит привязку лицензии продукта к конкретному устройству с хранением и обработкой данных на собственных серверах компании.

В связи с этим появилось чёткое представление о том, что необходимо автоматизировать процессы развертывания и поддержки программного обеспечения, разрабатываемого в компании.

В ходе анализа бизнес-процессов в компании были выделены процессы, автоматизация которых позволит в значительной мере оптимизировать и удешевить их. В таблице 1.1 и на рисунке 1.2 приведена часть бизнес-процесса продажи продукта клиенту, а именно первичная настройка купленного клиентом оборудования. Для упрощения его представления и анализа была опущена уже автоматизированная информационной системой UTS Marketplace часть процесса продажи продукта клиенту. В таблице 1.2 и на рисунке 1.3 приведён процесс обновления программного обеспечения на устройствах клиентов.

Таблица 1.1 - Описание бизнес-процесса настройки оборудования

Подразделение	Выполняемые действия	Используемые ресурсы
Отдел клиентского сервиса	Получение заявки на настройку оборудования для клиента	Сотрудник отдела клиентского сервиса; Номер покупки (order)
Отдел клиентского сервиса	Получение информации об покупке клиентом	Сотрудник отдела клиентского сервиса; Сервис UTS Marketplace; Номер покупки (order);
Отдел клиентского сервиса	Установка необходимого программного обеспечения	Сотрудник отдела клиентского сервиса; Сервис UTS Marketplace;
Отдел клиентского сервиса	Покупка ключей активации для продукта	Сотрудник отдела клиентского сервиса; Сервис StarForce;
Отдел клиентского сервиса	Активация продукта	Сотрудник отдела клиентского сервиса; Ключ активации;
Отдел клиентского сервиса	Регистрация информации об устройстве и версии установленного ПО	Сотрудник отдела клиентского сервиса; TeamViewer ID устройства; Версия продукта;

Отдел технического контроля	Осуществляет технический контроль и направляет оборудование на доставку клиенту	Сотрудник отдела технического контроля; Номер заказа;
Отдел технической поддержки	Настройка оборудования по месту использования и передаёт в пользование клиенту	Сотрудник отдела технической поддержки; TeamViewer ID устройства;

Таблица 1.2 – Описание бизнес-процесса обновления программного обеспечения

Отдел технической поддержки	Реакция на обращения пользователей об ошибках и отказах	Сотрудник отдела технической поддержки;
Отдел технической поддержки	Устраняет ошибки (если не может, то фиксирует проблему и передаёт отделу разработки)	Сотрудник отдела технической поддержки;
Отдел разработки	Разработка новой версии	Сотрудник отдела разработки;
Отдел разработки	Выпускает новую версию и сообщает технической поддержке о решении проблемы	Сотрудник отдела разработки; Сервис UTS Marketplace;
Отдел технической поддержки	Производит ручное обновление ПО для всей базы клиентов	Сотрудник отдела технической поддержки; Сервис UTS Marketplace; TeamViewer ID устройства;

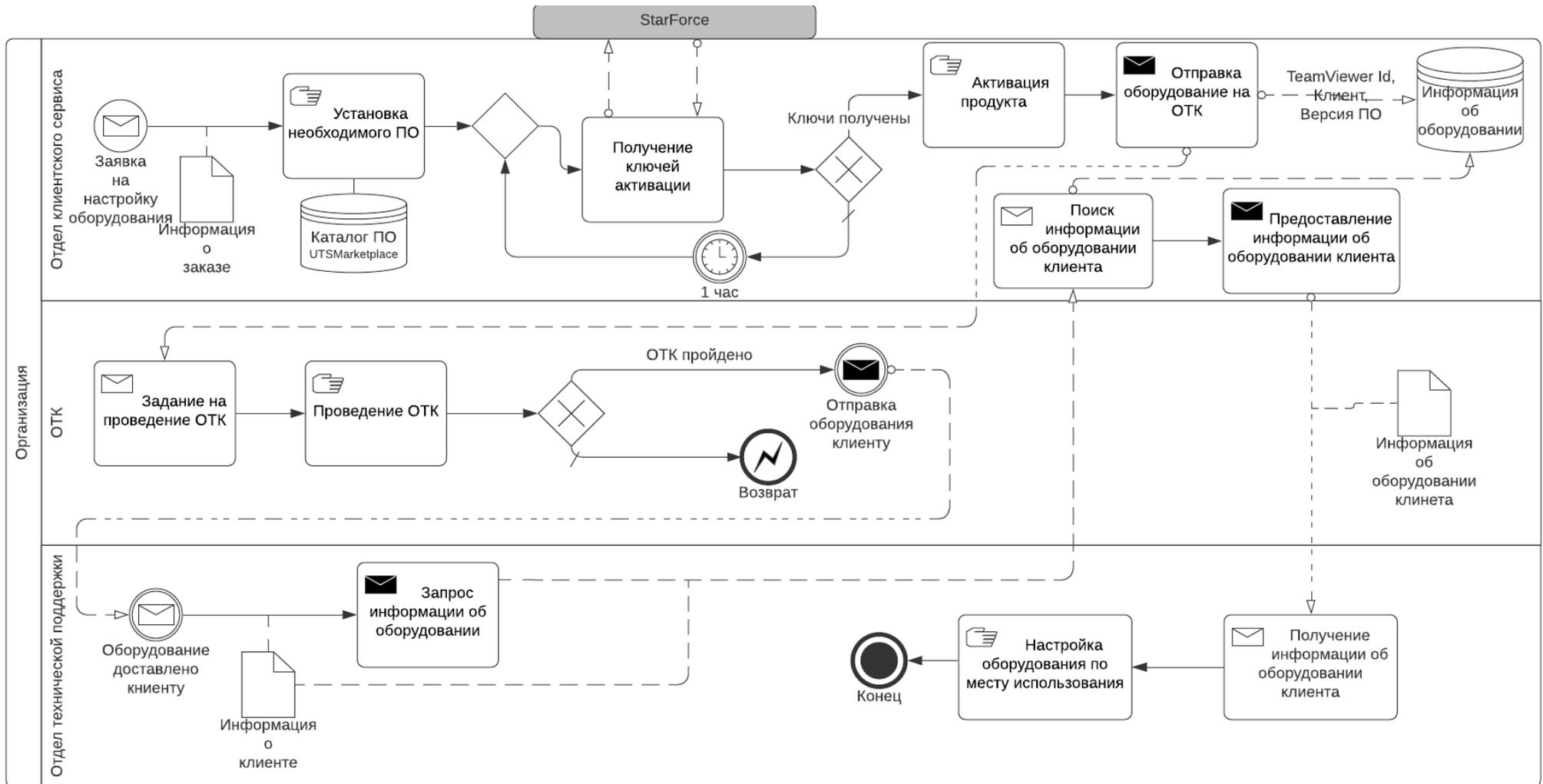


Рисунок 1.2 – Бизнес-процесс начальной настройки оборудования клиента

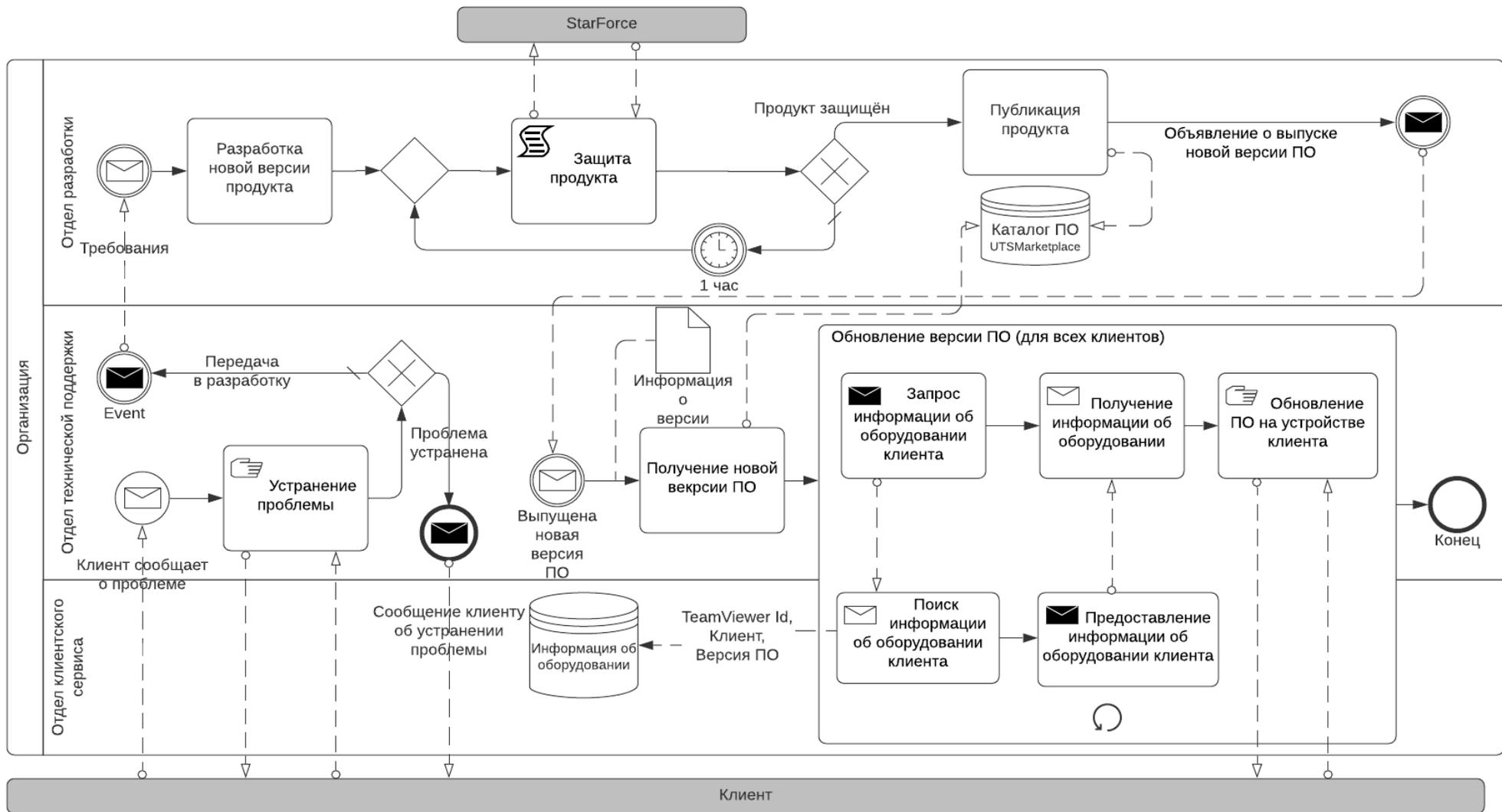


Рисунок 1.3 – Бизнес-процесс обновления программного обеспечения

В результате анализа бизнес-процессов доставки и обновления программных продуктов сделаны следующие выводы:

- Среди отделов компании присуще сложное и неоптимальное взаимодействие – отделы технической поддержки и клиентского сервиса обладают информацией об устройствах клиента и установленном на них программных продуктах, которая децентрализована и требует их непосредственного участия для выяснения этой информации другими отделами;

- При разработке и установке программного обеспечения существует долгая, ненадёжная и требующая участия третьих лиц операция в защите программного обеспечения и получение ключей активации для него на серверах компании StarForce;

- Процесс обновления программного обеспечения на устройствах клиентов осуществляется вручную и требует значительных ресурсов и взаимодействия с третьими лицами – клиентом, который также является ненадёжным звеном и может отказаться от обновления версии ПО, что может привести в дальнейшем к потере им прибыли и возникновению претензий к компании;

- Отдел разработки программного обеспечения не имеет возможности контролировать процессы обновления программного обеспечения на устройствах клиентов;

- Компания не имеет общей базы установленного программного обеспечения и не имеет возможность её собрать из-за участия в процессе продажи посредников (партнёров), которые могут не предоставить информацию о своих клиентах.

По результатам анализа существующих бизнес-процессов и выявления присущих им проблем были определены следующие цели автоматизации рассмотренных бизнес-процессов:

- Замещение существующих разрозненных баз знаний об устройствах клиентов путём создания единой базы устройств клиентов;

- Упрощение процедуры защиты программных продуктов;

– Упрощение процесса начальной настройки устройства путём реализации системы автоматической доставки программного обеспечения на устройства клиента

– Упрощение процесса обновления ПО путём его автоматического обновления без участия отдела технической поддержки;

## 1.4 Обзор существующих аналогов

Для удовлетворения нужд организации требуется программный продукт, объединяющий в себе функционал запуска, контроля исполнения, автоматического обновления программ, управления лицензиями и иметь возможность интеграции с существующим сервисом UTS Marketplace.

В ходе работы были изучены существующие программные продукты со схожим функционалом, которые можно разделить на три группы:

1. Сервисы, предоставляющие управление корпоративной средой. Такие сервисы позволяют компаниям настроить список необходимого программного обеспечения, провести его конфигурацию и распространить необходимое программное обеспечение на всех персональных компьютерах компании. Зачастую такие системы ограничивают пользователя персонального компьютера предоставляя доступ к заранее определённым списку программ. Примером таких сервисов служат: Microsoft System Center [16], Altiris [17] и другие продукты. В рассматриваемой работе такой подход к распространению программного обеспечения не применим в виду отсутствия возможности подключения клиентов к корпоративной сети;

2. Онлайн-сервисы предоставляющие услуги по организации магазинов приложений и сервисов их распространения «под ключ». Основой таких сервисов является подход в предоставлении услуг по организации бизнеса их клиентам (business to business). Рынок онлайн-сервисов, предоставляющих услуги по организации, настройке и сопровождению магазинов приложений, на текущий момент находится на стадии активного развития, но большинство решений, представленных на рынке, являются узкоспециализированными и предоставляют функционал для создания систем распространения программ для мобильных платформ, а именно для устройств под управлением мобильных операционных систем iOS и Android;

3. Онлайн-сервисы цифрового распространения компьютерных игр и программ. Наиболее часто такие сервисы являются узконаправленными на распространение компьютерных игр. Отличительной особенностью таких

сервисов является наличие специальной программы (называемой лайучером - launcher) для загрузки, установки, контроля целостности и лицензии. Существующие площадки дистрибуции можно разделить на два типа: закрытые и открытые площадки. Закрытые площадки не позволяют стороннему пользователю интегрировать в систему собственное программное обеспечение, а предоставляют возможность пользоваться продуктами лишь определенных компаний, в то время как, открытые площадки в основном нацелены на привлечение авторов и публикацию их программного обеспечения и контента, и, зачастую, взимают фиксированный процент с каждой продажи опубликованного пользователем программного продукта.

Среди сервисов второй группы было выделены сервисы компании Ivanti и AppDirect.

**Ivanti** – это компания, предоставляющая услуги по автоматизации бизнес-процессов компаний. Один из продуктов компании - Диспетчер рабочего пространства пользователя (User Workspace Manager) представляет собой решение для управления рабочим пространством, которое помогает упростить развертывание и администрирование настольных систем, сократить расходы и обеспечить безопасность конечных точек. С помощью User Workspace Manager предоставляется возможность доставить персонализированный совместимый рабочий стол независимо от местоположения пользователя или устройства. Решение также облегчает выполнение сложных ИТ-задач, таких как миграция Windows 10, и обеспечивает безопасную синхронизацию файлов, совместное использование и доступ к данным.

Официальный сайт <https://www.ivanti.ru>.

Ценовая политика – существует бесплатный пробный период, предоставляемый по запросу. Услуга предоставляется в виде корпоративная версия для компаний, устанавливаемая на сервер компании. Цена формируется в зависимости от требований по конфигурации с ежемесячной или ежеквартальной оплатой и начинается с 1000\$ в месяц.

**AppDirect** предоставляет единственную комплексную платформу облачной коммерции для достижения успеха в цифровой экономике. Экосистема AppDirect соединяет каналы разработчиков и заказчиков через свою платформу, упрощая цепочку цифровых поставок, позволяя подключать и продавать продукты со сторонними сервисами для любого канала, на любом поддерживаемом устройстве. С помощью AppDirect предоставляется возможность организовать интернет магазин приложений с возможностью установки купленных пользователем приложений на мобильные устройства и персональные компьютеры. Основной акцент у сервиса в организации и поддержке магазина приложений.

Официальный сайт - <https://www.appdirect.com>.

Ценовая политика – доступен бесплатный тариф для ознакомления с возможностями сервиса, ограниченный кол-вом публикуемых товаров (приложений). Доступны корпоративные тарифы, с различным набором опций. Цена формируется в зависимости от требований конфигурации по конфигурации системы.

Среди онлайн-сервисов распространения программного обеспечения были выделены следующие наиболее популярные сервисы: Steam, Epic Games Store, Microsoft Store. Отличительной особенностью этих сервисов является наличие программы клиента, устанавливаемой на персональном компьютере, которая осуществляет загрузку программного обеспечения, проверку лицензии, обновление и предоставляет возможность его запуска.

**Steam** [18] – это цифровой онлайн-сервис для распространения компьютерных игр и программ. Он разработан и поддерживается компанией Valve. Клиент Steam выполняет роль средства технической защиты авторских прав и многие другие функции для многопользовательских игр и потокового вещания, а также социальной сети для игроков. Программный клиент Steam позволяет устанавливать и регулярно обновлять игры, использовать облачные сохранения игр, а так же осуществлять текстовую и голосовую связь между игроками.

Поддержка кроссплатформенности – клиент Steam доступен на всех основных операционных системах: Windows, Linux и macOS.

Официальный сайт <https://store.steampowered.com>.

Ценовая политика: 30% дохода с каждой продажи.

**Epic Games Store** [19] онлайн-сервис цифрового распространения компьютерных игр, разработанный и управляемый американской компанией Epic Games. Также, как и Steam, Epic Games Store выполняет роль средства технической защиты авторских прав, выполняет установку и обновления программ на компьютер пользователя.

Ценовая политика: 12% от каждой продажи, если игра реализована на игровом движке Unreal Engine, то 5%.

Поддержка кроссплатформенности – клиент Epic Games Launcher доступен на операционных системах: Windows и macOS.

**Microsoft Store** – это цифровой магазин приложений, предназначенный для публикации и загрузки универсальных приложений Windows (UMG) и другого цифрового контента [20]. Также возможна публикация классических программ, но для их публикации необходимо их дополнительно упаковать в формат MSIX – формат пакетов приложений Windows [21]. Одной из особенностей является необходимость сертификации публикуемых приложений на их совместимость перед публикацией. Магазин Microsoft предоставляет инструменты разработчика для отслеживания приложений в магазине. Можно отслеживать загрузки, финансы, сбои, отзывы и рейтинги.

Поддержка кроссплатформенности – клиента Microsoft Store доступен только для операционных систем Windows, начиная с Windows 8.

Ценовая политика: 30% от каждой продажи.

Официальный сайт <https://www.microsoft.com/store/apps>.

В ходе анализа аналогов разрабатываемой системы были исследованы функциональные возможности аналогов. В результате сравнения и анализа их возможностей была выявлена невозможность использовать их в целях замены разрабатываемой системы. Из достоинств рассмотренных аналогов можно

выделить наличие обширного функционала, позволяющего тем или иным способом доставлять программное обеспечение на устройства пользователей, обновлять и служить защитой авторских прав. Но все рассмотренные системы обладают рядом недостатков, негативно сказывающих на возможности из внедрения в рассматриваемые бизнес-процессы компании, а именно:

- дороговизна систем, предназначенных для организации собственного магазина приложений и сервисов его распространения;
- узкоспециализированная направленность сервисов дистрибуции программного обеспечения на распространении развлекательного программного обеспечения;
- невозможность адаптации существующих систем под особенности распространения программного обеспечения компанией;
- отсутствие возможности выделения отдельного магазина.

### **1.5 Выводы по разделу**

В результате анализа предметной области было сформировано представление об автоматизируемом бизнес-процессе и предназначении разрабатываемой информационной системы, что позволяет сформулировать функциональные требования к разрабатываемой системе.

По результатам проведённого аналитического обзора аналогов системы можно сделать следующий вывод - среди существующих аналогов нет информационной системы, способной удовлетворить в полной мере потребности компании, следовательно требуется разработать собственное решение, которое позволит автоматизировать рассмотренные бизнес-процессы, что в значительной мере оптимизирует работу отделов клиентского сервиса и технической поддержки и позволит уменьшить издержки компании на процессы доставки клиентам программных продуктов, их обновления и сопровождения.

В качестве метода лицензирования наиболее подходящим вариантом является OEM лицензирование с привязкой лицензии на продукт к оборудованию клиента, ввиду реализации компанией программных продуктов

непосредственно с оборудованием. В таком случае, средство контроля за исполнением лицензионных обязательств возлагается на веб-сервис UTS Marketplace.

Также анализ аналогов позволил изучить функциональные возможности информационных систем дистрибуции программного обеспечения, что позволяет сделать заключение о том, какие компоненты информационной системы требуется разработать. Для достижения поставленных целей требуется;

- разработать кроссплатформенное клиентское приложение (UTS Launcher), которое позволит осуществлять процессы установки, обновления, конфигурирования и контроля правомерности использования программных продуктов на устройствах клиентов, а также регистрации устройства клиента в системе;

- добавить в веб-сервис UTS Marketplace функционал по работе с устройствами и контролем лицензий;

- разработать консольную утилиту для разработчиков (UTS Packer) которая позволит упаковывать программное обеспечение в формат, пригодный для использования в UTS Launcher;

## 2 Проектирование системы

### 2.1 Функциональные требования

Функциональные требования к системе определяют её функциональные возможности, которые определяют, как система реагирует на определённые входные данные, как системы должна вести себя в определённых ситуациях и какие услуги предоставлять пользователю.

Формулировка требований должна ёмкой и в полной мере описывать требуемый пользователю функционал. Для достижения этого используется формулирование требований в виде пользовательских историй (user story) – описание того, что делает пользователь определённой роли для достижения результата на повседневном или деловом языке пользователя [22]. Общая структура описания требований в виде пользовательской истории включает в себя:

- роль пользователя (кто);
- необходимые действия, поведение или функция (что);
- выгода или ценность, которую получает пользователь, когда история реализуется (причина или мотив).

Ниже приведены функциональные требования к разрабатываемой информационной системе.

#### **Функциональные требования веб-сервису UTS Marketplace:**

- Техническая поддержка хочет видеть продукты доступные для установки на устройствах клиента, а также их количество, для понимания какие продукты имеется возможность установить клиенту;
- Техническая поддержка хочет добавлять новые продукты на устройства клиентов, для того чтобы управлять списком установленного программного обеспечения на устройствах;
- Техническая поддержка хочет отменять лицензии на установку продукта у клиента;
- Техническая поддержка хочет видеть лицензии своих клиентов;

- Техническая поддержка хочет добавлять, редактировать и удалять лицензии;
- Техническая поддержка хочет видеть устройства своих клиентов;
- Техническая поддержка хочет, чтобы лицензия автоматически выписывалась при добавлении нового заказа, если имеются доступные продукты и устройства, это позволит сократить время работы над одним устройством;
- Техническая поддержка хочет видеть продукты, сгруппированные по устройствам клиента, для того чтобы понимать какое программное обеспечение установлено на устройствах;
- Техническая поддержка хочет редактировать информацию об устройстве
- Техническая поддержка хочет удалять устройства клиентов;

#### **Функциональные требования к приложению UTS Launcher:**

- Установка программного обеспечения.
- Обновление установленного программного обеспечения.
- Запуск установленного программного обеспечения.
- Контроль выполнения запущенного программного обеспечения.
- Пользователь хочет входить, используя логин и пароль от кабинета UTS Marketplace;
- Пользователь хочет иметь возможность выйти из-под своего пользователя;
- Пользователь хочет видеть информацию о своём пользователе, когда он использует приложение, так он может убедиться, что вошёл под собой;
- Пользователь хочет видеть подробную информацию о своём пользователе на странице профиля, так он сможет узнать информацию, которую он установил или выдала система;
- Пользователь хочет видеть купленные продукты на главной странице, так он может посмотреть список продуктов и дополнений, которые он может установить или те, которые уже установлены;

- Пользователь хочет запускать продукт со страницы продуктов, это позволит ему запустить всё необходимое ПО без нужды разбираться в списке всего ПО;
- Пользователь хочет иметь возможность настроить параметры запуска ПО, это позволит ему удобнее настроить запускаемое ПО;
- Пользователь хочет иметь возможность сменить некоторые настройки работы программы (Launcher), так он сможет изменить поведение ПО для более комфортной работы;
- Пользователь хочет, чтобы файлы настроек ПО сохранялись в процессе обновления, так ему не будет необходимости каждый раз всё настраивать;
- Пользователь хочет, чтобы настройки автозапуска и параметры запуска сохранялись в процессе обновления, так ему не будет необходимости настраивать это каждый раз;
- Пользователь хочет режим работы программы (Launcher) в виде самостоятельной оболочки с автозапуском ПО, это позволит ему использовать приложение как оболочку терминала;
- Пользователь хочет блокировать выход из режима оболочки секретным паролем, это позволит ему не беспокоиться о возможном доступе третьих лиц к устройству во время его отсутствия;
- Пользователь хочет обновления продуктов без прекращения сессии пользователя во время скачивания и установки, это позволит избежать ему простоев оборудования;
- Пользователь хочет, чтобы приложение могло запускать вместе с запуском операционной системы, это позволит ему автоматизировать процесс запуска оборудования;
- Пользователь хочет автоматическое обновление продуктов или добавление новых купленных продуктов, это позволит ему не устанавливать самостоятельно на оборудование;

### **Функциональные требования к приложению UTS Packer:**

– Разработчик хочет единообразный процесс установки для различного программного обеспечения:

– Разработчик хочет гарантировать что опубликованные версии являются подлинными;

## **2.2 Варианты использования**

Варианты использования, или прецеденты – это способ записи требований. Прецедент это что-то, что должна делать система по желанию актёра. Прецедент представляется в виде описания последовательности действий, которые система, подсистемы или класс могут осуществлять, взаимодействуя с внешними актёрами [23].

