

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Школа информационных технологий и робототехники  
 Направление подготовки 09.04.01 «Информатика и вычислительная техника»  
 Отделение школы (НОЦ) Информационных технологий

### МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
<b>Разработка веб-приложения для формирования инвестиционного портфеля в соответствии с современной портфельной теорией</b>

УДК 004.774:004.925.84:330.322.1

Студент

Группа	ФИО	Подпись	Дата
8ВМ92	Иштуганов Руслан Александрович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Соколова В. В.	к.т.н., доцент		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Концепция стартап-проекта»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ШИП	Павлова И. А.	к.э.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООТ ШБИП	Антоневич О. А.	к.б.н., доцент		

### ДОПУСТИТЬ К ЗАЩИТЕ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель ООП	Кочегурова Е. А.	к.т.н., доцент		

Томск – 2021 г.

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

по направлению 09.04.01 «Информатика и вычислительная техника»

Код компетенции	Наименование компетенции
<b>Универсальные компетенции</b>	
УК(У)-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий
УК(У)-2	Способен управлять проектом на всех этапах его жизненного цикла
УК(У)-3	Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели
УК(У)-4	Способен применять современные коммуникативные технологии, в том числе на иностранном (-ых) языке (-ах), для академического и профессионального взаимодействия
УК(У)-5	Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия
УК(У)-6	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки
<b>Общепрофессиональные компетенции</b>	
ОПК(У)-1	Способен самостоятельно приобретать, развивать и применять математические, естественно-научные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте
ОПК(У)-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач
ОПК(У)-3	Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями
ОПК(У)-4	Способен применять на практике новые научные принципы и методы исследований
ОПК(У)-5	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
ОПК(У)-6	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования

ОПК(У)-7	Способен адаптировать зарубежные комплексы обработки информации и автоматизированного проектирования к нуждам отечественных предприятий
ОПК(У)-8	Способен осуществлять эффективное управление разработкой программных средств и проектов
<b>Профессиональные компетенции</b>	
ПК(У)-1	Способен разрабатывать и администрировать системы управления базами данных
ПК(У)-2	Способен проектировать сложные пользовательские интерфейсы
ПК(У)-3	Способен управлять процессами и проектами по созданию (модификации) информационных ресурсов
ПК(У)-4	Способен осуществлять руководство разработкой комплексных проектов на всех стадиях и этапах выполнения работ
ПК(У)-5	Способен проектировать и организовывать учебный процесс по образовательным программам с использованием современных образовательных технологий

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Информационных технологий и робототехники  
 Направление подготовки (специальность) 09.04.01 Информатика и вычислительная техника  
 Отделение школы (НОЦ) Информационных технологий

УТВЕРЖДАЮ:  
 Руководитель ООП  
 \_\_\_\_\_ Кочегурова Е. А.  
 (Подпись)     (Дата