Так как проектируемая система направлена на оптимизацию работы технической поддержки (включая отдел клиентского сервиса), в этом разделе были расписаны основные прецеденты связанный с работой технической поддержки в нотации UML (таблицы 2.1 - 2.3). Также в этой главе приведены диаграммы прецедентов, которые описывают варианты взаимодействия пользователей системы с её различными частями (рисунки 2.1 - 2.2).

Основное взаимодействие с системой осуществляется через веб-сервис UTS Marketplace и клиентское приложение UTS Launcher.

На рисунках 2.1 и 2.2 изображены варианты использования веб-сервиса UTS Marketplace актёром техническая поддержка (support). Основными действиями технической поддержки являются:

– добавление продукта на устройство пользователя. Это действие включает в себя выбор продукта, который необходимо добавить на устройство клиента, выбор устройства и выбор заказа, в котором клиенту продан продукт;

– удаление продукта с устройства пользователя, включающее в себя выбор лицензии для её удаления;

– просмотр списка лицензий. Включает в себя фильтрацию и поиск лицензий в списке лицензий;

- просмотр списка устройств. Включает в себя фильтрацию и поиск устройств в списке устройств;
- изменение информации об устройстве;
- удаление устройства;
- восстановление устройства.

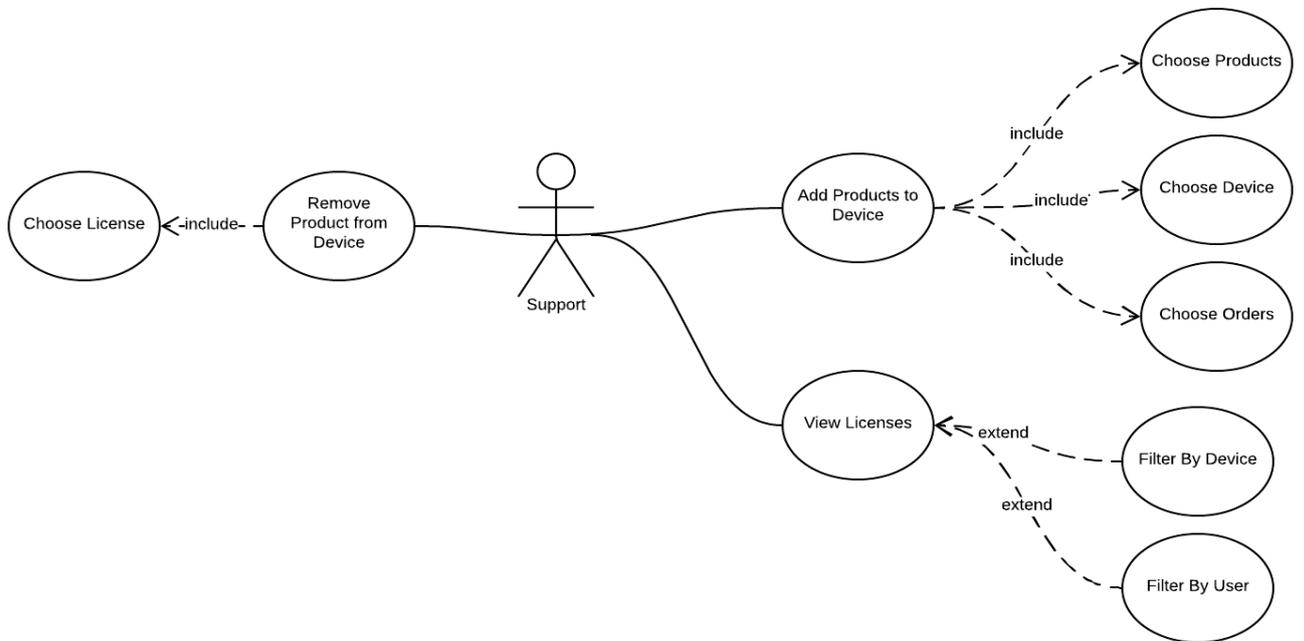


Рисунок 2.1 – Диаграмма вариантов использования UTS Marketplace часть 1

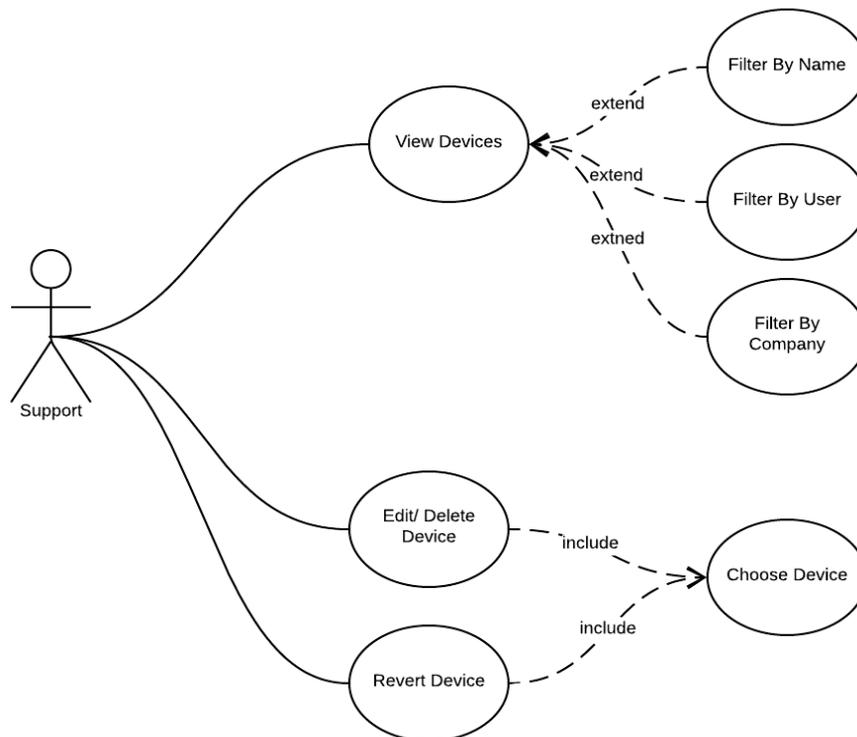


Рисунок 2.2 - Диаграмма вариантов использования UTS Marketplace часть 2

На рисунке 2.3 изображены варианты использования приложения Launcher актёрами Guest (гость) и User (пользователь). Актёр гость может осуществить авторизацию, тем самым стать пользователем и указать адрес сервера, к которому будет осуществляться подключение. Пользователю же доступны следующие действия:

- включение\выключение режима оболочки;
- установка пароля снятия режимы оболочки. Доступно только если не включен режим оболочки;
- изменение настроек приложения;
- выход из системы;
- просмотр статуса подключения к серверам;
- включение\отключение полноэкранного режима;
- обновление приложения;
- просмотр версии приложения.

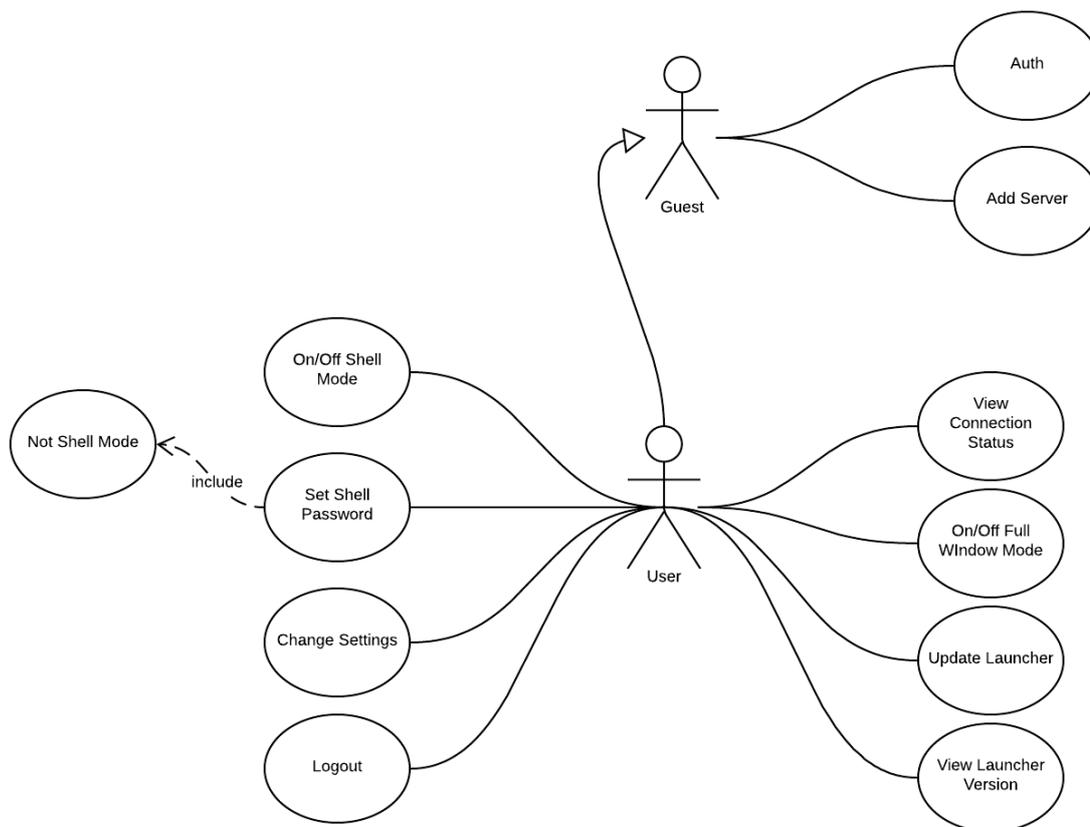


Рисунок 2.3 – Диаграмма вариантов использования приложения Launcher 1

Основные варианты использования приложения заключаются в работе пользователя с приобретёнными им продуктами. На рисунке 2.4 изображены варианты использования приложения при работе с продуктами:

- установка продукта без подключения к интернету;
- просмотр дерева продуктов;
- запуск и остановка исполнения продукта. Требуется выбора продукта в дереве продуктов;
- установка продукта. Требуется выбора продукта в дереве продуктов;
- обновление версии продукта. Требуется выбора продукта в дереве продуктов;
- удаление продукта. Требуется выбора продукта в дереве продуктов;
- просмотр версии продукта. Требуется выбора продукта в дереве продуктов;
- просмотр статуса продукта. Требуется выбора продукта в дереве продуктов;

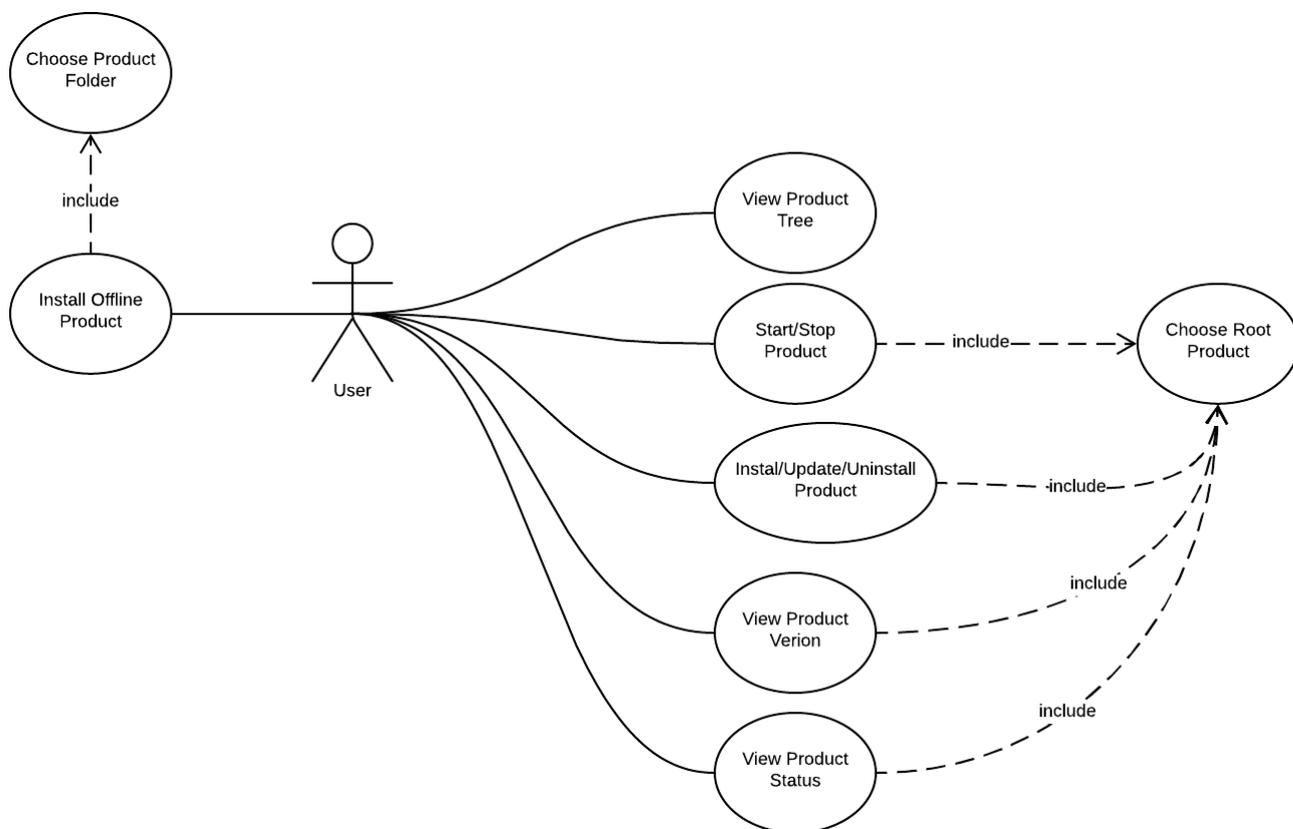


Рисунок 2.4 – Диаграмма вариантов использования приложения Launcher 2

На рисунке 2.5 изображены варианты использования пользователем функционала управления программами и процессами, контролируемые приложением UTS Launcher. Действия, связанные с программами:

- просмотр списка программ. Список программ формируется после установки продукта. Продукт может включить в себя от одной исполняемой программы;

- добавление и удаление сторонней программы. Помимо добавления программы путём установки продукта, пользователю доступно добавление любой сторонней программы путём указания пути до её исполняемого файла;

- изменение параметров (аргументов) запуска программы;

- добавление и удаление программы из списка автозапуска. Это действие позволяет определить какие программы из установленного продукта следует запускать в автоматическом режиме;

- запуск и остановка программы;

- просмотр статуса программы.

Действия, связанные с процессами (под процессом понимается идентификатор исполнения программы в системе):

- просмотр списка процессов;

- остановка и запуск процесса.

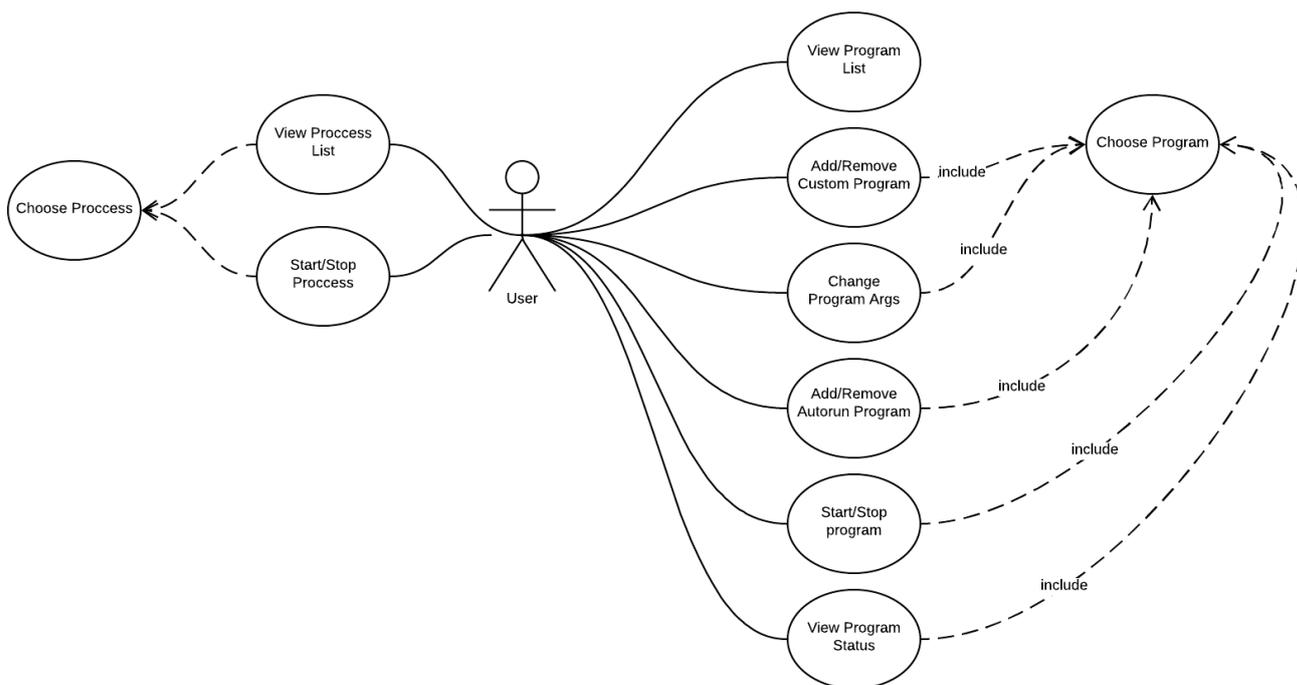


Рисунок 2.5 – Диаграмма вариантов использования приложения Launcher 3

После установки пользователем необходимых продуктов и настройки запуска программ устройство (терминал) должно перейти в рабочий режим. Управление работой терминала осуществляет администратором, который обеспечивает безотказную работу терминала. С целью управления работой терминала администратор может выйти из режима оболочки (разблокировать), перезагрузить терминал, закрыть приложение, переключить приложение в режим конфигурирования, перевести приложение в режим оболочки (заблокировать), остановить автозапуск приложений.

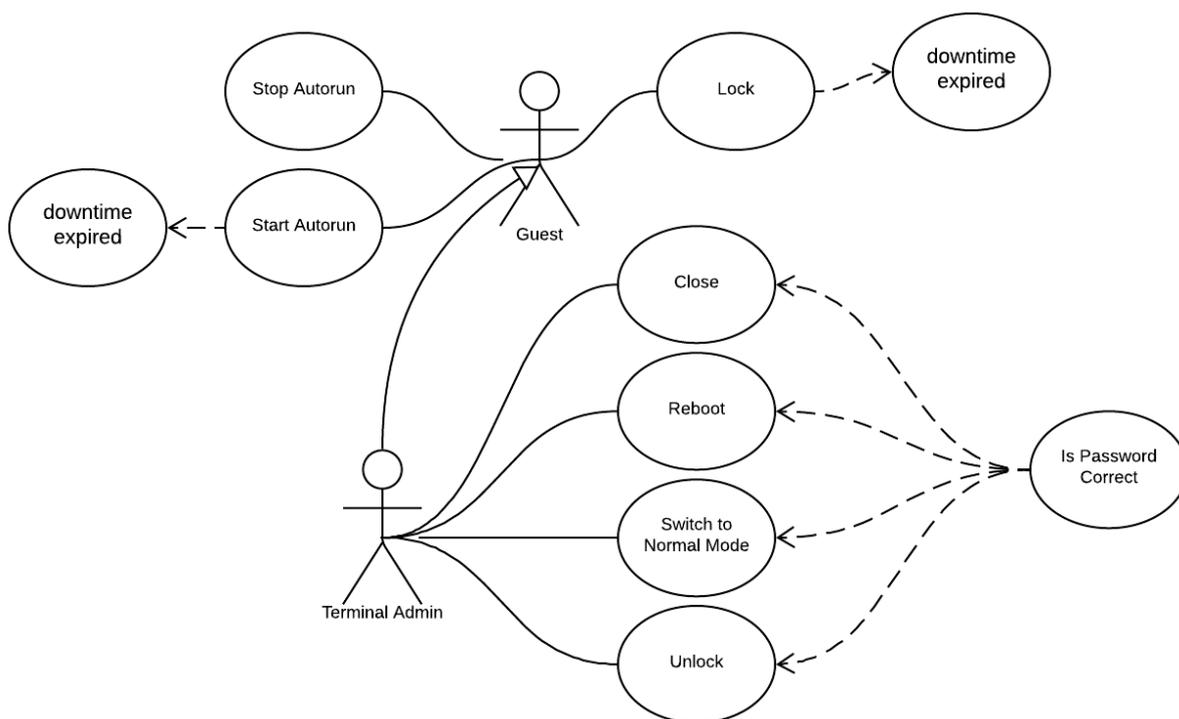


Рисунок 2.6 – Диаграмма вариантов использования приложения Launcher 3

Наиболее значимыми вариантами использования системы, помимо описанных выше вариантов использования, для автоматизации бизнес-процессов компании являются варианты использования системы по осуществлению доставки продуктов на устройства пользователя. Далее приведено подробное описание прецедентов по первоначальной настройке устройства клиента и добавлению продукта на устройство пользователя. Эти прецеденты включают в себя варианты использования, приведённые на рисунках выше, описывающие варианты взаимодействия различных актёров с системой.

Одним из постоянно используемых прецедентов является первоначальная настройка оборудования клиента перед его отправкой клиенту. В таблице 2.1 описан прецедент, когда устройство клиента находится в непосредственной доступности специалиста технической поддержки.

Вторым прецедентом является необходимость настройки оборудования, которое уже находится у клиента и к нему нет физического доступа у специалиста технической поддержки (таблица 2.2). В таком случае необходимо подключаться к оборудованию средствами удалённого администрирования (в данном случае используется программа Team Viewer). Такой прецедент

возникает при необходимости перевода «старых» клиента на использование новой информационной системы.

Третьим частым прецедентом является установка нового продукта на устройство клиента, который уже есть в системе. Данный прецедент приведён в таблице 2.3.

Таблица 2.1 – Прецедент начальной настройки оборудования

ID: 1
<p>Краткое описание:</p> <p>Техническая поддержка намеревается произвести начальную установку, настройку продуктов и всего программного обеспечения на оборудование клиента перед отправкой</p>
<p>Актёры:</p> <p>Техническая поддержка (Support)</p>
<p>Предусловия:</p> <p>Техническая поддержка имеет аккаунт в системе UTS Marketplace          Аккаунт технической поддержки обладает необходимыми правами доступа          Клиент зарегистрирован в системе          Техническая поддержка обладает всей необходимой информацией об аккаунте клиента          В системе зарегистрирована продажа продуктов клиенту          В наличии оборудование с предустановленной ОС и необходимыми драйверами</p>
<p>Основной поток:</p> <p>Прецедент начинается, когда технической поддержке поступает задача установки программного обеспечения на устройство клиента</p> <ol style="list-style-type: none"> <li>1. Загрузить последнюю версию приложения Launcher на устройство клиента</li> <li>2. Установить приложения Launcher</li> <li>3. Запустить приложения Launcher</li> <li>4. Зарегистрировать устройство в системе             <ol style="list-style-type: none"> <li>a. Авторизоваться в Launcher с учётной записью клиента</li> <li>b. Запомнить идентификатор устройства со страницы профиля пользователя</li> </ol> </li> <li>5. Выписать лицензии на устройство             <ol style="list-style-type: none"> <li>a. Зайти со своего рабочего места в Marketplace с использованием учётной записи технической поддержки</li> <li>b. Выписать лицензии на устройство клиента по идентификатору устройства на нужные продукты в Marketplace</li> </ol> </li> <li>6. Установить продукты в Launcher             <ol style="list-style-type: none"> <li>a. Нажать установить продукты в Launcher</li> <li>b. Дождаться окончания установки</li> </ol> </li> <li>7. Настройка запуска продуктов             <ol style="list-style-type: none"> <li>a. Добавить в автозапуск ПО на странице программ в Launcher</li> <li>b. Добавить в автозапуск дополнительное ПО на странице программ в Launcher</li> </ol> </li> <li>8. Включить автозапуск Launcher</li> <li>9. Переключить Launcher в режим оболочки</li> <li>10. Перезагрузить устройство и проверить работоспособность</li> </ol>
<p>Постусловия:</p> <p>Устройство клиента настроено и готово к работе.</p>
<p>Альтернативные потоки:</p> <p>Устройство уже настроено.</p>

Таблица 2.2 - Прецедент начальной настройки оборудования на устройстве клиента

ID: 2
<p>Краткое описание:</p> <p>Техническая поддержка намеревается произвести начальную установку, настройку продуктов и всего программного обеспечения на оборудовании, которое уже у клиентов</p>
<p>Актёры:</p> <p>Техническая поддержка (Support)</p>
<p>Предусловия:</p> <p>Техническая поддержка имеет аккаунт в системе UTS Marketplace          Аккаунт технической поддержки обладает необходимыми правами доступа          Клиент зарегистрирован в системе          Техническая поддержка обладает всей необходимой информацией об аккаунте клиента          В системе зарегистрирована продажа продуктов клиенту          Наличие Team Viewer на устройстве клиента</p>
<p>Основной поток:</p> <ol style="list-style-type: none"> <li>1. Зайти на устройство клиента по Team Viewer</li> <li>2. Отключить и удалить старое ПО</li> <li>3. Скачать последнюю версию Launcher из Store</li> <li>4. Установить Launcher</li> <li>5. Запустить Launcher</li> <li>6. Зарегистрировать устройство в системе             <ol style="list-style-type: none"> <li>a. Вести логин пароль пользователя и войти в Launcher</li> <li>b. Посмотреть ID устройства в Profile в Launcher</li> </ol> </li> <li>7. Выписать лицензии на устройство в системе             <ol style="list-style-type: none"> <li>a. Зайти со своего рабочего места в Marketplace под своим пользователем</li> <li>b. Выписать лицензии на устройство клиента по ID на нужные продукты в Marketplace</li> </ol> </li> <li>8. Установить продукты в Launcher             <ol style="list-style-type: none"> <li>a. Нажать установить продукты в Launcher</li> <li>b. Дождаться окончания установки</li> </ol> </li> <li>9. Настройка запуска продуктов             <ol style="list-style-type: none"> <li>a. Добавить в автозапуск ПО на странице программ в Launcher</li> <li>b. Добавить в автозапуск дополнительное ПО на странице программ в Launcher</li> </ol> </li> <li>10. Включить автозапуск Launcher</li> <li>11. Переключить Launcher в режим оболочки</li> <li>12. Перезагрузить устройство и проверить работоспособность</li> </ol>
<p>Постусловия:</p> <p>Устройство клиента настроено и готово к работе.</p>
<p>Альтернативные потоки:</p> <p>Устройство уже настроено.</p>

Таблица 2.3 - Прецедент добавления продукта

ID: 3
<p>Краткое описание:          Клиент купил новый продукт и техническая поддержка намеревается добавить его на устройство ему</p>
<p>Актёры:          Техническая поддержка (Support)</p>
<p>Предусловия:          Техническая поддержка имеет аккаунт в системе Marketplace          Аккаунт технической поддержки обладает необходимыми правами доступа          Клиент зарегистрирован в системе          Техническая поддержка обладает всей необходимой информацией об аккаунте клиента          В системе зарегистрирована продажа продуктов клиенту          Support имеет информацию об устройстве, к которому требуется привязать продукты          Support имеет информацию о продукте, который приобрёл клиент</p>
<p>Основной поток:</p> <ol style="list-style-type: none"> <li>1. Зайти в Marketplace</li> <li>2. Добавить лицензию             <ol style="list-style-type: none"> <li>a. Вывести список устройств клиента по его имени</li> <li>b. Добавить продукт на необходимое устройство клиенту</li> </ol> </li> <li>3. Дождаться автоматической установки продукта на устройство клиента</li> </ol>
<p>Постусловия:          Клиенту установлен купленный им продукт на его устройство.</p>
<p>Альтернативные потоки:          Установка вручную с подключением через Team Viewer.</p>

## 2.3 Компонентное архитектурное представление

На рисунке 2.7 приведена диаграмма компонентов системы, отражающая высокоуровневую структуру системы.

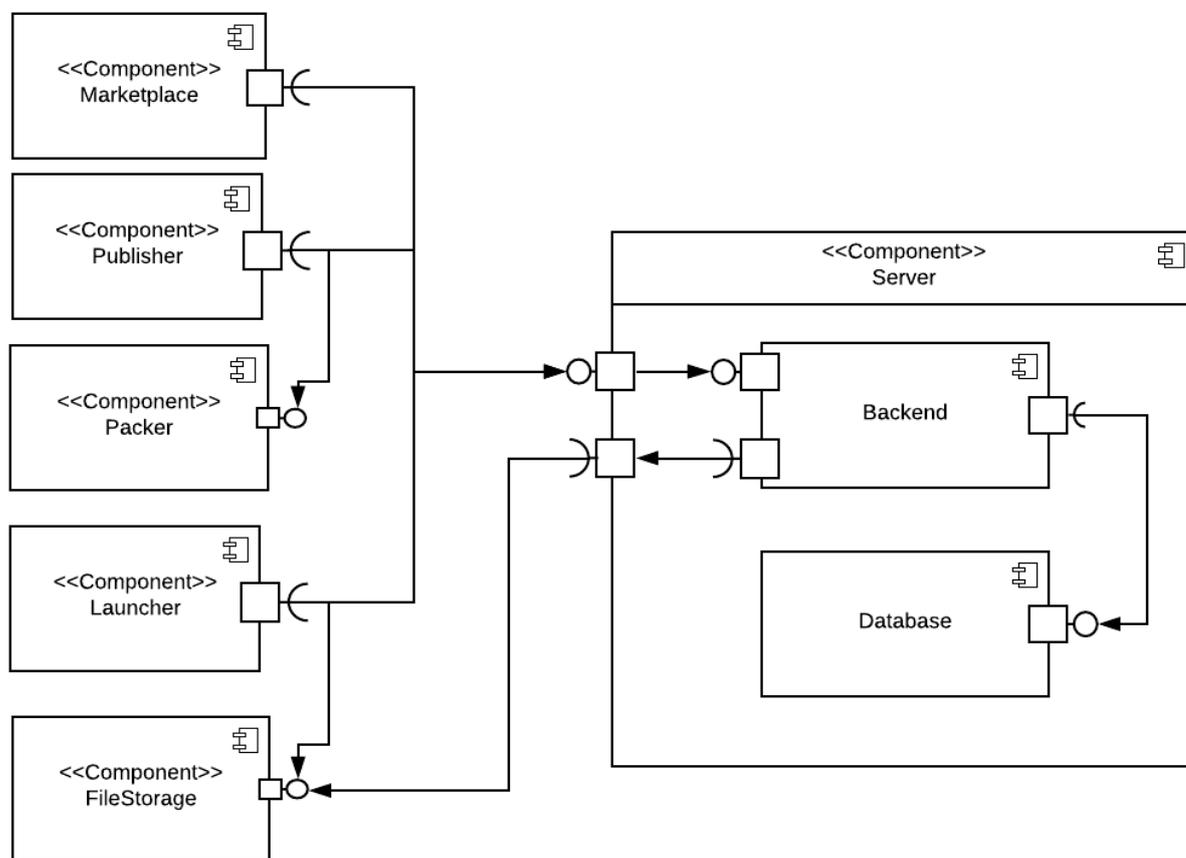


Рисунок 2.7 - Диаграмма компонентов системы

Компонент «Server» представляет собой компонент сервера, на котором развёртываются следующие компонентные части:

- Компонент «Backend», который представляет собой набор микросервисных приложений, осуществляющих логику работы веб-сервиса UTS Marketplace;
- Компонент «Database», который представляет собой установленную на сервере СУБД и базу данных, в которой хранится вся информация веб-сервиса UTS Marketplace.

Компонент «Marketplace», который представляет собой веб приложение, осуществляющее доступ пользователя к системе через предоставленный графический интерфейс путем взаимодействия с приложениями компонента

«Backend». Оно включает в себя личный кабинет пользователя и панель администрирования системы.

Компонент «FileStorage», который представляет собой хранилище файлов, таких как пакеты установки приложений и различный контент сервиса UTS Marketplace.

Компонент «Packer», который представляет собой консольное приложение, направленное на создание пакетов установки программного обеспечения.

Компонент «Publisher», который представляет собой консольное приложение, которое осуществляет доставку пакета установки программного продукта в компонент файлового хранилища «FileStorage» и осуществляющее регистрацию версии продукта в базе данных «Database», посредством взаимодействия с приложениями компонента «Backend».

Компонент «Launcher», который представляет собой клиентское настольное приложение, которое, как и веб-приложение Marketplace осуществляет взаимодействие с приложениями сервиса «Backend» для авторизации пользователя, получения списка продуктов, их версий и списка лицензий пользователя. Осуществляет получение пакета из файлового хранилища с целью его установки и запуска. Осуществляет контроль за исполнением, обновление и конфигурированием программ.

В рамках настоящей работы производилась разработка компонентов Launcher, Packer и доработка компонентов Backend и Marketplace. Компоненты Publisher и FileStorage уже были реализованы в рамках веб-сервиса UTS Marketplace и не требуют доработок.

## 2.4 Архитектурное представление развёртывания

Диаграмма развёртывания определяет физическое оборудование, на котором будет выполняться программная система, а также описывает, как программное обеспечение развёртывается на это оборудование [23]. На рисунке 2.8 приведена диаграмма развёртывания системы.

Информационная система состоит из следующих узлов:

- Storage Server – узел, представляющий собой сервер хранения файлов, куда размещаются пакеты приложений;
- Target Device – узел, представляющий собой устройство клиента на котором развёртывается приложение UTS Launcher и приобретённые клиентом программные продукты;
- Marketplace Docker Server – узел, представляющий собой серверную часть системы. Здесь развёртываются необходимые для работы сервисы и базы данных;
- Client – узел, представляющий собой устройство клиента через которое осуществляется доступ к веб сайту;
- Build Machine – узел, представляющий собой устройство осуществляющее процессы непрерывной интеграции, а именно сборка программных продуктов из исходных кодов, выполнение тестов и упаковка программных продуктов в пакеты, для дальнейшего распространения.

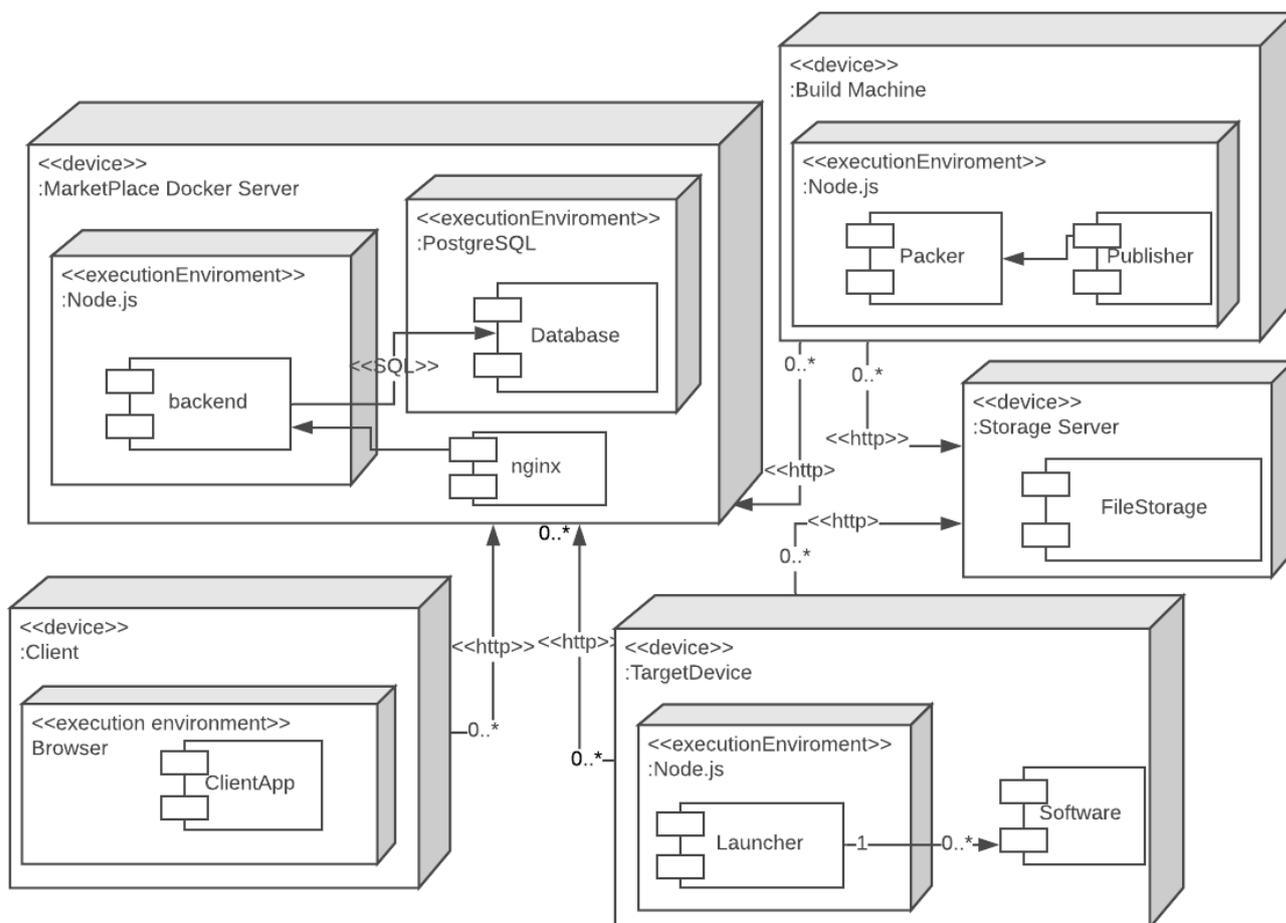


Рисунок 2.8 – Диаграмма развертывания

## 2.5 Проектирование пользовательского интерфейса

Проектирование пользовательских интерфейсов может выполняться в виде скетчей – набросков от руки, вайрфреймов – схематичных изображений, мокапов – макет интерфейса с детальной проработкой или прототипов – интерактивный вариант вайрфрейма позволяющий оценить дизайн в динамике [24].

В работе был использован подход вайрфреймов потому, что для реализации системы не требуется сложный графический интерфейс и достаточно схематично изобразить основные элементы управления и их взаимное расположение на страницах приложения.

На рисунке 2.9 показан вайрфрейм для страницы веб-сайта UTS Marketplace. В левой части экрана располагается список продуктов, которые можно добавить на устройства, список которых расположен в виде сетки в

центре страницы. Рядом с каждым продуктом указывается кол-во их возможных установок. Карточка устройства состоит из заголовка с названием устройства и списком установленных на него продуктов.

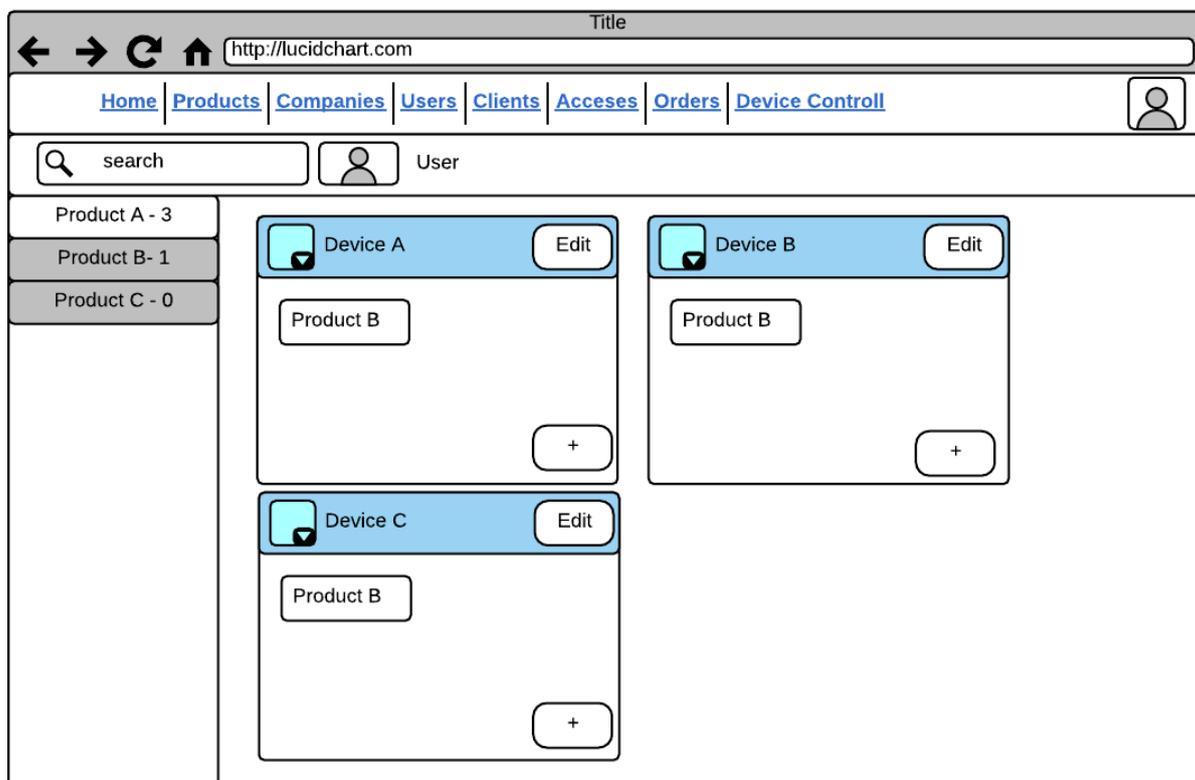


Рисунок 2.9 – Страница Smart Licenses в UTS Marketplace

На рисунке 2.10 представлен проект внешнего вида страницы продуктов пользователя в приложении Launcher. Слева отображается список доступных продуктов, в центральной части экрана отображается информация об выбранном продукте и элементы управления продуктом.

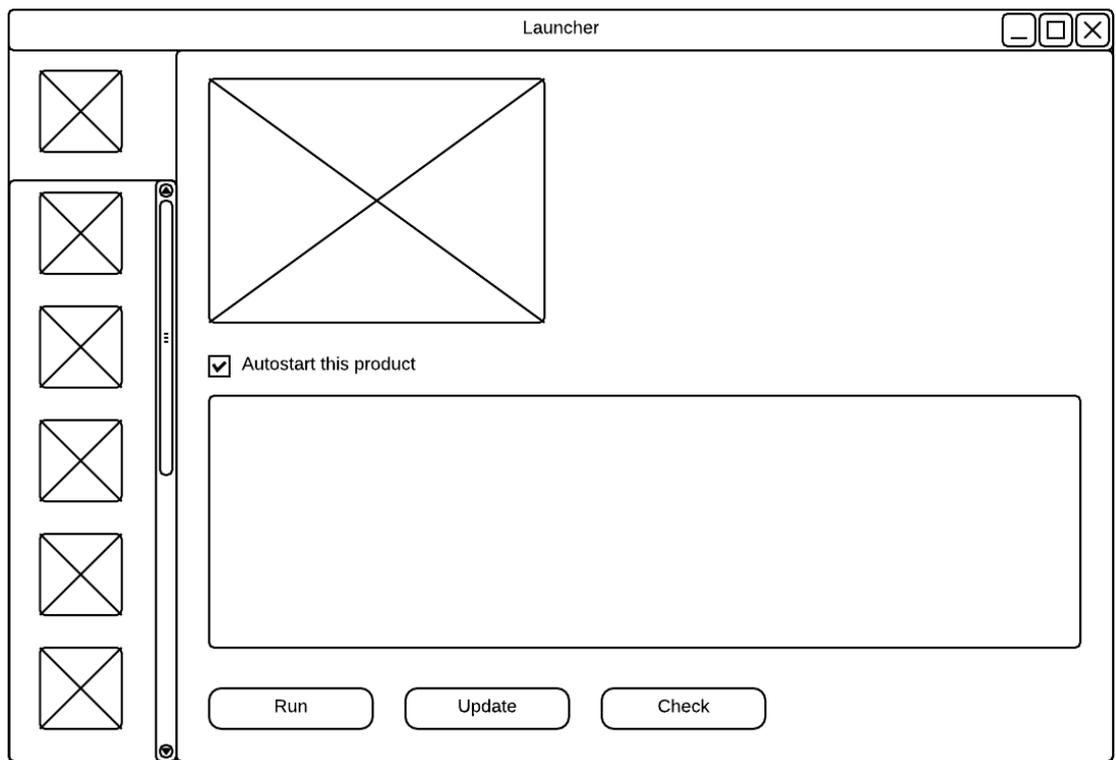


Рисунок 2.10 – Страница продуктов

На рисунке 2.11 представлен внешний вид страницы управления установленными программами. В центральной части экрана отображается список программ с основной информацией о них и элементами управления.

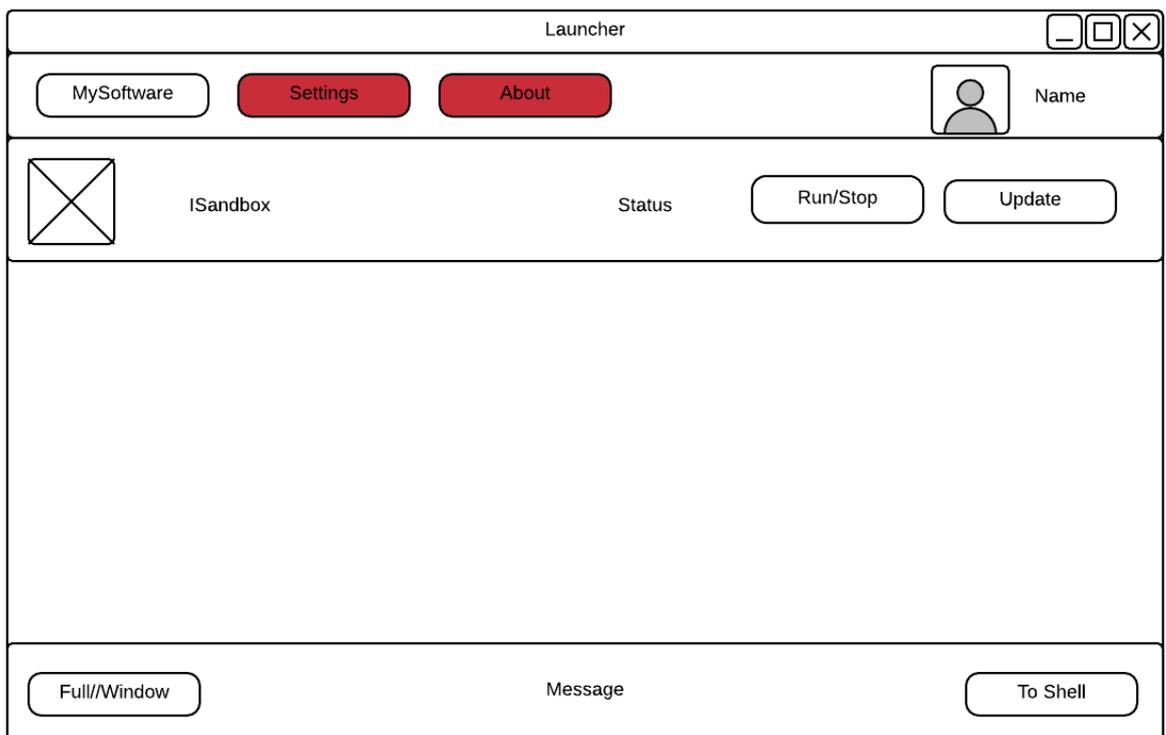


Рисунок 2.11 – Страница программ

На рисунке 2.12 изображены вайрфреймы работы программы Launcher в режиме оболочки. В режиме оболочки программа развёрнута на весь экран и блокирует доступ пользователя к системе. В центре экрана присутствуют элементы управления, позволяющие разблокировать снять блокировку с устройства или продолжить её.

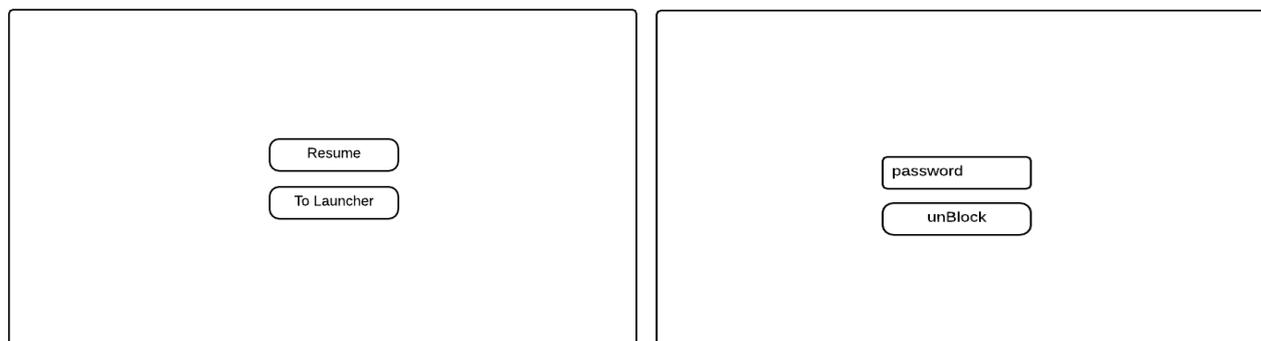


Рисунок 2.12 – Окно оболочки и её разблокировки

## 3 Реализация системы и анализ результатов

### 3.1 Методы и средства реализации

#### 3.1.1 Язык программирования

Для реализации веб-сервисов, s-device и s-license, и консольного приложения UTS Packer был использован язык программирования JavaScript.

Из-за того, что JavaScript является динамически типизированным языком, разработка сложных систем с большим кол-вом данных на нём является затруднительной. С целью реализации контроля за типами данных для реализации приложения UTS Launcher был использован язык программирования TypeScript.

**JavaScript** – язык программирования, являющийся прототипно-ориентированным. Особенность языка является возможность в короткий срок приступить к разработке программ на нём, благодаря низкому порогу вхождения и простоте его понимания.

Преимущества JavaScript:

1. Обширная поддержка браузерами;
2. Лёгкость в освоении;
3. Полезные функциональные настройки;
4. Быстро развивающийся и постоянно совершенствующийся язык;

**TypeScript** — язык программирования, позиционируемый как средство разработки веб-приложений, расширяющий возможности языка JavaScript. Особенность TypeScript в том, что это строго типизированный и компилируемый язык. Строгая типизация уменьшает количество потенциальных ошибок при разработке ПО, которые могли бы возникнуть при разработке на JavaScript.

#### 3.1.2 Фреймворки

Основным способом разработки приложений на JavaScript является использование среды исполнения Node.js. Для реализации графического интерфейса в веб-сервисе UTS Marketplace и приложении UTS Launcher был использован фреймворк React, а для реализации хранения состояний перехода

пользователя по страницам приложения фреймворк Redux. Реализовать кроссплатформенное настольное приложение UTS Launcher на JavaScript позволил фреймворк Electron.

**Node.js** — программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера.

**React** - JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React может использоваться для разработки одностраничных и мобильных приложений.

**Redux** - библиотека для JavaScript с открытым исходным кодом, предназначенная для управления состоянием приложения. Чаще всего используется в связке с React или Angular для разработки клиентской части. Содержит ряд инструментов, позволяющих значительно упростить передачу данных хранилища через контекст.

**Electron** - фреймворк, разработанный GitHub. Позволяет разрабатывать нативные графические приложения для настольных операционных систем с помощью веб-технологий. Фреймворк включает в себя Node.js для работы с back-end и библиотеку рендеринга из Chromium.

### 3.1.3 Система управления базами данных

Наиболее распространённым и бесплатным в использовании средством управления базами данных в среде веб-разработки является СУБД PostgreSQL, которая была использована в веб-сервисе UTS Marketplace.

В отличие от веб-сервиса для настольного приложения важна простота развертывания и отсутствие лишних зависимостей. С этой целью для хранения состояний приложения UTS Launcher была использована СУБД SQLite.

**PostgreSQL** - это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных.

**SQLite** - компактная встраиваемая система управления базами данных. SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite.

## 3.2 Серверная часть

Для реализации требуемого функционала потребовалось расширить функционал веб-сервиса UTS Marketplace. Для этого было реализовано два микро-сервиса, обеспечивающие контроль за устройствами и лицензиями.

Так как используется микро-сервисная архитектура, то для хранения данных в каждом отдельном микро-сервисе требуется использовать свой независимый экземпляр базы данных. В таком случае в базе содержится всего одна таблица, что позволяет обеспечить высокую автономность таких микро-сервисов.

### 3.2.1 Сервис s-device

Данный сервис создан для управления списком устройств клиентов. Для этого были реализованы следующие функции, доступные для вызова по API:

- создание устройства;
- получение информации об устройстве;
- получение списка устройств;
- обновление информации об устройстве;

- удаление устройства.

На рисунке 3.1 приведена схема таблицы device, используемой для представления информации об устройстве. Для надёжной идентификации устройства в базу заносится информация об идентификаторах материнской платы устройства и его процессоре. Для возможности идентификации пользователем каждое устройство имеет имя и дату его регистрации. Привязка устройством осуществляется к компании, а не к определённому пользователю, так как в компании может быть множество различных специалистов с разными правами доступа, которые могут взаимодействовать с устройством. Для защиты данных об устройстве и возможности безопасно передать информацию об лицензии на устройство в описании устройства присутствуют ключи шифрования – приватный и публичный.

device		
pk	id	serial
	company_id*	integer
	motherboard_id*	text
	processor_id*	text
	create_datetime*	datetime [now()]
	last_update_datetime*	datetime [now()]
	name	text
	public_key*	text
	private_key*	text
	body*	json [{}]

Рисунок 3.1 – Схема базы данных сервиса s-device

### 3.2.2 Сервис s-license

Данный сервис предназначен для управления лицензиями на установку продуктов, приобретаемых клиентом. Для этого в сервисе были реализованы следующие функции, доступные для выполнения через API:

- получение информации об лицензии;
- получение списка лицензий;
- создание лицензии;
- удаление лицензии.

На рисунке 3.2 приведена схема таблицы license базы данных сервиса s-license, в которой представлена информация об лицензии. Каждая лицензия обязана ассоциироваться с продуктом, компанией, которая приобрела лицензию на продукт, заказом, который и устройством, на котором будет установлен продукт. Помимо этого, важными атрибутами лицензии являются её тип, тариф, по которому она был приобретена, и дата её окончания.

license		
pk	id*	serial
	company_id*	integer
	order_id*	integer
	product_id*	integer
	device_id	integer
	tariff*	json
	type*	varchar(40) ['default']
	create_datetime*	datetime [now()]
	expire_datetime*	datetime
	body*	json ['{}']

Рисунок 3.2 - Схема базы данных сервиса s-license

## 3.3 Клиентская часть

### 3.3.1 Страницы в UTS Marketplace

Для реализации функционала выписки лицензии на продукт и привязки к устройствам в Marketplace были добавлены новые страницы:

- Smart Licenses – страница привязки лицензий к устройствам;
- Devices – страница управления устройствами;
- Licenses – страница управления лицензиями.

Страница Smart Licenses (рисунок 3.3) создана для упрощения привязки лицензии на продукт к устройству. На странице слева расположен список продуктов, принадлежащих выбранному в верхней части страницы пользователю. Справа же в табличном виде приведены устройства, принадлежащие выбранному пользователю. Перетаскиванием продукта из списка на карточку устройства автоматически производится выписка лицензии и привязка ее к выбранному устройству.

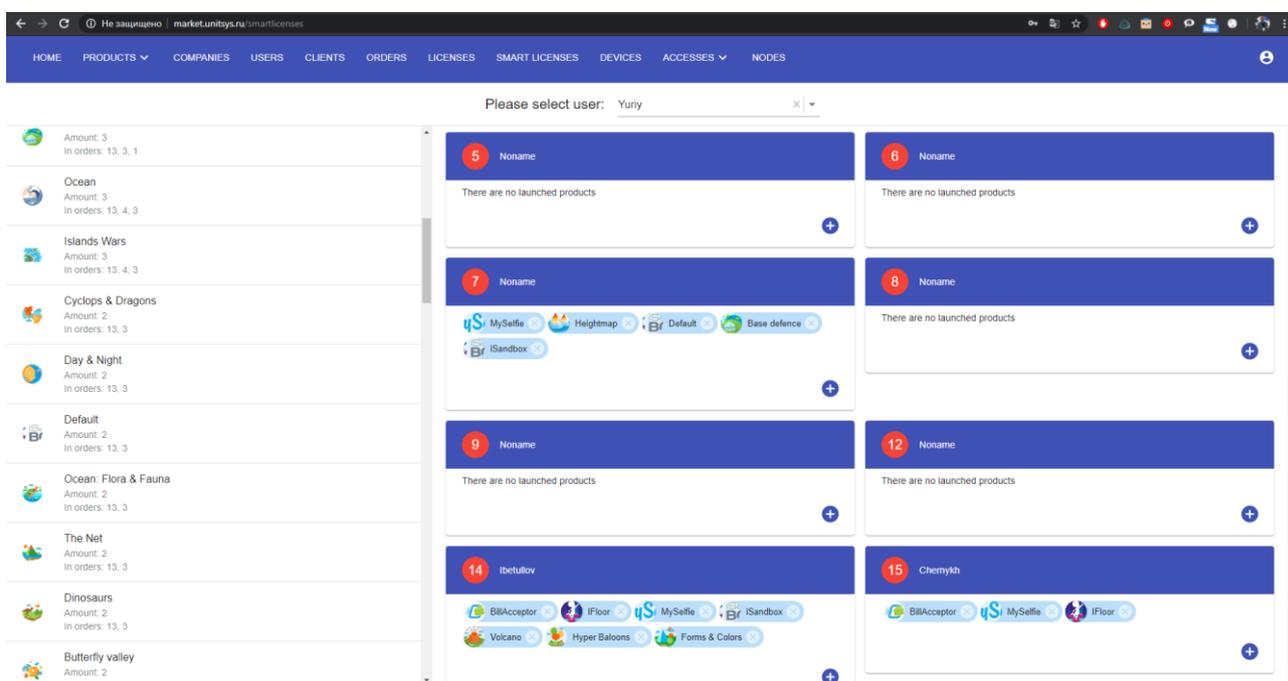
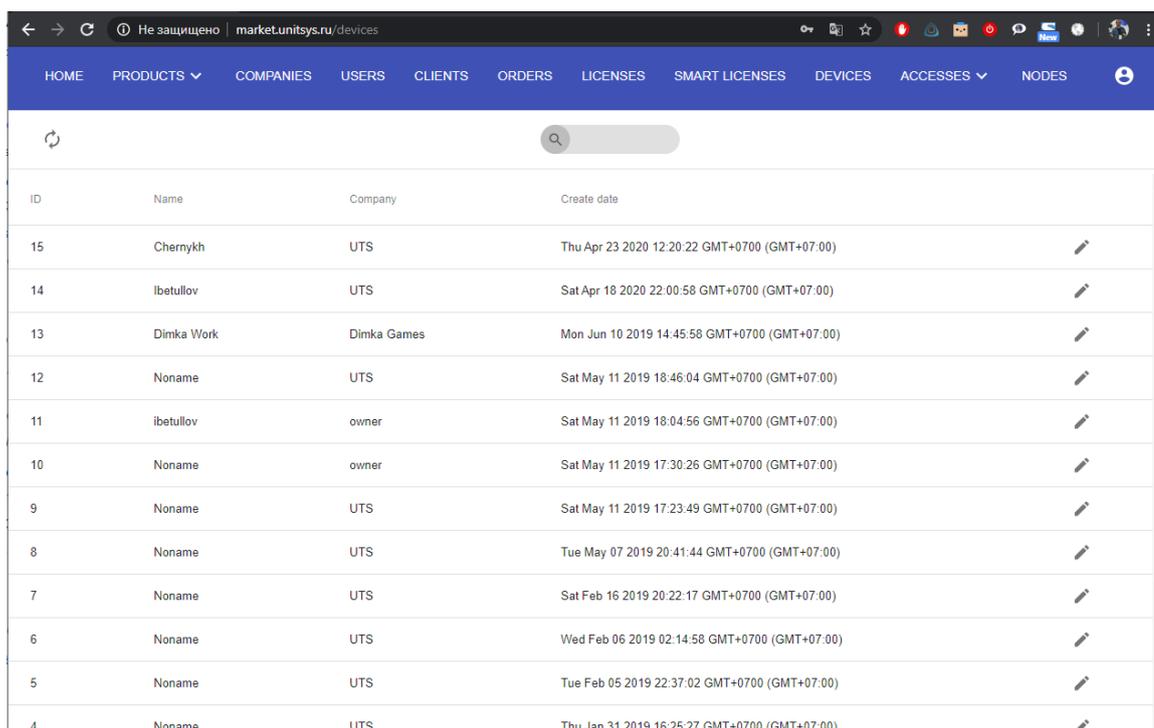


Рисунок 3.3 - Страница Smart Licenses

Страница Devices (рисунок 3.4) содержит список устройств, зарегистрированных в системе с указанием имени, компании владельца и даты добавления устройства в систему. Также имеется возможность изменить имя

устройства. Для поиска определённого устройства реализован поиск устройств имени.



The screenshot shows a web browser window with the URL 'marketunitsys.ru/devices'. The page has a blue navigation bar with the following menu items: HOME, PRODUCTS, COMPANIES, USERS, CLIENTS, ORDERS, LICENSES, SMART LICENSES, DEVICES, ACCESSSES, and NODES. Below the navigation bar is a search bar with a magnifying glass icon. The main content area displays a table with the following columns: ID, Name, Company, and Create date. The table contains 12 rows of data, each with a unique ID and a corresponding create date. The 'Name' and 'Company' columns are empty for most rows, while the 'Create date' column shows various timestamps. Each row has a small edit icon (pencil) to its right.

ID	Name	Company	Create date
15	Chernykh	UTS	Thu Apr 23 2020 12:20:22 GMT+07:00 (GMT+07:00)
14	Ibetullov	UTS	Sat Apr 18 2020 22:00:58 GMT+07:00 (GMT+07:00)
13	Dimka Work	Dimka Games	Mon Jun 10 2019 14:45:58 GMT+07:00 (GMT+07:00)
12	Noname	UTS	Sat May 11 2019 18:46:04 GMT+07:00 (GMT+07:00)
11	ibetullov	owner	Sat May 11 2019 18:04:56 GMT+07:00 (GMT+07:00)
10	Noname	owner	Sat May 11 2019 17:30:26 GMT+07:00 (GMT+07:00)
9	Noname	UTS	Sat May 11 2019 17:23:49 GMT+07:00 (GMT+07:00)
8	Noname	UTS	Tue May 07 2019 20:41:44 GMT+07:00 (GMT+07:00)
7	Noname	UTS	Sat Feb 16 2019 20:22:17 GMT+07:00 (GMT+07:00)
6	Noname	UTS	Wed Feb 06 2019 02:14:58 GMT+07:00 (GMT+07:00)
5	Noname	UTS	Tue Feb 05 2019 22:37:02 GMT+07:00 (GMT+07:00)
4	Noname	UTS	Thu Jan 31 2019 16:25:27 GMT+07:00 (GMT+07:00)

Рисунок 3.4 - Страница Devices

На странице Licenses (рисунок 3.5) осуществляется управление лицензиями: создание и удаление. При создании лицензии (рисунок 3.6) указывается компания, номер заказа, номер устройства, к которому будет привязана лицензия, продукт, входящий в заказ и дата окончания действия лицензии. Удаление лицензии влечёт за собой удаление соответствующего ей продукта с устройства клиента.

ID	Company	Product	Order ID	Device ID	Create date	Expire date
112	UTS	Forms & Colors	13	14	Sun May 17 2020 17:26:21 GMT+0700 (GMT+07:00)	Fri May 17 2030 00:00:00 GMT+0700 (GMT+07:00)
111	UTS	Hyper Balloons	13	14	Sun May 17 2020 17:26:13 GMT+0700 (GMT+07:00)	Fri May 17 2030 00:00:00 GMT+0700 (GMT+07:00)
110	UTS	Volcano	13	14	Sun May 17 2020 17:26:10 GMT+0700 (GMT+07:00)	Fri May 17 2030 00:00:00 GMT+0700 (GMT+07:00)
109	UTS	iSandbox	13	14	Sun May 17 2020 17:26:06 GMT+0700 (GMT+07:00)	Fri May 17 2030 00:00:00 GMT+0700 (GMT+07:00)
108	UTS	BillAcceptor	12	14	Mon May 11 2020 18:04:28 GMT+0700 (GMT+07:00)	Sat May 11 2030 00:00:00 GMT+0700 (GMT+07:00)
107	UTS	BillAcceptor	11	15	Mon May 11 2020 17:59:45 GMT+0700 (GMT+07:00)	Sat May 11 2030 00:00:00 GMT+0700 (GMT+07:00)
106	UTS	MySelfie	10	15	Thu Apr 23 2020 12:23:43 GMT+0700 (GMT+07:00)	Tue Apr 23 2030 00:00:00 GMT+0700 (GMT+07:00)
105	UTS	IFloor	10	15	Thu Apr 23 2020 12:23:36 GMT+0700 (GMT+07:00)	Tue Apr 23 2030 00:00:00 GMT+0700 (GMT+07:00)
104	UTS	IFloor	9	14	Thu Apr 23 2020 11:44:08 GMT+0700 (GMT+07:00)	Tue Apr 23 2030 00:00:00 GMT+0700 (GMT+07:00)
103	UTS	MySelfie	9	14	Thu Apr 23 2020 11:44:04 GMT+0700 (GMT+07:00)	Tue Apr 23 2030 00:00:00 GMT+0700 (GMT+07:00)
102	UTS	MySelfie	8	7	Sun Apr 19 2020 21:59:26 GMT+0700 (GMT+07:00)	Fri Apr 19 2030 05:00:00 GMT+0700 (GMT+07:00)
101	UTS	Heightmap	4	7	Sun Apr 19 2020 20:21:40 GMT+0700 (GMT+07:00)	Fri Apr 19 2030 05:00:00 GMT+0700 (GMT+07:00)

Рисунок 3.5 - Страница Licenses

**Create license**

Company:  
None

Order:  
None

Device:  
None

Products:  
Please select an order first.

Expire date:  
16.05.2021

SUBMIT    CLOSE

Рисунок 3.6 – Диалог создания лицензии

### 3.3.2 Утилита UTS Packer

UTS Packer это консольная утилита, разработанная на Node.js, которая предназначена для использования разработчиками и упаковки приложений в пакеты перед их публикацией. упаковка приложения в пакет необходима для его корректной идентификацией приложения UTS Launcher. Для этого каждое приложение снабжается описанием. Пакет представляет собой сжатый архив формата *zip*.

В качестве описания пакета выступает текстовый файл в формате JSON в котором указываются идентификатор приложения в системе UTS Marketplace, его версия и путь для исполняемого файла. Для стандартизации и удобства использования файл описания имеет расширение: *softmeta*. Пример файла описания для приложения IFloor приведён на рисунке 3.7.



```
IFloor.softmeta – Блокнот
Файл  Правка  Формат  Вид  Справка
{
  "alias": "ifloor",
  "version": {
    "title": "1.0.5",
    "number": 1035
  },
  "persistentFiles": {
    "all": [],
    "win": [],
    "linux": []
  },
  "run": {
    "all": [],
    "win": [
      {
        "main": true,
        "subPath": "",
        "launchFile": "IFloor\\Binaries\\Win64\\IFloor.exe",
        "args": []
      }
    ],
    "linux": []
  }
}
```

Рисунок 3.7 - Пример файла описания

Использования утилиты предполагает указание обязательных параметров запуска, а именно:

- путь до приложения, требующее упаковки. Необходимо чтобы в указанном расположении присутствовал файл описания пакета, благодаря которому утилита поймёт какие файлы необходимо упаковать в пакет;
- ключ для публикации версии приложения. Ключ публикации, которым будет подписан пакет. Он выдаётся администратором системы и позволяет гарантировать, что пакет является подлинным.

### **3.3.3 Клиентское приложение UTS Launcher**

Приложение UTS Launcher представляет собой кроссплатформенное приложение, имеющее два режима работы: режим оболочки и режим загрузчика.

В режиме оболочки приложение осуществляет контроль за запущенными программами и блокирует доступ пользователя к системе. Из режима оболочки можно переключиться в режим загрузчика введя пароль администратора.

В режиме загрузчика приложение позволяет управлять доступным программным обеспечением: загружать, устанавливать, обновлять, настраивать способ запуска и режим работы приложения. Из режима загрузчика можно перейти в режим оболочки, при этом будут автоматически запущены все указанные в настройке автозапуска программы.

При первом запуске приложения запускается в режиме загрузчика и открывается страница авторизации пользователя (рисунок 3.8). После авторизации пользователю доступны следующие страницы:

- Мои программы – страница с доступными продуктами;
- Обновления – страница со списком обновлений;
- Лицензии – страница со списком доступных лицензий;
- Программы – страница со списком установленных программ;
- Настройки – страница с настройками;
- Журнал ошибок – страница со списком ошибок;
- Информация о пользователе – страница с информацией о пользователе.

При первой авторизации пользователя происходит регистрация его устройства в системе – для этого собираются идентификаторы оборудования для его точной идентификации. Эта информация отправляется в сервис s-device. Эта информация используется при привязке купленной лицензии к устройству. Каждая лицензия может быть привязана только к одному устройству и в случае несовпадения идентификационных данных устройства с данными привязанными к лицензии продукт не будет установлен на устройство.

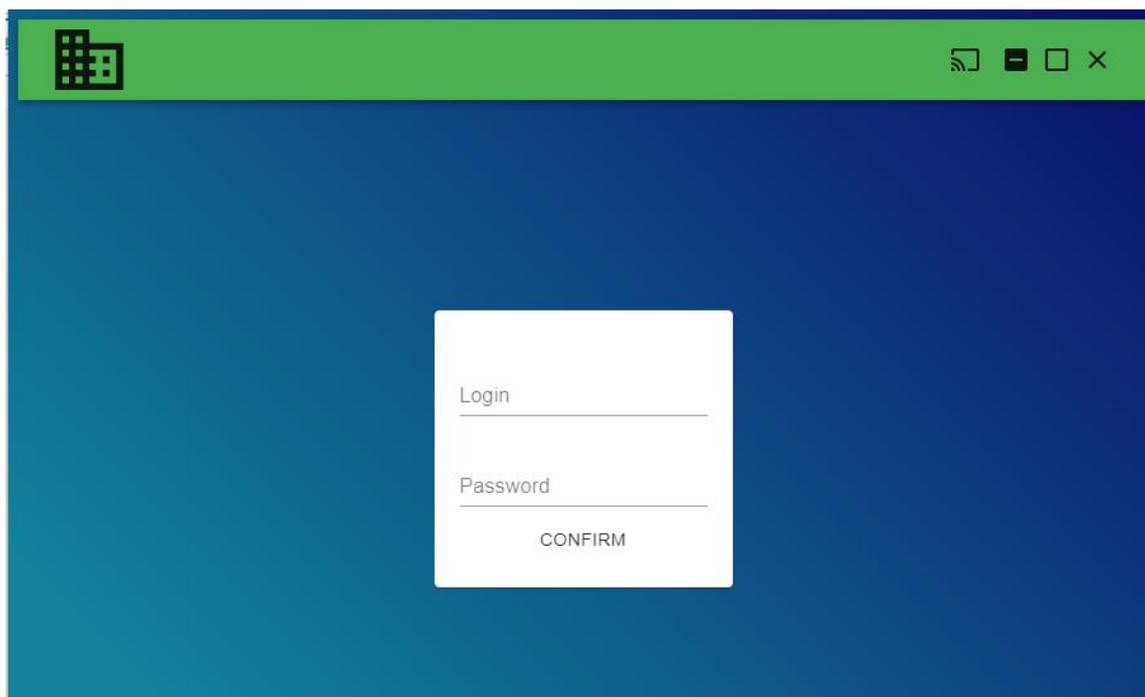


Рисунок 3.8 - Окно авторизации

В режиме оболочки приложение разворачивается во весь экран, блокируя пользователю доступ к системным элементам управления. Выход из этого режима возможен только после ввода пароля во избежание нарушения работы устройства (Рисунок 3.9).

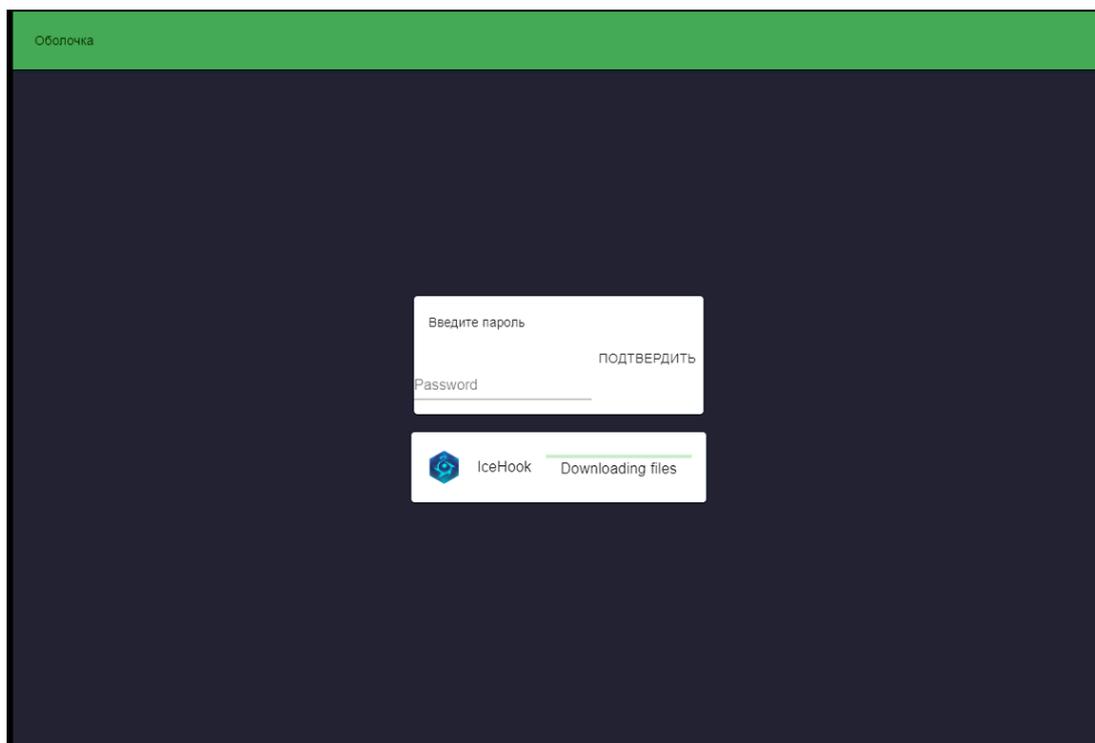


Рисунок 3.9 - Окно разблокировки режима оболочки

На странице Мои программы пользователю представляется список продуктов, доступных для установки в левой части экрана (Рисунок 3.10). Там же показан статус каждого их продуктов.

При выборе определённого продукта показывается информация о нём и списке подпрограмм (DLC) доступных для выбранного продукта с указанием статуса их установки. Также на этой странице можно установить, удалить, обновить и запустить выбранный продукт (рисунок 3.11).

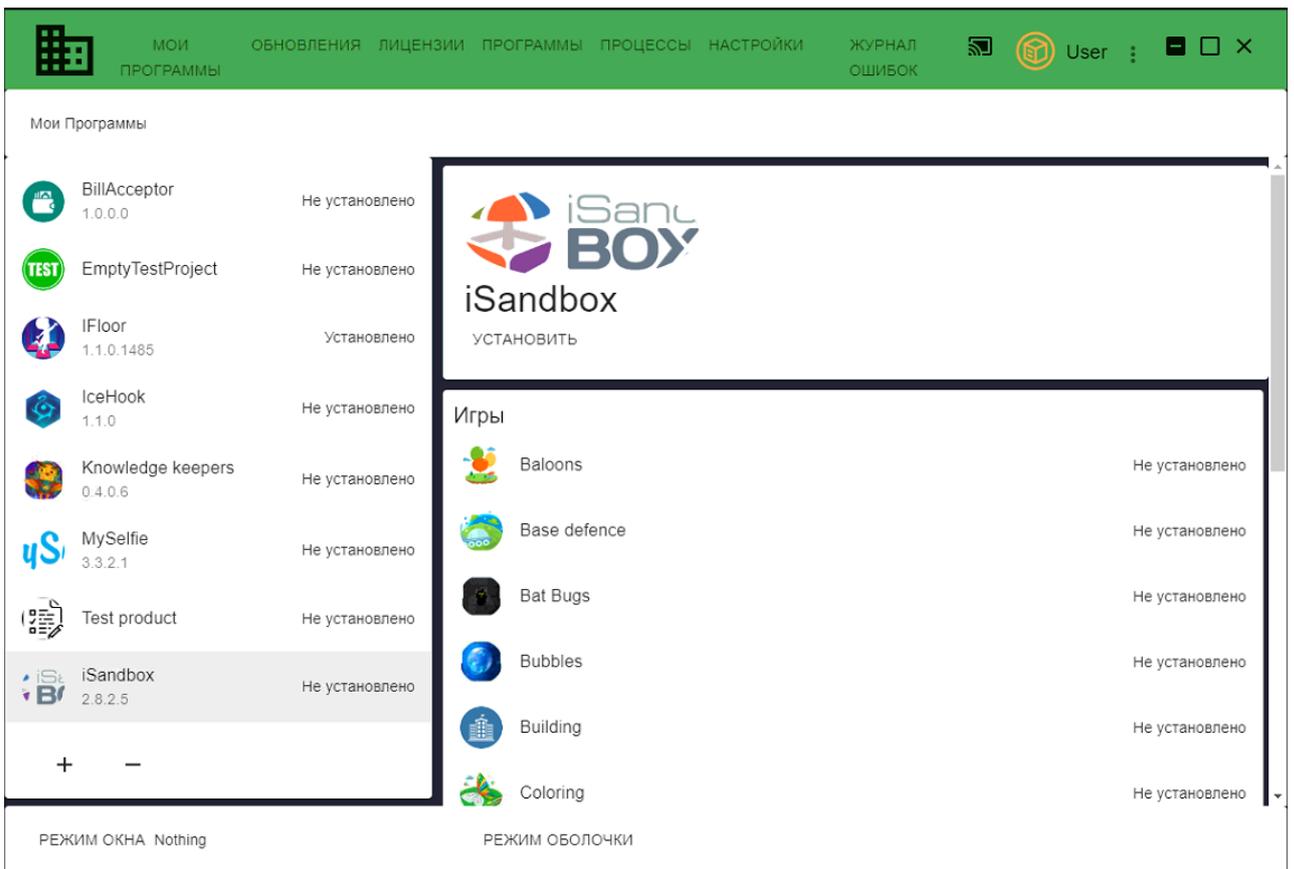


Рисунок 3.10 - Страница "Мои программы"

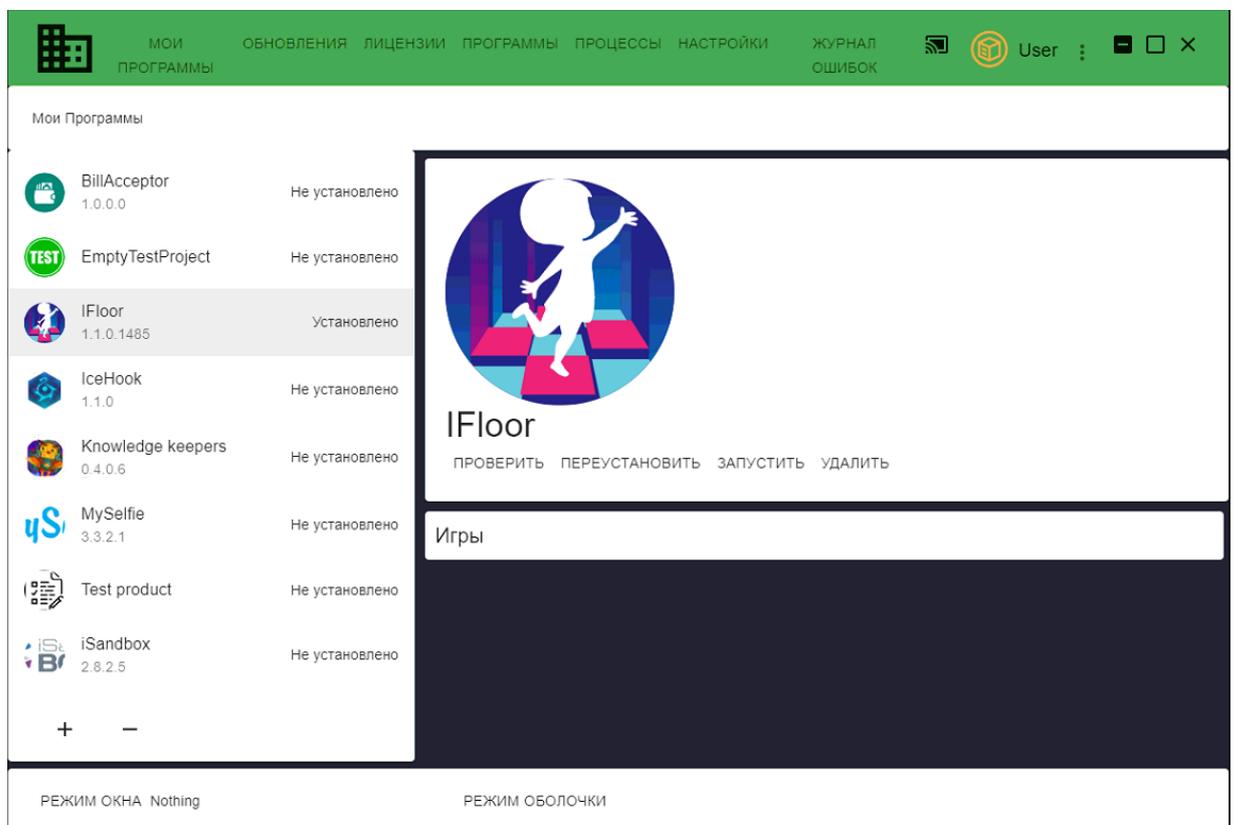


Рисунок 3.11 – Страница установленного приложения

На странице Обновления пользователю представляется список доступных к обновлению программ. На рисунке 3.12 видно, что доступно к обновлению приложение IFloor.

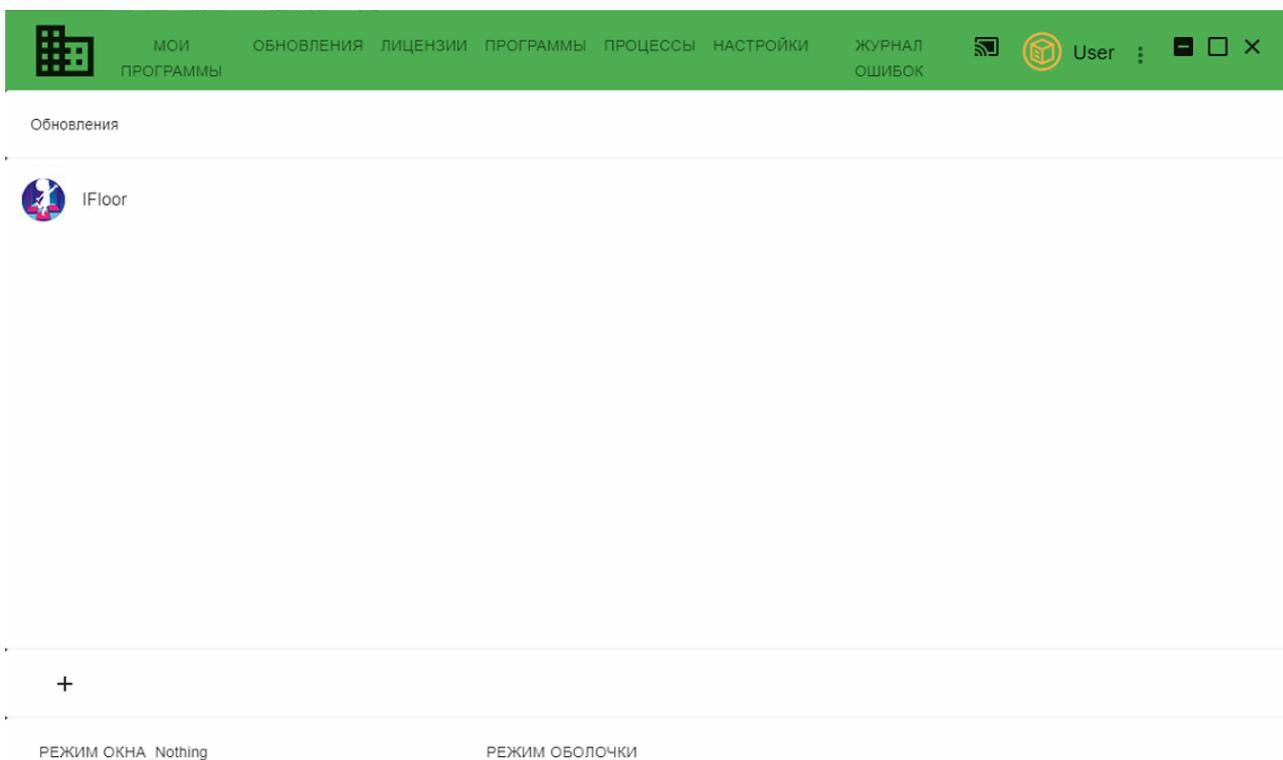


Рисунок 3.12 - Страница обновления

Во время получения обновлений в режиме оболочки отображается процесс загрузки файлов и установки программы. При этом обновление запущенных программ производится во время их непосредственной работы, не прерывая их работы. В случае успешной загрузки и установки новой версии работающее приложение будет закрыто и перезапущено с новой установленной версией. Старая версия будет удалена после успешного запуска. На рисунке 3.13 изображён процесс обновления приложения IceHook.

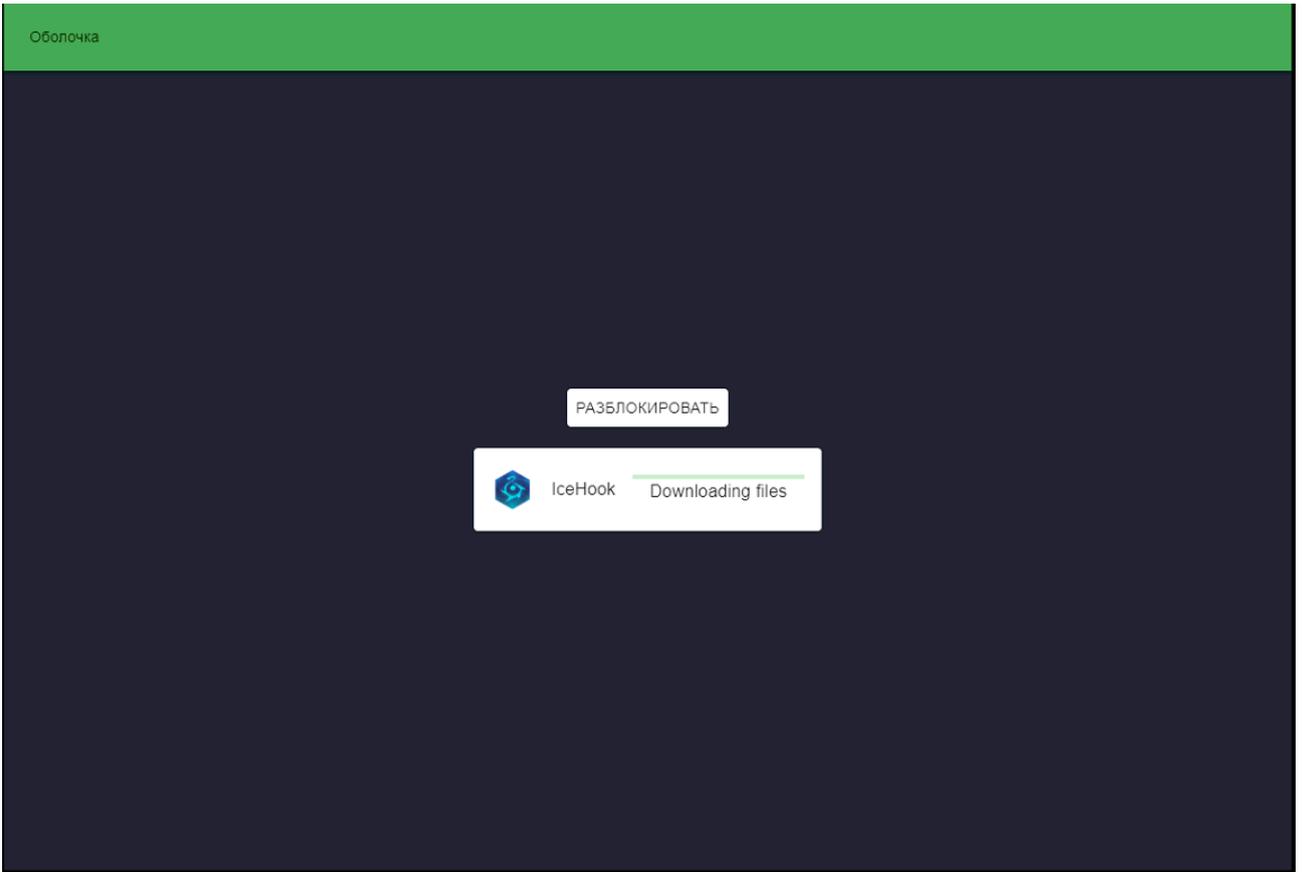


Рисунок 3.13 - Отображение загрузки обновление в режиме оболочки

На странице лицензий пользователю представляется информация об полученных им лицензиях на устройство, на котором запущено приложение (рисунок 3.14).

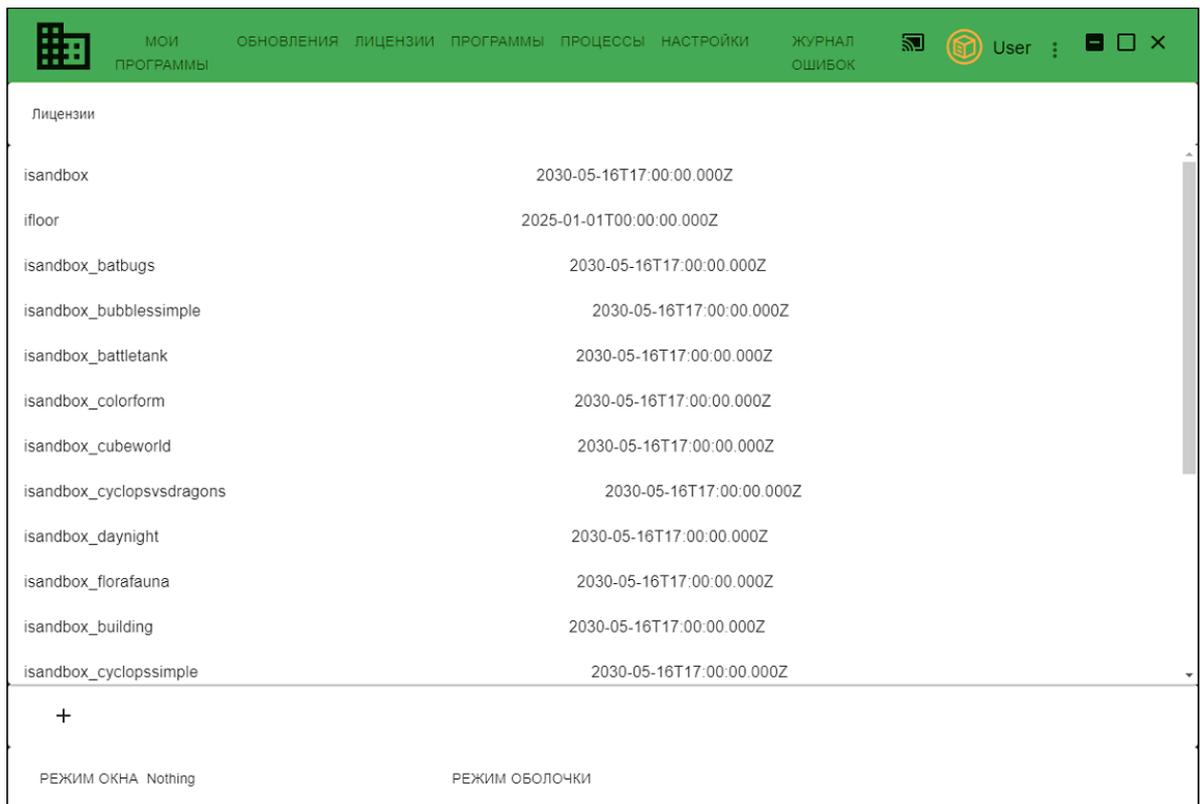


Рисунок 3.14 - Страница лицензии

Страница настроек позволяет пользователю настроить такие параметры, как:

- Язык приложения
- Пути установки программы
- Пути хранения временных файлов
- Пути хранения файлов восстановления
- Пароль разблокировки режима оболочки
- Настройки режима запуска приложения (в режиме оболочки или загрузчика).
- Адреса серверов
- Параметры автоматического обновления приложения и другие параметры.

На рисунке 3.15 показана страница настроек приложения.

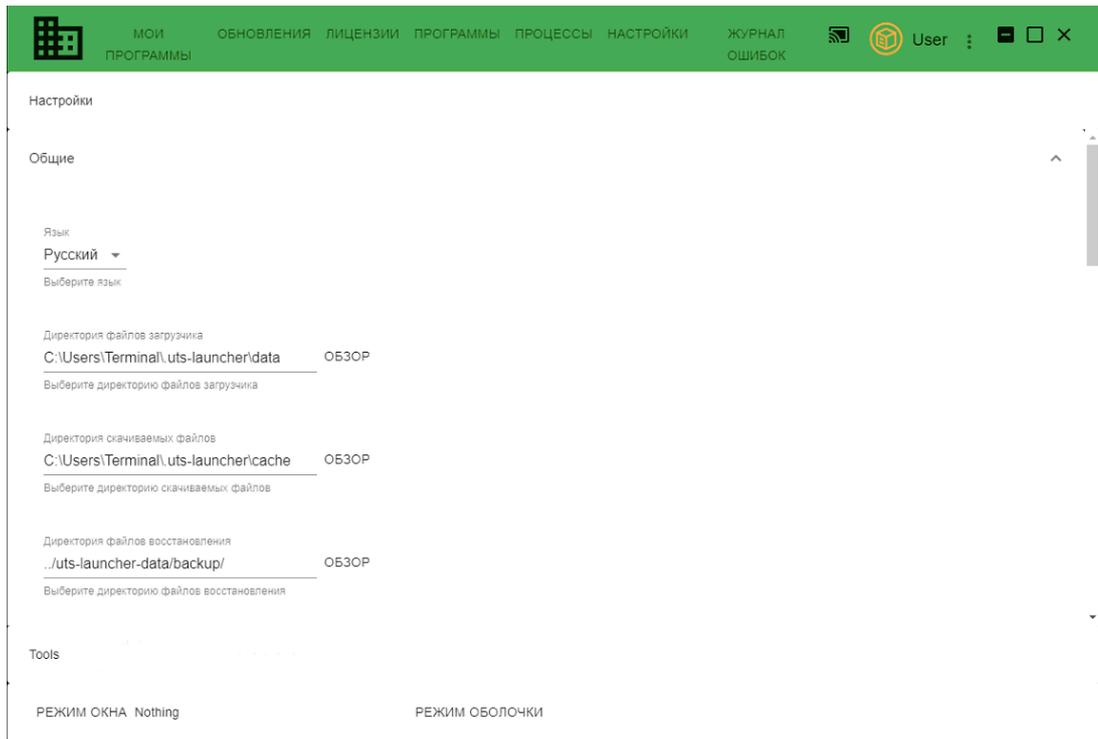


Рисунок 3.15 - Страница настроек

Страница Журнал ошибок предназначена для диагностики работы приложения. Здесь выводится список возникших ошибок при работе лаунчера (рисунок 3.16).

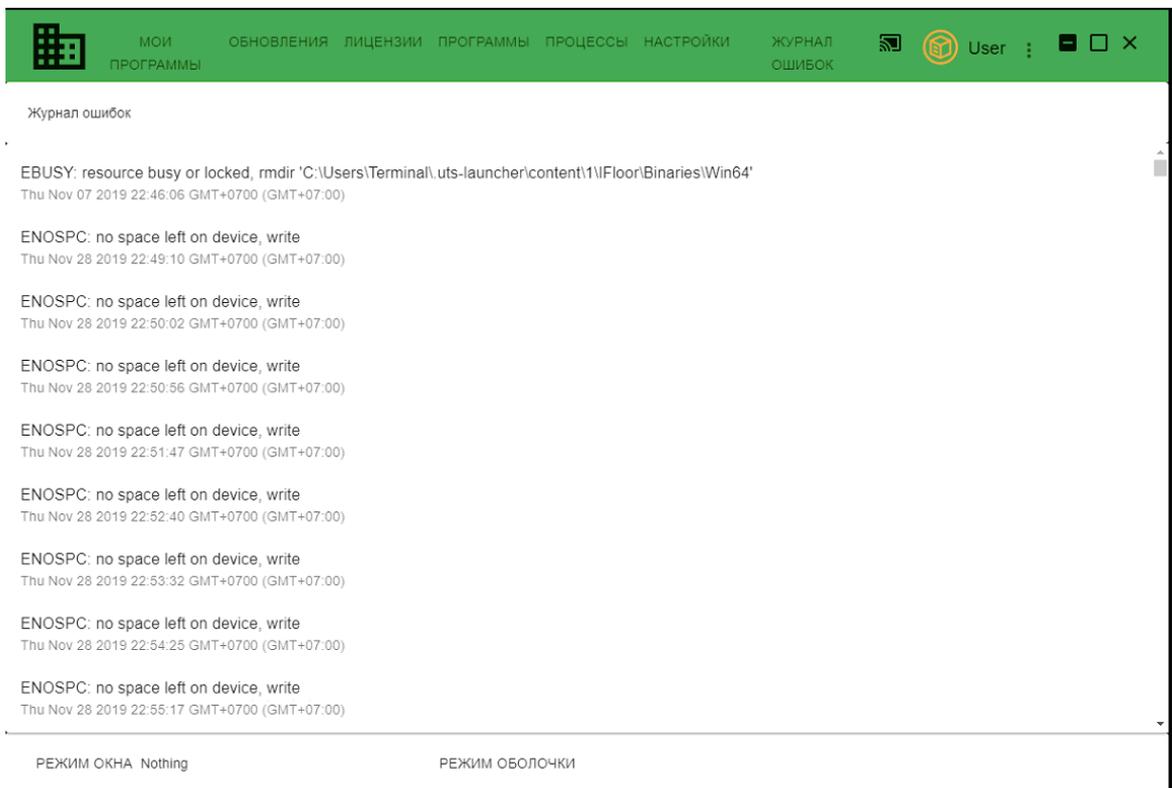


Рисунок 3.16 - Страница журнала ошибок

Страница программы представляет пользователю список всех установленных программ. На этой странице можно выбрать программы, которые будут автоматически запускаться в режиме оболочки и состояние которых будет отслеживаться лаунчером – в случае возникновения ошибки приложение будет перезапущено. Здесь же отображается статус работы приложения, можно настроить путь до исполняемого файла и запустить или остановить запущенное приложение (рисунок 3.17).

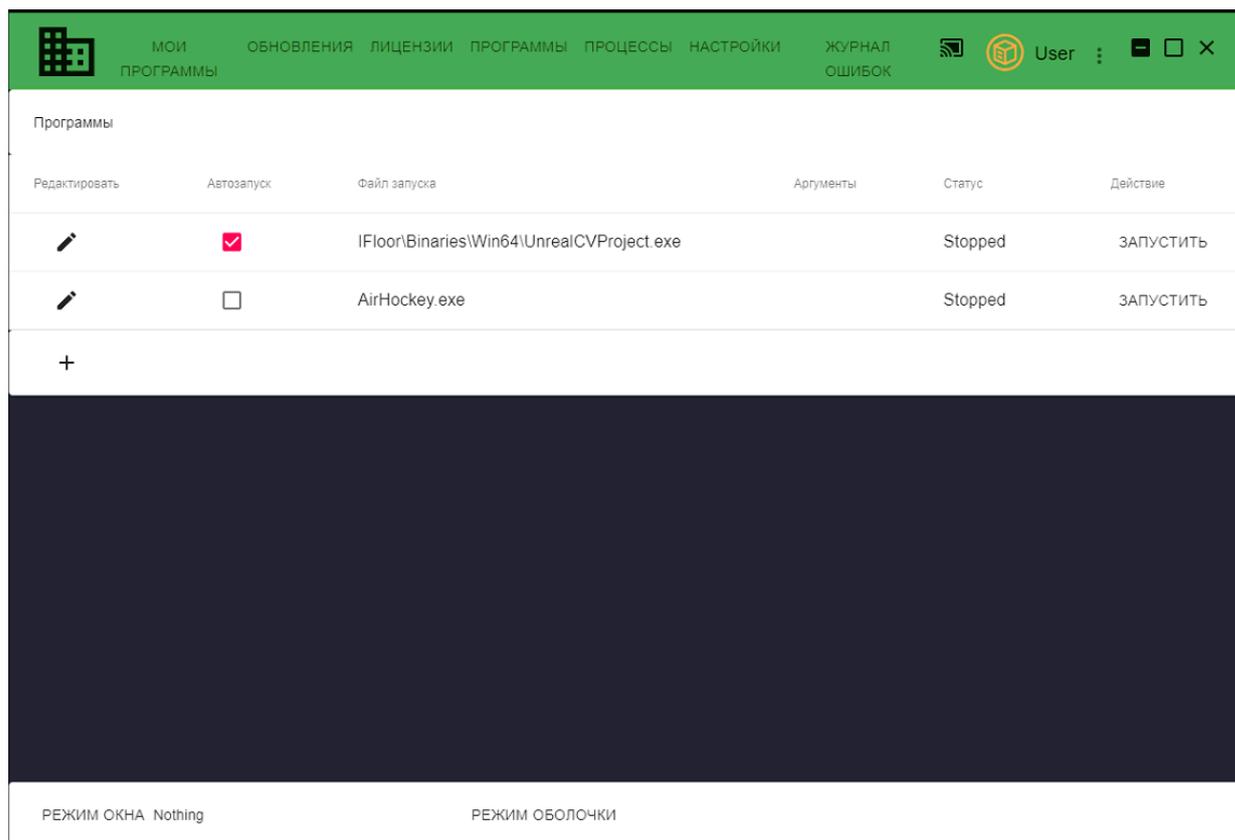


Рисунок 3.17 - Страница с установленными программами

Просмотреть запущенные процессы и их статус можно на странице Процессы. В списке процессов отображается идентификатор (PID) процесса и путь до исполняемого файла. Здесь же можно остановить запущенный процесс (рисунок 3.18).

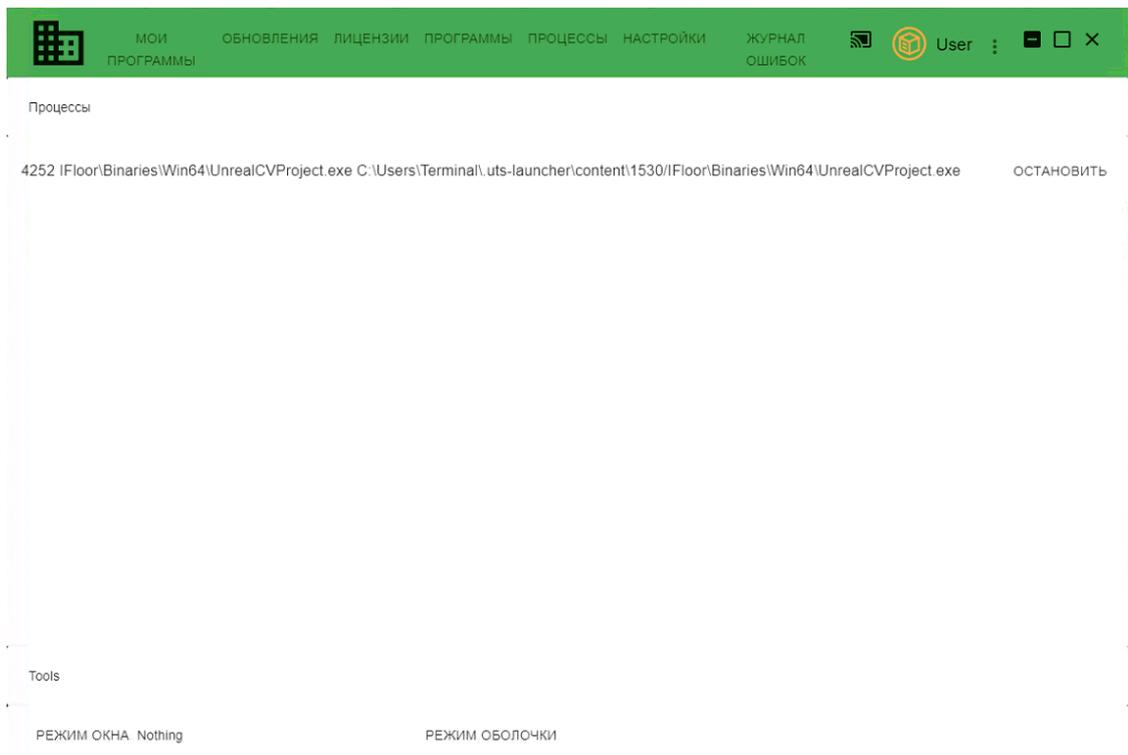


Рисунок 3.18 - Запущенные процессы

На странице пользователя отображается информация об авторизованном пользователе и кол-ве установленных им продуктов (рисунок 3.19).

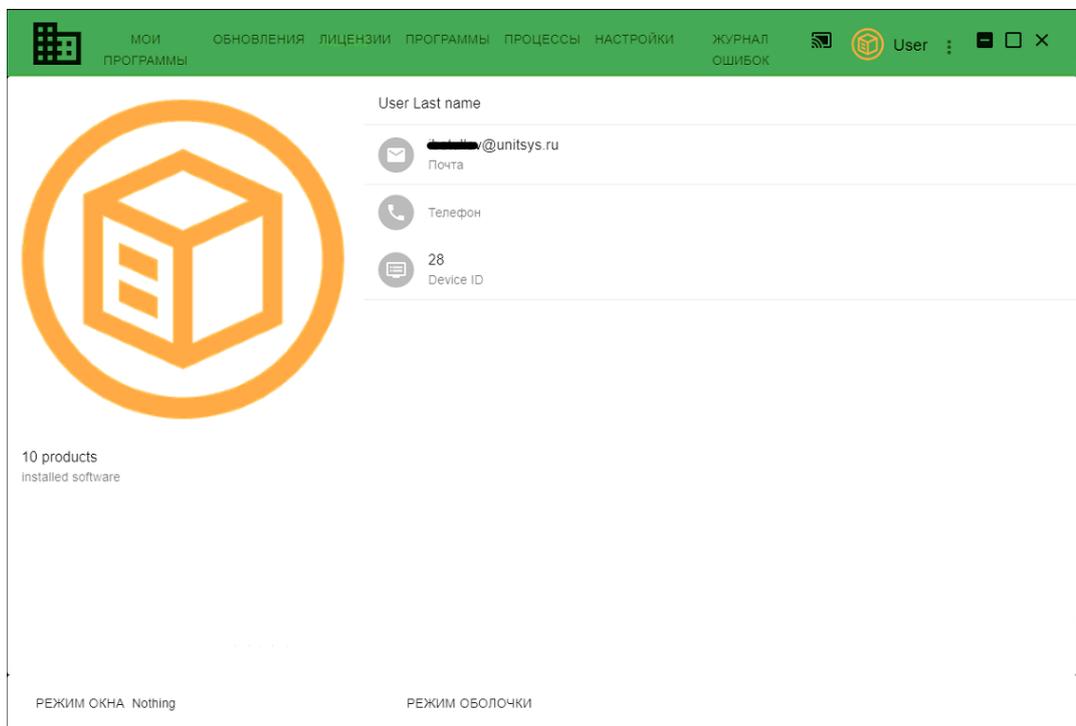


Рисунок 3.19 - Страница профиля пользователя

## 4 Финансовый менеджмент

Создание любого продукта подразумевает определенные затраты ресурсов как материальных, так и трудовых. Расчет ресурсозатратности продукта учитывается при оценке его коммерческого потенциала, что, в свою очередь, влияет на конкурентоспособность продукта на рынке. Оценка коммерческого потенциала также необходима при поиске источников финансирования, так как позволяет оценить рентабельность продукта.

Цель раздела – комплексное описание и анализ финансово-экономических аспектов выполненной работы. Оцениваются полные денежные затраты на исследование (проект), а также дается хотя бы приближенная экономическая оценка результатов ее внедрения. Это в свою очередь позволит с помощью традиционных показателей эффективности инвестиций оценить экономическую целесообразность осуществления работы [25].

### 4.1 Организация и планирование работ

При организации процесса реализации конкретного проекта (ОКР) или научного исследования (НИ) необходимо рационально планировать занятость каждого из его участников и сроки проведения отдельных работ.

Для выполнения работ требуются:

- Инженер (И) – исполнитель НИР;
- Научный руководитель (НР).

Таблица 4.1 - Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
Определение тематики работы	НР, И	НР – 100% И – 30%
Анализ предметной области	НР, И	НР – 30% И – 100%
Составление и утверждение технического задания	НР, И	НР – 50% И – 100%
Разработка календарного плана	НР, И	НР – 100% И – 10%
Проектирование информационной системы	НР, И	НР – 30%

		И – 100%
Проектирование пользовательских интерфейсов	НР, И	НР – 30% И – 100%
Разработка информационной системы	И	И – 100%
Отладка информационной системы	И	И – 100%
Оформление пояснительной записки	НР, И	НР – 30% И – 100%

#### 4.1.1 Продолжительность этапов работ

Расчёт продолжительности этапов работ осуществлён опытно-статистическим методом, реализуемый экспертным способом, ввиду отсутствия актуальных аналогичных работ.

Для определения вероятных (ожидаемых) значений продолжительности работ  $t_{ож}$  используется формула:

$$t_{ож} = \frac{3*t_{min} + 2*t_{max}}{5}, \quad (4.1)$$

где  $t_{min}$  – минимальная продолжительность работы, дн;

$t_{max}$  – максимальная продолжительность работы, дн.

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести её в календарные дни. Расчёт продолжительности выполнения каждого этапа в рабочих днях ( $T_{РД}$ ) производится по формуле:

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} * K_{Д}, \quad (4.2)$$

где  $t_{ож}$  – продолжительность работы, дн.;

$K_{ВН}$  – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определённых длительностей.

$K_{Д}$  – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ.

В работе приняты следующие значения:  $K_{ВН} = 0,9$ ;  $K_{Д} = 1,2$ .

Расчёт продолжительности этапа в календарных днях ведётся по формуле:

$$T_{КД} = T_{РД} * T_{К}, \quad (4.3)$$

где  $T_{\text{КД}}$  – продолжительность выполнения этапа в календарных днях;

$T_{\text{К}}$  – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях, и рассчитываемый по формуле:

$$T_{\text{К}} = \frac{T_{\text{КАЛ}}}{T_{\text{КАЛ}} - T_{\text{ВД}} - T_{\text{ПД}}}, \quad (4.4)$$

где  $T_{\text{КАЛ}}$  – календарные дни ( $T_{\text{КАЛ}} = 366$ );

$T_{\text{ВД}}$  – выходные дни;

$T_{\text{ПД}}$  – праздничные дни.

Количество праздничных и выходных дней для шестидневной рабочей недели, с учетом нерабочих дней, установленных Указами Президента РФ от 25.03.2020 № 206, от 02.04.2020 N 239, от 28.04.2020 № 294 составляет 97 дней [25], тогда:

$$T_{\text{К}} = \frac{366}{366 - 97} = 1,36.$$

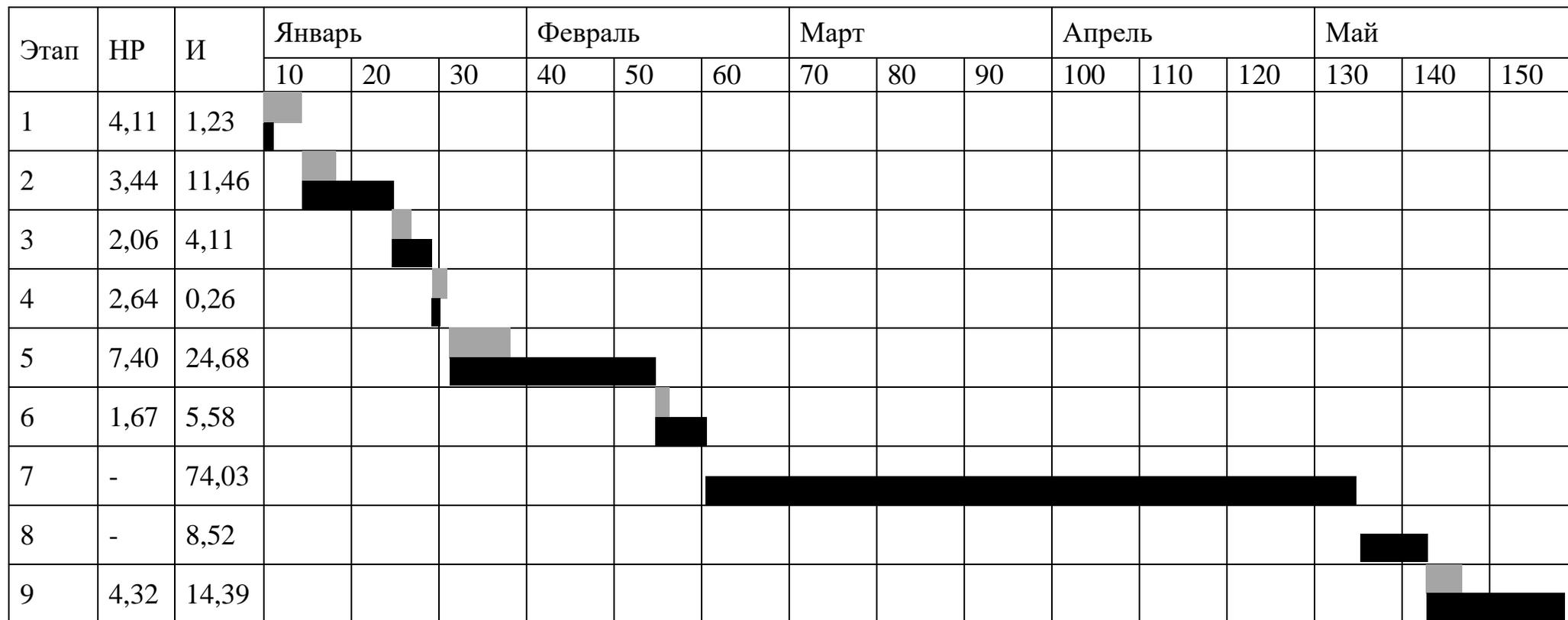
В рамках планирования научного проекта определены трудозатраты на выполнение проекта (таблица 4.1) и построен линейный график проекта (таблица 4.2).

В столбцах (3–5) таблицы 4.1 реализован расчёт продолжительности работ экспертным способом по формуле 4.1. В столбцах 6 и 7 таблицы представлены величины трудоёмкости этапа для каждого из участников проекта. В столбцах 8 и 9 представлены трудоёмкости для участников проекта. Итог по столбцу 5 даёт общую ожидаемую продолжительность работы над проектом в рабочих днях, итоги по столбцам 8 и 9 – общие трудоёмкости для каждого из участников проекта. Две последних величины далее будут использованы для определения затрат на оплату труда участников и прочие затраты. Величины трудоёмкости этапов по исполнителям  $T_{\text{КД}}$  (данные столбцов 8 и 9 кроме итогов) позволяют построить линейный график выполнения проекта, приведённый в таблице 4.2.

Таблица 4.2 - Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.- дн.			
		t <sub>min</sub>	t <sub>max</sub>	t <sub>ож</sub>	Т <sub>рд</sub>		Т <sub>кд</sub>	
					НР	И	НР	И
1	2	3	4	5	6	7	8	9
Определение тематики работы	НР, И	2,00	4,00	2,8	3,02	0,91	4,11	1,23
Анализ предметной области	НР, И	5,00	12,00	7,8	2,53	8,42	3,44	11,46
Составление и утверждение технического задания	НР, И	2,00	4,00	2,8	1,51	3,02	2,06	4,11
Разработка календарного плана	НР, И	1,00	3,00	1,8	1,94	0,19	2,64	0,26
Проектирование информационной системы	НР, И	14,00	21,00	16,8	5,44	18,14	7,40	24,68
Проектирование пользовательских интерфейсов	НР, И	3,00	5,00	3,8	1,23	4,10	1,67	5,58
Разработка информационной системы	И	48,00	54,00	50,4	-	54,43	-	74,03
Отладка информационной системы	И	5,00	7,00	5,8	-	6,26	-	8,52
Оформление пояснительной записки	НР, И	7,00	14,00	9,8	3,18	10,58	4,32	14,39
Итого:				101,8	18,86	106,08	25,65	144,27

Таблица 4.3 - Линейный график работ



НР - ■ И - ■

## 4.2 Расчет сметы затрат на выполнение проекта

В состав затрат на создание проекта включается величина всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Расчет сметной стоимости ее выполнения производится по следующим статьям затрат:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- прочие (накладные расходы) расходы.

### 4.2.1 Расчет затрат на материалы

Расчет стоимости материальных затрат производится по действующим прейскурантам или договорным ценам. В стоимость материальных затрат включают транспортно-заготовительные расходы (3 – 5 % от цены). В эту же статью включаются затраты на оформление документации (канцелярские принадлежности, тиражирование материалов) [25]. Результаты по данной статье занесены в таблицу 4.4.

Таблица 4.4 - Сырье, материалы, комплектующие изделия и покупные полуфабрикаты

Наименование	Марка, размер	Кол-во	Цена за единицу, руб	Сумма, руб
Ручка	MaxRiter	4	50	200
Бумага	Снегурочка	2 упаковки	150	300
Маркер для доски	Index	4	100	400
Всего за материалы				900
Транспортно-заготовительные расходы (5%)				45
Итого по статье $C_{\text{мат}}$				945

## 4.2.2 Расчет заработной платы

В настоящую статью включается основная заработная плата научных и инженерно-технических работников, рабочих макетных мастерских и опытных производств, непосредственно участвующих в выполнении работ по данной теме. Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада.

Среднедневная тарифная заработная плата ( $ЗП_{\text{дн-т}}$ ) рассчитывается по формуле, учитывающей что в году 269 рабочих дней при 6-дневной рабочей неделе [26]:

$$ЗП_{\text{дн-т}} = MO/22,4, \quad (4.5)$$

где  $MO$  – месячный оклад сотрудника;

Величина месячных окладов для исполнителей проекта приняты: для инженера согласно окладу соответствующего специалиста, в ООО «Универсальный терминал системы», для научного руководителя согласно окладу сотрудников ТПУ.

Расчёты затрат на полную заработную плату приведены в таблице 4.5. Для учёта в её составе премий и районной надбавки используются следующие коэффициенты:

- Коэффициент учёта премий -  $K_{\text{ПР}} = 1,1$ ;
- Коэффициент учёта дополнительной заработной платы для 6-дневной недели -  $K_{\text{доп.ЗП}} = 1,188$ ;

- Коэффициент учёта районной надбавки -  $K_{\text{р}} = 1,3$ .

Тогда, интегральный коэффициент  $K_{\text{и}} = 1,1 \cdot 1,188 \cdot 1,3 = 1,699$ ;

Для перехода от тарифной (базовой) суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку (зарплатной части сметы) необходимо первую умножить на интегральный коэффициент  $K_{\text{и}}$  [25].

Таблица 4.5 - Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб. день	Затраты времени, раб. дни	К <sub>и</sub>	Фонд з/платы, руб.
НР	33 664,00	1 502,86	18,86	1,699	48 148,09
И	14 703,00	656,38	106,08	1,699	118 297,35
Итого:					166 445,44

### 4.2.3 Расчет затрат на социальный налог

Статья включает в себя отчисления во внебюджетные фонды, а именно единый социальный налог:

$$C_{\text{соц}} = k_{\text{соц}} * C_{\text{ЗП}}, \quad (4.6)$$

где  $k_{\text{соц}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Тогда сумма отчислений во внебюджетные фонды равна, руб.:

$$C_{\text{соц}} = 0,3 * 166\,445,44 = \mathbf{49\,933,33 \text{ руб.}}$$

### 4.2.4 Расчет затрат на электроэнергию

В данном пункте рассчитываются затраты на электричество за время работы над проектом. Вычисление производится по формуле:

$$C_{\text{эл.об}} = P_{\text{об}} * t_{\text{об}} * Ц_{\text{э}}, \quad (4.7)$$

где  $P_{\text{об}}$  - мощность, потребляемая оборудованием, кВт, равно приблизительно 120 Вт;

$t_{\text{об}}$  – время работы оборудования, час;

$Ц_{\text{э}}$  - тариф на 1 кВт\*час, для ТПУ  $Ц_{\text{э}} = 6,59 \text{ руб./кВт}\cdot\text{час}$  (с НДС).

Время работы оборудования рассчитывается на основе итоговых данных из таблицы 4.2 для инженера ( $T_{\text{рд}}$ ) из расчета, что продолжительность рабочего дня равна 8 часов и составляет 848,62 часа.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{\text{об}} = P_{\text{ном}} * K_{\text{с}}, \quad (4.8)$$

где  $P_{\text{ном}}$  – номинальная мощность оборудования, кВт;

$K_C \leq 1$  – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности  $K_C = 1$ .

Расчета затрат на электроэнергию для технологических целей приведен в таблице 4.6.

Таблица 4.6 - Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{\text{ОБ}}$ , час	Потребляемая мощность $P_{\text{ОБ}}$ , кВт	Затраты $\text{Э}_{\text{ОБ}}$ , руб.
Персональный компьютер	848,62*0,6	0,7	2348,81
Итого:			<b>2348,81</b>

#### 4.2.5 Расчет амортизационных расходов

В данном пункте рассчитывается амортизация используемого, только в период работы над выполнением проекта, оборудования. Для выполнения вычислений используется формула:

$$C_{\text{АМ}} = \frac{N_{\text{А}} * C_{\text{ОБ}} * t_{\text{рф}} * n}{F_{\text{д}}}, \quad (4.9)$$

где  $N_{\text{А}}$  – годовая норма амортизации единицы оборудования;

$C_{\text{ОБ}}$  – балансовая стоимость единицы оборудования с учётом транспортно-закупочных расходов;

$F_{\text{д}}$  – действительный годовой фонд времени работы соответствующего оборудования;

$t_{\text{рф}}$  – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта;

$n$  – число задействованных однотипных единиц оборудования.

$$C_{\text{АМ}} = \frac{0,4 * 57500 * 848,62 * 1}{269 * 8} = 9\,069,82 \text{ руб.}$$

#### 4.2.6 Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на выполнение проекта, которые не учтены в предыдущих статьях, их следует принять равными 10% от суммы всех предыдущих расходов, т.е. [25]:

$$C_{\text{проч}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}}) * 0,1, \quad (4.10)$$

Тогда, для настоящей работы:

$$\begin{aligned} C_{\text{проч}} &= (945 + 226382,07 + 67914,62 + 1761,65 + 9\,069,82) * 0,1 \\ &= \mathbf{30\,380,59} \text{ руб.} \end{aligned}$$

#### 4.2.7 Расчет общей себестоимости разработки

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	945
Основная заработная плата	$C_{\text{зп}}$	166445,44
Отчисления в социальные фонды	$C_{\text{соц}}$	49933,63
Расходы на электроэнергию	$C_{\text{эл.}}$	2348,81
Амортизационные отчисления	$C_{\text{ам}}$	9062,82
Прочие расходы	$C_{\text{проч}}$	30380,59
Итого:		259 116,29

Таким образом, суммарные на разработку составили  $C = \mathbf{259\,116,29}$  руб.

#### 4.2.8 Расчет прибыли

Прибыль от реализации проекта в зависимости от конкретной ситуации (масштаб и характер получаемого результата, степень его определённости и коммерциализации, специфика целевого сегмента рынка и т.д.) может определяться различными способами. Из-за недостатка данных прибыль следует принять в размере от 5 до 20 % от полной себестоимости проекта [25].

В данном случае прибыль принята равной  $\mathbf{51\,823,26}$  руб., что составляет 20% от расходов на разработку проекта.

#### 4.2.9 Расчет НДС

НДС составляет 20% от суммы затрат на разработку и прибыли [25].

$$\text{НДС} = (259116,29 + 51823,26) * 0,2 = \mathbf{62\ 187,91 \text{ руб.}}$$

#### 4.2.10 Цена разработки ОКР (НИР)

Цена равна сумме полной себестоимости, прибыли и НДС.

$$C_{\text{НИР(КР)}} = 259116,29 + 51823,26 + 62187,91 = \mathbf{373\ 127,46 \text{ руб.}}$$

### 4.3 Оценка экономической эффективности проекта

Разработка информационной системы велась по заказу компании ООО «Универсальные терминал системы».

Количественная оценка эффективности проекта не представляется возможной из-за недостатка данных.

Производственный эффект заключается в упрощении и автоматизации бизнес-процессов компании, упрощении взаимодействия отделов между собой.

Для компании в целом:

- Сокращение расходов на техническую поддержку клиентов компании;
- Увеличение эффективности работы технической поддержки;
- Высокая сохранность данных.

Для клиентов компании:

- Экономия времени и «нервов» при обновлении программного обеспечения, что обеспечивает большую лояльность к компании.

Для службы технической поддержки:

- Сокращение времени на выяснение информации об установленном программном обеспечении у клиента;
- Оперативное исправление ошибок в программном обеспечении без привлечения к процессу клиента.

Для отдела клиентского сервиса:

- Ускорение процесса первичной настройки оборудования;

- Автоматизация установки программного обеспечения на устройство клиента;

- Решение проблемы с необходимостью выяснения актуальной версии программного обеспечения;

Для разработчиков:

- Готовое решение для непрерывной доставки (*CD*) и его интеграция с системами непрерывной интеграции (*CI*);

- Автоматическое обновление продуктов на тестовых стендах;

- Автоматическое обновление продуктов клиентов до новой версии, без задействования отдела технической поддержки;

- Своевременное исправление ошибок в программном обеспечении у всей базы клиентов;

- Защита продуктов при их распространении.

В данной работе экономический эффект представлен в форме экономии рабочего времени и носит внутрисистемный характер, которому можно дать условную оценку: сокращение затрат на заработную плату, сокращение издержек на производство.

## 5 Социальная ответственность

Понятие «Социальная ответственность» сформулировано в международном стандарте ICCSR26000:2011 «Социальная ответственность организации». В нём рассматриваются вопросы соблюдения прав персонала на труд, выполнения требований к безопасности и гигиене труда, к промышленной безопасности, охране окружающей среды и ресурсосбережению [27].

В соответствии со стандартом целями составления настоящего раздела является принятие проектных решений, исключая несчастные случаи в производстве, и снижение вредных воздействий на окружающую среду.

### 5.1 Введение

Объектом исследования является автоматизация процессов установки, обновления, запуска и контроля выполнения программного обеспечения, а именно:

- изучение и анализ предметной области, обзор возможных аналогов системы;
- выявление и документирование требований к системе;
- проектирование компонентов системы;
- проектирование пользовательского интерфейса;
- изучение и выбор программных средств и технологий разработки;
- реализация намеченных функциональных компонентов системы,
- разработка пользовательского интерфейса системы;
- интеграция разработанного функционала в систему цифрового распространения программного обеспечения UTS Marketplace.

Разработанная информационная система позволит автоматизировать и значительно сократить расходы на процессы подготовки оборудования перед отправкой клиенту, обновления программного обеспечения на устройствах клиентов. Позволит сократить расходы на защиту программного обеспечения от копирования.

Проектирование и разработка обозначенной информационной системы проводилась в офисном помещении компании ООО «Универсальные терминал системы», находящемся по адресу Россия, Томская область, город Томск, улица Мокрушина, дом 19 строение 40, этаж 2, кабинет 18. Весь проводимый объём работ связан с вычислительной техникой, а именно персональным компьютером с периферийными устройствами ввода-вывода информации.

## **5.2 Правовые и организационные вопросы обеспечения безопасности**

В соответствии с ГОСТ 12.2.032-78 ССБТ «Рабочее место при выполнении работ сидя. Общие эргономические требования» к рабочему месту предъявляются следующие основные требования [28]:

- Конструкцией рабочего места должно быть обеспечено выполнение трудовых операций в пределах зоны досягаемости моторного поля;
- При организации рабочего места следует учитывать антропометрические показатели женщин (если работают только женщины) и мужчин (если работают только мужчины); если работают и женщины, и мужчины – общие средние показатели женщин и мужчин;
- Конструкцией рабочего места должно быть обеспечено оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, сиденья и пространства для ног.

В соответствии с трудовым кодексом РФ и правовыми нормами обеспечения безопасности, указанных выше, предусмотрена рациональная организация труда в течение смены, которая предусматривает:

- длительность рабочей смены не более 8 часов;
- установление двух регламентируемых перерывов (не менее 20 минут после 1-2 часов работы, не менее 30 минут после 2 часов работы);
- обеденный перерыв не менее 40 минут.

Каждый сотрудник должен пройти инструктаж по технике безопасности перед приемом на работу и в дальнейшем, должен быть пройден инструктаж по электробезопасности и охране труда.

### **5.3 Производственная безопасность**

Поскольку все работы при разработке и эксплуатации решения выполняются в помещении, далее офис, с постоянно работающими электронными устройствами, то основными источниками вредных и опасных факторов являются электронно-вычислительные устройства и элементы электрической сети офиса. Перечень вредных и опасных факторов, характерных для настоящего решения представлены ниже (таблица 5.1).

Технологический процесс представляет собой работы пользователя с программным продуктом за персональной ЭВМ.

Таблица 5.1 - Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разрабо тка	Изготов ление	Эксплу атация	
1.Отклонение показателей микроклимата	+	+	+	1. Уровень электромагнитного излучения: СанПиН 2.2.2/2.4.1340-03 [29]; 2. Требования к микроклимату: СанПиН 2.2.4.548–96 [30]; 3. Освещение – СанПиН 52.13330.2016 [31]; 4. Физические факторы - СанПиН 2.2.4.3359-16 [32]; 5. Шумы – СанПиН 2.2.4/2.1.8.562-96 [33];.
2.Превышение уровня шума	+	+	+	
3.Недостаточная освещенность рабочей зоны	+	+	+	
4.Повышенный уровень электромагнитных излучений	+	+	+	
5. Статическое электричество	+	+	+	
6. Опасность поражения электрическим током	+	+	+	

### 5.3.1 Анализ вредных и опасных производственных факторов

#### 5.3.1.1 Отклонение показателей микроклимата

Микроклимат производственных (рабочих) помещений – это комплекс метеорологических условий в помещении, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха, а также интенсивности теплового излучения от нагретых поверхностей. К показателям, характеризующим микроклимат в производственных помещениях, относятся:

- влажность воздуха – содержание в воздухе водяного пара.
- абсолютная влажность  $W$  – масса водяного пара в  $1 \text{ м}^3$  воздуха.
- максимальная влажность  $F$  – масса водяного пара, который может насытить  $1 \text{ м}^3$  воздуха при данной температуре.
- относительная влажность  $R$  – это отношение абсолютной влажности к максимальной.

Указанные параметры – каждый в отдельности и в совокупности – оказывают значительное влияние на работоспособность человека, его самочувствие и здоровье. При определенных их значениях человек испытывает состояние теплового комфорта, что способствует повышению

производительности труда, предупреждению простудных заболеваний. И, наоборот, неблагоприятные значения микроклиматических показателей могут стать причиной снижения производственных показателей в работе, привести к таким заболеваниям работающих как различные формы простуды, радикулит, хронический бронхит, тонзиллит и др.

Требования к микроклимату определяются исходя из категории тяжести работ (таблицы 5.2 и 5.3). Работа разработчика-программиста относится к первой категории тяжести в соответствии с СанПиН 2.2.4.548-96 [30].

Таблица 5.2 - Оптимальные и допустимые значения микроклимата

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный	21-23	20-24	40-60	0,1
Тёплый	23-25	22-26	40-60	0,1

Таблица 5.3 - Допустимые величины показателей микроклимата на рабочих местах

Период года	Температура воздуха, °С		Относительная влажность, %	Скорость движения воздуха, м/с	
	Ниже оптимальных, не более	Выше оптимальных, не более		Ниже оптимальных, не более	Выше оптимальных, не более
Холодный	20-21,9	24,1-25	15-75	0,1	0,1
Тёплый	21-22,9	25,1-28	15-75	0,1	0,2

Параметры микроклимата в помещении, где находится рабочее место, регулируются системой центрального отопления и приточновытяжной вентиляцией, и имеют следующие значения: влажность 45%, скорость движения воздуха 0,15 м/с, температура летом (тёплый период года) 23..25°С, зимой (холодный период года) 20..25°С.

Для обеспечения оптимальных показателей микроклимата на рабочем месте, в холодное время года используются пассивные радиаторы отопления, в теплое время года используется система кондиционирования воздуха.

### **5.3.1.2 Превышение уровня шума**

Шум - это совокупность различных звуков, возникающих в процессе производства и неблагоприятно воздействующих на организм. Это понятие обычно рассматривается с точки зрения экологии и медицины, то есть как угрозу жизнедеятельности, а не как фактор, мешающий работе, потому что постоянное его воздействие может принести непоправимый вред здоровью.

Шум может привести к нарушениям слуха (в случае постоянного нахождения при шуме более 85 децибел(dB)), может являться фактором стресса и повысить систолическое кровяное давление. В том числе, он может способствовать несчастным случаям, маскируя предупреждающие сигналы и мешая сконцентрироваться.

Источниками шума по СанПиН 2.2.4/2.1.8.562-96 являются [33]:

- вычислительные электронные цифровые машины;
- персональные вычислительные электронные цифровые машины (включая портативные ЭВМ);
- периферийные устройства: принтеры, сканеры, модемы, сетевые устройства, блоки бесперебойного питания и т.д.

В помещениях, оборудованных ПЭВМ, которые являются основным источником шума при выполнении данных видов работ, уровень шума на рабочем месте не должен превышать 50 дБ.

### **5.3.1.3 Недостаточная освещенность рабочей зоны**

Рациональное освещение помещений и рабочих мест – одно из важнейших условий создания благоприятных и безопасных условий труда. Нерационально организованное освещение может явиться причиной травматизма: плохо освещенные опасные зоны, слепящие источники света и блики от них, резкие тени и пульсации освещенности ухудшают видимость и могут вызвать неадекватное восприятие наблюдаемого объекта.

*Искусственное* освещение предусматривается в помещениях, в которых испытывается недостаток естественного света, а также для освещения помещения в те часы суток, когда естественная освещенность отсутствует. По принципу организации искусственное освещение можно разделить на два вида [31]:

- *Общее освещение.* Предназначено для освещения всего помещения, оно может быть равномерным или локализованным. Общее равномерное освещение создает условия для выполнения работ в любом месте освещаемого пространства. При общем локализованном освещении светильники размещают в соответствии с расположением оборудования, что позволяет создавать повышенную освещенность на рабочих местах.

- *Комбинированное освещение.* Состоит из общего и местного, его целесообразно устраивать при работах высокой точности, а также при необходимости создания в процессе работы определенной направленности светового потока.

*Местное* освещение предназначено для освещения только рабочих поверхностей и не создает необходимой освещенности на прилегающих к ним участках. Оно может быть стационарным и переносным [31].

Размещение светильников в помещении определяется следующими параметрами, м (рисунок 5.1):

$H$  – высота помещения;

$h_c$  – расстояние светильников от перекрытия (свес);

$h_n = H - h_c$  – высота светильника над полом, высота подвеса;

$h_{рп}$  – высота рабочей поверхности над полом;

$h = h_n - h_{рп}$  – расчётная высота, высота светильника над рабочей поверхностью.

$L$  – расстояние между соседними светильниками или рядами;

$l$  – расстояние от крайних светильников или рядов до стены;

Оптимальное расстояние  $l$  от крайнего ряда светильников до стены рекомендуется принимать равным  $L/3$ .

Интегральным критерием оптимальности расположения светильников является величина  $\lambda = L/h$ , уменьшение которой удорожает устройство и обслуживание освещения, а чрезмерное увеличение ведёт к резкой неравномерности освещённости.

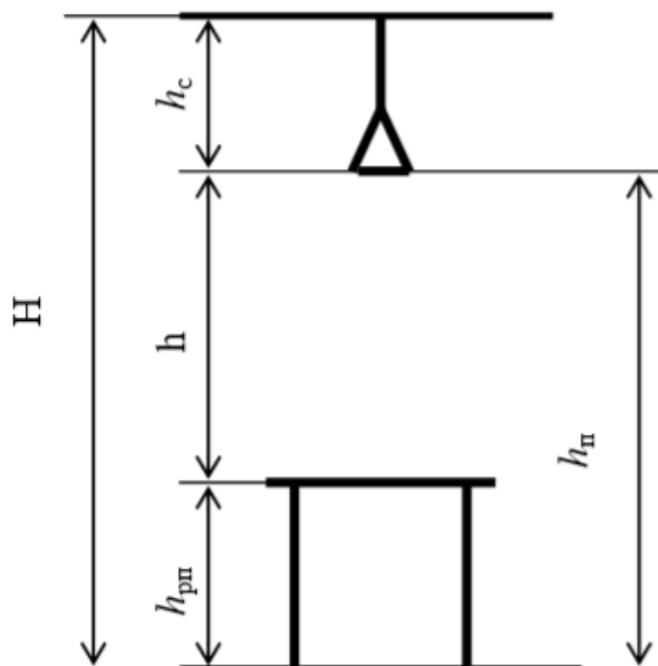


Рисунок 5.1 - Основные расчетные параметры

Расчет параметров освещения осуществлен для светильника ШОД – 2-40, характеристики которого приведены в таблице ниже [34].

Таблица 5.4 – Характеристика светильника ШОД 2-40

Количество и мощность лампы, Вт	Размеры, мм			КПД, %
	Длина	Ширина	Высота	
2 x 40	1230	266	198	85

Исходные характеристики помещения:

$H = 3$  м;  $A = 4$  м;  $B = 7$  м;  $h_{pn} = 0,7$  м;  $\lambda = 1,1$ ;  $h_c = 0,198$  м;

Тогда, расчётная высота, высота светильника над рабочей поверхностью равна:  $h = h_n - h_{pn} = H - h_c - h_{pn} = 3 - 0,198 - 0,7 = 2,102$  м  $\sim 2,1$  м.

Соответственно, расстояние между соседними светильниками или рядами равно:  $L = \lambda * h = 1,1 * 2,1 = 2,31$  м.

Тогда, расстояние от крайних светильников или рядов до стены:

$$l = L / 3 = 2,31 / 3 = 0,77 \text{ м.}$$

Исходя из рассчитанных значений размещения светильников для люминесцентных ламп, определено, что в помещении с заданными характеристиками может быть размещено 6 светильников типа ШОД 2-40 (Рисунок 5.2).

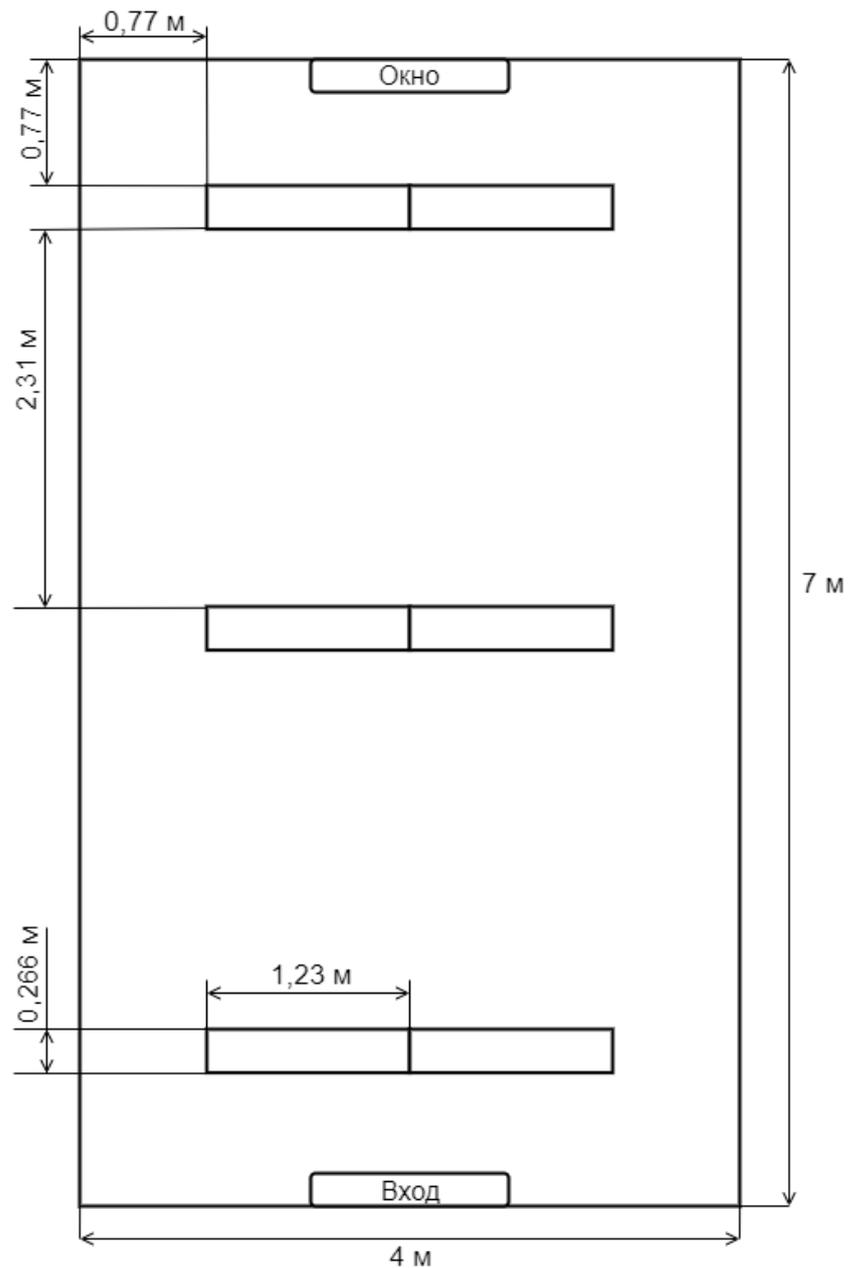


Рисунок 5.2 - План размещения общего освещения (вид сверху)

Работа оператора ЭВМ относится к третьему разряду зрительных работ и считается работой высокой точности [32]. Согласно СНиП 23-05-95 норма освещенности для рассматриваемого помещения при системе комбинированного освещения оставляет 400 Лк, в том числе от общего освещения – 200 Лк.

Расчёт общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента светового потока, учитывающим световой поток, отражённый от потолка и стен [31].

Световой поток лампы накаливания или группы люминесцентных ламп светильника определяется по формуле:

$$\Phi = \frac{E_H * S * K_3 * Z * 100}{n * \eta},$$

где  $E_H$  – нормируемая минимальная освещённость по СНиП 23-0595, лк;  $S$  – площадь освещаемого помещения, м<sup>2</sup>;  $K_3$  – коэффициент запаса, учитывающий загрязнение светильника (источника света, светотехнической арматуры, стен и пр., т. е. отражающих поверхностей), наличие в атмосфере цеха дыма, пыли;  $Z$  – коэффициент неравномерности освещения, отношение  $E_{cp}/E_{min}$ . Для люминесцентных ламп при расчётах берётся равным 1,1;  $N$  – число ламп в помещении;  $\eta$  – коэффициент использования светового потока.

Коэффициент использования светового потока показывает, какая часть светового потока ламп попадает на рабочую поверхность. Он зависит от индекса помещения  $i$ , типа светильника, высоты светильников над рабочей поверхностью  $h$  и коэффициентов отражения стен  $\rho_C$  и потолка  $\rho_{П}$ .

Индекс помещения определяется по формуле:

$$i = \frac{S}{h * (A + B)},$$

где  $h$  – допустимая высота подвеса светильников с люминесцентными лампами;  $A$  – ширина;  $B$  – длина.

Помещение имеет длину  $A = 4$  м, ширину  $B = 7$  м, допустимая высота подвеса светильников с люминесцентными лампами  $h = 2,1$  м. Требуется создать освещение  $E = 400$  лк.

Стены помещения оклеены светлыми обоями, следовательно коэффициент отражения светлых стен  $\rho_C = 30\%$ , и светлого деревянного потолка  $\rho_{\Pi} = 50\%$ . Тогда коэффициент запаса  $K_3 = 1,5$  и коэффициент неравномерности  $Z = 1,1$  [34].

Определено, что в помещении можно разместить 3 ряда светильников с люминесцентными лампами, в каждом из которых можно разместить 2 светильника. С учетом того, что в каждом светильнике типа ШОД 2-40 установлено 2 люминесцентные лампы, общее число люминесцентных ламп в помещении  $N = 12$ .

Тогда индекс помещения будет равен:

$$i = \frac{S}{h*(A+B)} = \frac{4*7}{2,1*(4+7)} = \frac{28}{2,31} = 1, (21);$$

В соответствии со значениями коэффициентов отражения стен  $\rho_C = 30\%$ , потолка  $\rho_{\Pi} = 50\%$  и индекса помещения  $i = 1.21$  - коэффициент использования светового потока светильников с люминесцентными лампами  $\eta = 37\%$  [34].

Тогда световой поток равен:

$$\Phi = \frac{E_H * S * K_3 * Z * 100}{n * \eta} = \frac{200 * 28 * 1,5 * 1,1 * 100}{12 * 37} = 2081 \text{ Лм.}$$

Определяем потребный световой поток ламп в каждом из рядов:

Выбираем ближайшую стандартную лампу ЛД 40 Вт с потоком 2300 Лм.

Делаем проверку выполнения условия:

$$-10\% \leq \frac{\Phi_{\text{л.станд}} - \Phi_{\text{л.расч}}}{\Phi_{\text{с.станд}}} * 100\% \leq 20\%$$

Получаем:

$$-10\% \leq \frac{2300 - 2081}{2300} * 100\% \leq 20\% = -10\% \leq 9,52\% \leq 20\%$$

Определяем электрическую мощность осветительной установки:

$$P = 12 * 40 = 480 \text{ Вт.}$$

### 5.3.1.4 Повышенный уровень электромагнитных излучений

Основные работы, связанные с разработкой и эксплуатацией решения, выполняются в офисном помещении, где находятся различные технические приборы: персональные ЭВМ, устройства бесперебойного питания, факсы, принтеры, сканеры, мобильные устройства, электрическая проводка и прочее. Электромагнитное излучение на рабочем месте оператора ЭВМ является вредным фактором производственной среды.

Уровень допустимого электромагнитного излучения определен в СанПиН 2.2.2/2.4.1340-03 [29]. Санитарно-эпидемиологические нормы устанавливают допустимые уровни электромагнитного излучения, испускаемого персональным компьютером (таблица 5.5).

Для защиты оператора ЭВМ от электромагнитного излучения должны применяться следующие средства:

- Приэкранные защитные фильтры для видеомониторов. Оказываемое защитное действие - снижение уровня напряжённости электростатического и электрического полей, а также уменьшение бликов и повышение контрастности изображения;

- Нейтрализаторы электрических полей промышленной частоты. Оказываемое защитное действие - снижение уровня электрического поля промышленной частоты (50 Гц).

Таблица 5.5 - Временные допустимые уровни электромагнитного поля, создаваемого персональными ЭВМ на рабочих местах (по СанПиН 2.2.2/2.4.1340-03)

Наименование параметров	Диапазон частот	Временный допустимый уровень
Напряженность электрического поля	5 Гц - 2 кГц	25 В/м
	2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	5 Гц - 2 кГц	250 нТл
	2 кГц - 400 кГц	25 нТл
Напряженность электростатического поля		15 кВ/м

### 5.3.1.5 Статическое электричество

Неблагоприятные разрядные токи статического электричества обычно возникают при соприкосновении с любым элементом ЭВМ (корпусом, монитором, системным блоком, терминалом сбора данных, клавиатурой, мышью и т.д.). Для защиты от статического электричества применяют специальные средства защиты (Таблица 5.6).

Установлены следующие допустимые уровни статического электричества [35]:

- напряженность электростатического поля – 15 кВ/м;
- электростатический потенциал экрана монитора – 500 В.

Таблица 5.6 – Средства защиты от статического электричества (по ГОСТ 12.4.124-83)

Тип средства защиты	Виды средств защиты
Коллективные	Нейтрализаторы (аэродинамические, высоковольтные, индукционные, лучевые);
	Заземляющие устройства;
	Увлажняющие устройства (испарительные, распылительные);
	Экранирующие устройства (козырьки, перегородки);
	Антиэлектростатические вещества (вводимые в объем, наносимые на поверхность);
Индивидуальные	Антиэлектростатическая одежда;
	Антиэлектростатическая обувь;
	Антиэлектростатические предохранительные приспособления;
	Антиэлектростатические средства защиты рук.

### 5.3.1.6 Опасность поражения электрическим током

В процессе использования электроприборов и электрооборудования может возникнуть опасность поражения электрическим током. По опасности поражения током рабочая зона относится к помещениям без повышенной опасности. Чтобы исключить опасность поражения необходимо соблюдать следующие правила электробезопасности:

- перед включением прибора в сеть должна быть визуально проверена его электропроводка на отсутствие возможных видимых нарушений изоляции, а также на отсутствие замыкания токопроводящих частей на корпус;

- при появлении признаков замыкания необходимо немедленно, отключить от электрической сети устройство и устранить неисправность;

- запрещается при включенном устройстве одновременно прикасаться к приборам, имеющим естественное заземление (например, радиаторы отопления, водопроводные краны и др.)

- запрещается эксплуатация оборудования в помещениях с повышенной опасностью;

- запрещается включать и выключать устройство при помощи штепсельной вилки. Штепсельную вилку включать и выключать из розетки можно только при выключенном устройстве.

Существуют следующие способы защиты от поражения током в электроустановках:

- предохранительные устройства;
- защитное заземление;
- применение устройств защитного отключения (УЗО);
- зануление.

Самый распространенный способ защиты от поражения током при эксплуатации измерительных приборов и устройств - защитное заземление, которое предназначено для превращения "замыкания электричества на корпус" в "замыкание тока на землю" для уменьшения напряжения прикосновения и напряжения шага до безопасных величин (выравнивание потенциала).

### **5.3.2 Обоснование мероприятий по снижению воздействия**

В ходе анализа вредных и опасных факторов было выявлено, что показатели освещенности соответствуют нормам и не требуют дополнительных мероприятий по снижению факторов риска недостаточной освещенности в помещении.

Показатели повышенного уровня шума, также находятся в допустимых пределах за счёт использования современных сертифицированных персональных ЭВМ, соответствующих всем нормам экологической безопасности, что также обуславливает низкие риски связанные с возможностью поражения электрическим током, статическим электричеством и повышенным уровнем электромагнитного излучения.

Одним из вредных факторов, требующих мероприятий по снижению его воздействия является отклонение показателей микроклимата. Для обеспечения комфортных показателей микроклимата требуется проводить регулярные проветривания помещения, а для достижения комфортных температур: в жаркое время года использовать систему кондиционирования, которой оборудовано рабочее помещение, в холодное время года регулировать интенсивность центрального отопления.

#### **5.4 Экологическая безопасность**

Экологическая безопасность и охрана окружающей среды являются одними из важнейших факторов при выполнении работ любого характера. При работе в офисном помещении за персональным ЭВМ отсутствуют выбросы и другие влияния на окружающую среду. По этой причине объект, на котором производилась разработка программного обеспечения и объект, на котором будет производиться его эксплуатация операторами ЭВМ и пользователями программного приложения относятся к предприятиям пятого класса, размер санитарно-защитной зоны для которых равен 50 м [36].

Программный продукт, разработанный в ходе выполнения магистерской диссертации, не способен причинять вред окружающей среде ни на стадии разработки, ни на стадии эксплуатации. Однако, средства, применяемые для его разработки и средства, которые будут применяться при его эксплуатации могут причинить окружающей среде вред.

Следовательно, охрана окружающей среды сводится к устранению отходов бытового мусора и отходам жизнедеятельности человека. В случае

выхода из строя ПЭВМ, они списываются и отправляются на специальный склад, который при необходимости принимает меры по утилизации списанной техники и комплектующих [37].

Для организации искусственного освещения на рабочих местах используются люминесцентные лампы, которые также требуют особой утилизации, так как в них содержится от 10 до 70 мг ртути, являющейся чрезвычайно опасным веществом, способным стать причиной отравления живых существ, а также загрязняющее атмосферу, гидросферу и литосферу. Срок службы люминесцентных ламп составляет около пяти лет, по истечении которых, их необходимо сдавать на переработку в специальные пункты приёма [38].

## **5.5 Безопасность в чрезвычайных случаях**

### **5.5.1 Анализ вероятных ЧС, которые могут возникнуть на рабочем месте**

Наиболее вероятной чрезвычайной ситуацией в офисном помещении является возникновение пожара или взрыва.

В основном возникновение пожара является следствием нарушения правил пожарной безопасности и правил эксплуатации электрических устройств. Такая ситуация может возникнуть по ряду причин: короткое замыкание в электрической проводке; возгорание электрических устройств; возгорание мебели и устройств искусственного освещения.

### **5.5.2 Мероприятия по предотвращению ЧС**

К противопожарным мероприятиям в помещении относят следующие мероприятия [39]:

1. помещение должно быть оборудовано: средствами тушения пожара (огнетушителями, ящиком с песком, стендом с противопожарным инвентарем);

средствами связи; должна быть исправна электрическая проводка осветительных приборов и электрооборудования;

2. каждый сотрудник должен знать место нахождения средств пожаротушения и средств связи; помнить номера телефонов для сообщения о пожаре; уметь пользоваться средствами пожаротушения.

Помещение, в котором проходит разработка настоящей дипломной работы, обеспечено средствами пожаротушения в соответствии с нормами:

1. пенный огнетушитель ОП-10 – 1 шт;
2. углекислотный огнетушитель ОУ-5 – 1 шт.

При невозможности самостоятельно потушить пожар необходимо вызвать пожарную команду, после чего поставить в известность о случившемся инженера по технике безопасности.

Вынужденная эвакуация при пожаре протекает в условиях нарастающего действия опасных факторов пожара. Кратковременность процесса вынужденной эвакуации достигается устройством эвакуационных путей и выходов, число, размеры и конструктивно-планировочные решения которых регламентированы строительными нормами СНиП 2.01.02-85.

Помещение и этаж оборудованы следующими средствами оповещения:

1. световая индикация в коридорах этажа;
2. звуковая индикация в виде громкоговорителя;
3. пассивные датчики задымленности.

## **5.6 Выводы по разделу**

В главе социальная ответственность настоящей диссертационной работы были рассмотрены вопросы обеспечения безопасных и безвредных условий труда. Также в главе были выделены факторы, оказывающие вредное и опасное влияние на исполнителя при разработке информационной системы.

В результате проведенной работы было выявлено, что рабочее место исполнителя оснащено достаточным для комфортной работы освещением, уровень шума соответствует нормам, безопасность уровня электромагнитного

излучения и опасности поражения электрическим током и статическим электричеством гарантируются использованием современного оборудования.

С точки зрения комфортности микроклимата рассматриваемого помещения есть смысл применить искусственную (механическую) вентиляцию (кондиционеры).

С точки зрения пожарной и экологической безопасности, рабочее место соответствует необходимым нормам.

## ЗАКЛЮЧЕНИЕ

По итогам выполнения магистерской диссертации была разработана система автоматизация процессов установки, обновления, запуска и контроля выполнения программного обеспечения в составе: настольного клиента UTS Launcher, представляющий собой кроссплатформенное приложение, основанное на фреймворках Electron и React.js и языке программирования TypeScript; консольного приложение упаковки пакетов для их публикации UTS Packer, основанное на Node.js; backend сервисов управления устройствами и лицензиями на Node.js. которые интегрированы в веб-сервисы UTS Marketplace; страниц управления устройствами, лицензиями и страница привязки лицензии в веб-клиенте UTS Marketplace.

Для создания системы был проведен анализ предметной области, проведено проектирование информационной системы и базы данных, осуществлен аналитический обзор используемых технологий и реализация запланированного функционала и интерфейса системы.

Созданная система позволяет автоматизировать процессов установки, обновления, запуска и контроля выполнения программного обеспечения.

Разработанное решение предоставляет функционал, который требуется компании для упрощения и автоматизации бизнес-процессов и упрощения взаимодействия отделов между собой. Результатом работы является:

- Сокращение расходов на техническую поддержку клиентов компании и увеличение эффективности работы отдела технической поддержки благодаря автоматизации развертывания программного обеспечения и его обновления;
- Экономия времени и «нервов» клиентами при обновлении программного обеспечения, что позволит им быть более лояльными к компании.
- Сокращение времени на выяснение информации об установленном программном обеспечении у клиента благодаря единой базе клиентов и их устройств;
- Оперативное исправление ошибок в программном обеспечении у всей базы клиентов без привлечения к процессу клиентов;

– Решение проблемы с необходимостью выяснения актуальной версии программного обеспечения;

– Готовое решение для непрерывной доставки (*CD*) и его интеграция с системами непрерывной интеграции (*CI*);

– Защита продуктов при их распространении.

Разработанная система является востребованной на сегодняшний день и имеет практическую значимость: система внедрена в организацию и успешно используется в бизнес-процессах компании ООО «Универсальные терминал системы».

## CONCLUSION

In results of master's thesis, a software deploy update and control system was developed, in which: the cross platform desktop application UTS Launcher based on React and Electron frameworks and written on TypeScript, the console application UTS Packer for software distribution packages creation written on Node.js, an backend services for devices and licenses control written on Node.js and integrated with UTS Marketplace services, an licenses and devices control pages and license attach page at web-client UTS Marketplace.

For system creation the domain analysis, system development, database development and analytical review of technologies used was investigated and planned functionality and user interfaces of the system was implemented.

The created system allows you to automate the processes of installing, updating, launching and monitoring the execution of software. The developed solution provides the functionality that the company needs to simplify and automate business processes and simplify the interaction of departments among themselves. The result of the work is:

- reducing the cost of technical support for company customers and increasing the efficiency of the technical support department by automating the deployment of software and updating it,
- saving time and "nerves" by customers when updating software, which will allow them to be more loyal to the company.
- reducing the time to find out information about the installed software from the client thanks to a single database of clients and their devices;
- prompt correction of errors in software for the entire customer base without involving clients in the process;
- solving the problem with the need to find out the current version of the software;
- a turnkey solution for continuous delivery (CD) and its integration with continuous integration systems (CI);
- protection of products during their distribution.

The developed system is in demand today and has practical significance: the system has been introduced into the organization and is successfully used in the business processes of Universal Terminal Systems LLC.

## Список использованных источников

- 1 Software Deployment // Wikipedia [Электронный ресурс]. Режим доступа: [https://en.wikipedia.org/wiki/Software\\_deployment](https://en.wikipedia.org/wiki/Software_deployment) (дата обращения 30.05.2020);
- 2 Paul Duvall. Automation for the people: Deployment-automation patterns, Part 1 // IBM Developer [Электронный ресурс]. Режим доступа: <https://www.ibm.com/developerworks/java/library/j-ap01139/j-ap01139-pdf.pdf> (дата обращения 30.05.2020);
- 3 Непрерывное развертывание ПО: автоматизация процессов сборки, тестирования и внедрения новых версий программ. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2011. — 432 с.;
- 4 Лицензия на программное обеспечение // Национальная библиотека им. Н. Э. Баумана [Электронный ресурс]. Режим доступа: [https://ru.bmstu.wiki/Лицензия\\_на\\_программное\\_обеспечение](https://ru.bmstu.wiki/Лицензия_на_программное_обеспечение) (дата обращения 30.05.2020);
- 5 Лицензирование программного обеспечения в России: законодательство и практика / А. И. Савельев. – М. : Инфотропик Медиа, 2012. – 432 с.;
- 6 ГК РФ Статья 1296. Произведения, созданные по заказу // Консультант Плюс. [Электронный ресурс]. Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_64629/e1a2a66199f94c314b8613a44d2abbaad7203f7ee/](http://www.consultant.ru/document/cons_doc_LAW_64629/e1a2a66199f94c314b8613a44d2abbaad7203f7ee/) (дата обращения 30.05.2020);
- 7 Симонович С.В. Информатика. Базовый курс: Учебник для вузов. Стандарт третьего поколения. – СПб.: Питер, 2015. – 640 с.;
- 8 Е. Н. Ливак. Лицензирование программного обеспечения // К вам пришла проверка (специализированное практическое издание). – Минск, 2005. – №7. – С. 39-45.
- 9 Груздев С.Л. Лицензирование программного обеспечения в сетях // [Электронный ресурс]. Режим доступа: [www.alladin.ru/secure/licence.htm](http://www.alladin.ru/secure/licence.htm) (дата обращения 30.05.2020);

10 Вичугова А.А., Вичугов В.Н., Дмитриева Е.А., Цапко Г.П. Информационные технологии: учебное пособие. – Томск: Изд-во Томского Политехнического университета, 2012. – 105 с.;

11 Интерактивный аэрохоккей Ice Hook. [Электронный ресурс]. Режим доступа: <https://ice-hook.ru/> (дата обращения 30.05.2020);

12 Интерактивная песочница ISandbox [Электронный ресурс]. Режим доступа: <https://isandbox.ru/> (дата обращения 30.05.2020);

13 Remote Desktop Protocol // Национальная библиотека им. Н. Э. Баумана [Электронный ресурс]. Режим доступа: [https://ru.bmstu.wiki/RDP\\_\(Remote\\_Desktop\\_Protocol\)](https://ru.bmstu.wiki/RDP_(Remote_Desktop_Protocol)) (дата обращения 30.05.2020);

14 TeamViewer // Национальная библиотека им. Н. Э. Баумана [Электронный ресурс]. Режим доступа: <https://ru.bmstu.wiki/TeamViewer> (дата обращения 30.05.2020);

15 Защита программ и приложений от копирования // Star Force [Электронный ресурс]. Режим доступа: <http://www.star-force.ru/> (дата обращения 30.05.2020);

16 Управление центром обработки данных // Microsoft System Center [Электронный ресурс]. Режим доступа: <https://www.microsoft.com/ru-ru/system-center> (дата обращения 30.05.2020);

17 Altiris // Свободная энциклопедия Википедия [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/Altiris> (дата обращения 30.05.2020);

18 Steam – превосходная игровая интернет-платформа // Steam [Электронный ресурс]. Режим доступа: [https://en.wikipedia.org/wiki/Steam\\_\(service\)](https://en.wikipedia.org/wiki/Steam_(service)) (дата обращения 30.05.2020);

19 Epic Games Store // About [Электронный ресурс]. Режим доступа: <https://www.epicgames.com/store/en-US/about> (дата обращения 30.05.2020);

20 Microsoft Store // Свободная энциклопедия Википедия [Электронный ресурс]. Режим доступа: [https://en.wikipedia.org/wiki/Microsoft\\_Store\\_\(digital\)](https://en.wikipedia.org/wiki/Microsoft_Store_(digital)) (дата обращения 30.05.2020);

21 Документация по MSIX // Microsoft Docs [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/windows/msix/> (дата обращения 30.05.2020);

22 Джефф Паттон. Пользовательские истории. Искусство гибкой разработки ПО. – изд. Питер, 2019. 288 с;

23 Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектноориентированный анализ и проектирование, 2е издание. – Пер. с англ. – СПб: Символ Плюс, 2007. – 624 с., ил;

24 Garrett, Jesse James. The Elements of User Experience: User-Centered Design for the Web and Beyond – Изд-во New Riders, 2010. 192 с.;

25 Методические указания к выполнению раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» для всех специальностей/ сост. В.Ю. Конотопский; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2015. – 29 с.;

26 Производственный календарь на 2020 год // Консультант Плюс. [Электронный ресурс]. Режим доступа: <https://www.consultant.ru/cons/cgi/online.cgi?rnd=FB165DD379C721997E49B07CBE3F55A9&req=obj&base=LAW&n=324428&dst=1&REFFIELD=134&REFDST=1000000006&REFDOC=324428&REFBASE=LAW&stat=picture%3D1%3Bindex%3D7> (дата обращения 30.05.2020);

27 Пашков Е.Н., Мезенцева. И.Л. Методические указания по разработке раздела «Социальная ответственность» выпускной квалификационной работы магистра, специалиста и бакалавра всех направлений (специальностей) и форм обучения ТПУ – Т. ТПУ, 2019;

28 ГОСТ 12.2.032-78 ССБТ «Рабочее место при выполнении работ сидя. Общие эргономические требования»;

29 СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы;

- 30 СанПиН 2.2.4.548–96. Гигиенические требования к микроклимату производственных помещений;
- 31 СанПиН 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95\*;
- 32 СанПиН 2.2.4.3359-16. Санитарно-эпидемиологические требования к физическим факторам на рабочих местах;
- 33 СанПиН 2.2.4/2.1.8.562–96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки;
- 34 Справочная книга для проектирования электрического освещения / Под ред. Г.М. Кнорринга. – СПб.: Энергоатомиздат, 1992. – 448 с.;
- 35 ГОСТ 12.4.124-83. Система стандартов безопасности труда. Средства защиты от статического электричества. Общие технические требования. Москва: Издательство стандартов, 1983. – 8 с.;
- 36 СанПиН 2.2.1/2.1.1.1200-03. Санитарно-защитные зоны и санитарная классификация предприятий, сооружений и иных объектов: утверждены Главным государственным санитарным врачом РФ 15 июня 2003 г.;
- 37 ГОСТ 17.4.3.04-85. «Охрана природы. Почвы. Общие требования к контролю и охране от загрязнения».
- 38 Постановление Правительства РФ от 03.09.2010 N 681 (ред. от 01.10.2013) "Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде // Консультант Плюс. 2015. [Электронный ресурс]. Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_104420/e1b31c36ed1083efeb6cd9c63ed12f99e2ca77ed/#dst100007](http://www.consultant.ru/document/cons_doc_LAW_104420/e1b31c36ed1083efeb6cd9c63ed12f99e2ca77ed/#dst100007) (дата обращения: 14.05.2020);
- 39 СНиП 2.01.02-85\* Противопожарные нормы;

**Приложение А**  
(справочное)

Licensed software delivery, installation and updating automation based on the  
UTS Marketplace service

**Студент**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8BM81	Ибетуллов Юрий Юзикович		

**Руководитель ВКР**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Руководитель ВКР, доцент ОИТ ИШИТР	Цапко Ирина Валерьевна	к.т.н.		

**Консультант-лингвист отделения иностранных языков ШБИП**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент ОИЯ	Сидоренко Татьяна Валерьевна	к.п.н.		

# 1 Domain Analysis

## 1.1 Software deployment automation

Software deployment is all the activities that make a software system available for use. This process is part of the software life cycle [1]. Software deployment is often treated as a necessary evil that can be cobbled together before going live. But you can and should apply software-engineering principles to deployment as well as to other parts of the development cycle. When manually applied, deployment is a repetitive and error prone process. Just as you can automate builds to reduce errors and speed software development, you can automate deployment processes to reduce errors and expedite software delivery [2].

Software deployment activities include:

- **release** is activity follows from the completed development process, and is sometimes classified as part of the development process rather than deployment process. It includes all the operations to prepare a system for assembly and transfer to the computer system(s) on which it will be run in production. Therefore, it sometimes involves determining the resources required for the system to operate with tolerable performance and planning and/or documenting subsequent activities of the deployment process,

- **installation and activation** - for simple systems, installation involves establishing some form of command, shortcut, script or service for executing the software (manually or automatically). For complex systems it may involve configuration of the system – possibly by asking the end-user questions about its intended use, or directly asking them how they would like it to be configured – and/or making all the required subsystems ready to use. Activation is the activity of starting up the executable component of software for the first time,

- **deactivation** is the inverse of activation, and refers to shutting down any already executing components of a system. Deactivation is often required to perform other deployment activities, e.g., a software system may need to be deactivated before an update can be performed. The practice of removing infrequently used or obsolete

systems from service is often referred to as application retirement or application decommissioning,

- **update** process replaces an earlier version of all or part of a software system with a newer release. It commonly consists of deactivation followed by installation,
- **version tracking** systems help the user find and install updates to software systems.

Modern methods of software development involve its deployment in several environments. The production and test environments are most commonly used. The production and test environments are most commonly used but there are many other environments which are different from product to product.

A production environment is environment when software used with actual data. Such environments have a special requirement - they must contain an up-to-date and stable version of the software product that has passed through various stages of testing. A testing environment is a setup of software and hardware for the testing teams to execute test cases. Ordinary users do not have access to this kind of environment.

There are differences between releasing software into production and deploying it to testing environments—not least, in the level of adrenaline in the blood of the person performing the release. However, in technical terms, these differences should be encapsulated in a set of configuration files. When deployment to production occurs, the same process should be followed as for any other deployment [3].

An important feature of software deployment in production environments is the ability to roll back the latest published version of a product. Customer dissatisfaction and financial loss may occur as a result of software errors after new version of software is release.

All of these processes—deploying to testing and production environments and rolling back—need to form part of your deployment pipeline implementation. It should be possible to see a list of builds available for deployment into each of these environments and run the automated deployment process by pressing a button or clicking a mouse to select the version to deploy and the environment to deploy it to. This, in fact, should be the only way to make changes to these environments of any

kind—including the configuration of the operating system and third-party software. Thus, it becomes possible to see exactly which versions of your application are in which environments, who authorized the deployment, and what changes have been made to the application since the last time it was deployed [3].

Releasing a new version of your application to a production environment you control is one thing. Releasing a new version of software installed by users on their own machines - client-installed software - is another. There are several issues to consider:

- managing the upgrade experience,
- migrating binaries, data, and configuration,
- testing the upgrade process,
- getting crash reports from users.

There are several ways in which clients can handle the upgrade process:

1. have your software check for new versions and prompt the user to download and upgrade to the latest version. This is the easiest to implement, but the most painful to use. Nobody wants to watch a download progress bar,

2. download in the background and prompt for installation. In this model, your software periodically checks for updates while running and downloads them silently. After the download is successful, it keeps prompting the user to upgrade to the latest version,

3. download in the background and silently upgrade the next time the application is restarted. Your application might also prompt you to restart now if you'd like to upgrade (as Firefox does).

Modern practice shows that the third option is the most effective way to update. However, in the case of upgrading, users have no understanding of why they might want to delay the upgrade. It forces them to think about upgrading without providing any information to help them decide one way or the other. As a result, the rational choice is usually not to upgrade, simply because any upgrade might break the application.

The correct solution is to make the upgrade process bullet proof—and to upgrade silently. In particular, if the upgrade process fails, the application should automatically revert to the previous version and report the failure to the development team. They can then fix the problem and roll out a new version which will (hopefully) upgrade correctly. All this can happen without the user even having to know anything. The only good reason for prompting the user is if there is some corrective action that needs to be taken.

## **1.2 Software licensing**

Any company protect its software from unauthorized use. Software is intellectual property product which has a protection. This protection applies to the legality of copying, changing a software product and is issued in the form of a license. Obtaining a license means acceptance of all points of use specified in the contract. Any unauthorized use of the software is an illegal act. A software license is a legal instrument governing the use or redistribution of software.

### **Types of Licenses**

Most distributed software can be categorized according to its license type [4]:

- public domain software which includes: Freeware, GPL, Adware, Postcardware, Donationware, Nagware/Beware and Open Source licenses,
- non-free software - the main goal of such programs is profit and its program code is closed.

Another kind of software is shareware which can be attributed to both groups. Such software is allowed to use for free, but with some limitations of the functionality that can be removed by paying for a license.

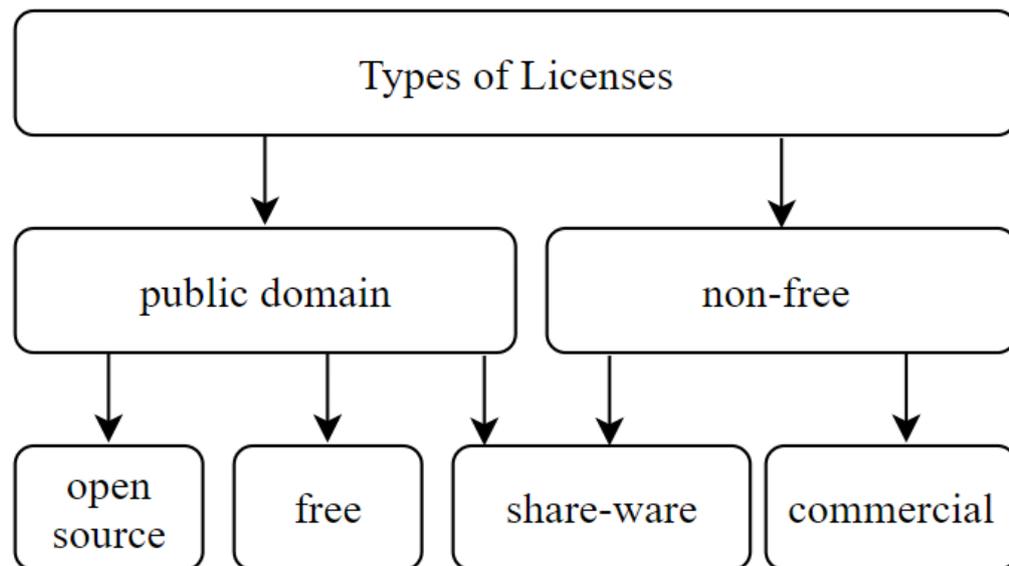


Figure 1.1 - Types of Licenses

Possible ways to use an intellectual property are enshrined in Art. 1270 of Civil Code of Russian Federation. Its includes the following powers [5]:

1. reproduction of a computer program - production of one or more copies, including a computer program record in the computer's memory,
2. distribution of a computer program by sale or other alienation of its original or copies,
3. import computer program instances for the purposes of their distribution,
4. rental copies of a computer program. The owner of the exclusive right to a computer program has this authority only if the computer program is the main rental object,
5. public execution of a computer program - its presentation using technical means in a place open for free visits, or in a place where there is a significant number of persons not belonging to the usual family circle,
6. processing a computer program - any change in it, including the translation of such a program or such a database from one language to another, with the exception of adaptation, i.e. making changes solely for the purpose of its functioning on specific technical means of the user or under the control of specific user programs,

7. bringing the computer program to the public in such a way that any person can access the computer program from anywhere and at any time of their choice (for example, posting it on the Internet).

The protection of copyright for the developer - these are the various schemes for licensing software. For each individual type of software product, different types of licensing are used. Depending on the volume of the granted license, many different types of licenses are allocated in commercial practice, in particular:

1. single user license provides the right to install and use the software product by one user on one device,

2. enterprise license allowing unlimited use of the software product within the framework of the organization of the licensee,

3. OEM-license (original equipment manufacturer) license refers to the license that a manufacturer installs on new devices. In this case the product key isn't transferable,

4. full Package Product license is a term used to describe a physical shrink-wrapped box (the packaging) of any of its licensed products. Full Packaged Product are the packages sent to retail stores and software retailers for consumer purchase.

Depending on the model of payment for the license, there are:

1. perpetual license it is the possession of an “timeless” right to use a software product,

2. temporary license implies the possession of a right of use limited by the established time period. In this case the user has full access to the product for a certain period after which the program loses some of the functionality,

3. subscription license. A yearly subscription license grants the user with a valid license key for the product for a period up until one year. When the one-year period expires, the user needs to purchase a new license key,

4. adware license is a license model when a software that generates revenue for its developer by automatically generating online advertisements in the user interface of the software or on a screen presented to the user during the installation process [7].

## **License enforcement**

Modern methods of software licensing, in addition to legal aspects, include a technical protection mechanism. License manager is a type of programs that solve the license technical problems whose task is to control licenses and restricting access to illegal users. Usually it's implemented by the user's key search in the list of legal users of the software product [8].

Another way is network applications monitor. Many systems record and track running applications, control the legality of all users of the system and use administrators monitor to display overall operation of the software.

A disadvantage of this approach is the need for administrator participation in license control. This method is not protected from cheating.

Modern software licensing systems to control and limit number of working programs at the same time [9]:

- compare the number of launched at the same time program copies with the permissible number of copies according to the license agreement,
- integrate accounting functions into applications,
- license control adherence by creating on the server a special license table.

### **1.4 Overview of existing analogues**

A software product required for meet the needs of the company. It must combine functionality of software launch, its execution control and automatically updating, license management and be able to integrate with existing UTS Marketplace services.

In the course of work, existing software products with similar functionality were studied, which can be divided into three groups:

1. services providing corporate environment management. Such services provide abilities to configure list of necessary software and deploy it on employer's personal computers and configure it. Often, such systems restrict the user of a personal computer by providing access to a predefined list of programs. An example of such services are: Microsoft System Center [16], Altiris [17], etc. In this work, this approach

to software distribution is not applicable due to the lack of the ability to connect clients to the corporate network,

2. online services providing services for the organization of application stores and distribution services. The basis of such services is the approach to providing business organization services to their clients (business to business). Most of the solutions on the market are highly specialized and provide functionality for creating software distribution systems for mobile platforms,

3. online services for the digital distribution of computer games and programs. Most often, such services distribute computer games. A distinctive feature of such services is the availability of a special program (launcher) for downloading, installing, integrity monitoring and a license. Existing distribution sites can be divided into two types: open and closed platforms. Closed platforms do not allow users to integrate their third party software in system and provide the opportunity to use the products of only certain companies. Open platforms are mainly aimed at attracting authors and publishing their software and content and charge a fixed percentage on each sale of a user-published software product.

Considered services of the second group:

**Ivanti** is a company that provides services for automating business processes of companies. One of the company's products, User Workspace Manager, is a workspace management solution that helps simplify desktop deployment and administration, reduce costs and ensure endpoint security. With User Workspace Manager, you can deliver a personalized, compatible desktop regardless of the location of the user or device. The solution also facilitates the execution of complex IT tasks, such as the migration of Windows 10, and provides secure file synchronization, sharing and data access.

The official website is <https://www.ivanti.ru>. Price policy - there is a free trial available upon request. The service is provided in the form of a corporate version for companies that is installed on the company's server. The price is determined depending on the configuration requirements with a monthly or quarterly payment and starts at \$ 1000 per month.

**AppDirect** provides the only comprehensive cloud commerce platform for success in the digital economy. AppDirect ecosystem connects developer and customer channels through its platform, simplifying the digital supply chain, allowing you to connect and sell products with third-party services for any channel, on any supported device. Using AppDirect, you can organize an online application store with the ability to install user-purchased applications on mobile devices and personal computers. The main emphasis of the service is in organizing and supporting the application store.

The official website is <https://www.appdirect.com>.

Pricing policy - a free tariff is available to get acquainted with the capabilities of the service, limited by the number of published goods (applications). Corporate rates are available, with a variety of options. The price is formed depending on the requirements of the configuration of the system configuration.

Among the online software distribution services, the following most popular services were highlighted: Steam, Epic Games Store, Microsoft Store. A distinctive feature of these services is the client program which is installed on a PC and downloads software, checks licenses, updates and provides the ability to run it.

**Steam** is a video game digital distribution service by Valve. Steam offers digital rights management (DRM), server hosting, video streaming, and social networking services. It also provides the user with installation and automatic updating of games, and community features such as friends lists and groups, cloud storage, and in-game voice and chat functionality [18].

Cross-platform support - Steam client is available on all major operating systems: Windows, Linux and macOS. The official website is <https://store.steampowered.com>. Pricing policy: 30% of each sale.

**Epic Games Store** an online digital gaming distribution service developed and operated by the American company Epic Games as well as Steam, Epic Games Store acts as a means of technical protection of copyrights, installs and updates programs on a user's computer [19].

Cross-platform support - Epic Games Launcher client is available on operating systems: Windows and macOS. Price policy: 12% of each sale, if the game is implemented on Unreal Engine game engine, then 5%.

**Microsoft Store** is a digital application store designed to publish and download Universal Windows Applications (UMG) and other digital content [20]. It is also possible to publish classic programs, but to publish them you need to additionally pack them in the MSIX format - the format of Windows application packages [21]. One of the features is the need for certification of published applications for their compatibility before publication. Microsoft Store provides developer tools for tracking store applications. You can track downloads, finances, crashes, reviews and ratings.

Cross-platform support - Microsoft Store Client is available only for Windows operating systems, starting with Windows 8. Pricing policy: 30% of each sale. Official site <https://www.microsoft.com/store/apps>.

In the course of the analysis of analogues of the developed system, the functionality of analogues was investigated. As a result of comparing and analyzing their capabilities, it was revealed that they cannot be used to replace the system being developed.

All the considered systems have a number of disadvantages that negatively affect the possibilities of implementation in the considered business processes of the company, namely:

- the high cost of systems designed to organize your own application store and services for its distribution,
- a highly specialized focus of software distribution services on the distribution of entertainment software,
- the difficulty of adapting existing systems to the needs of the company,
- the lack of the ability to allocate a separate store.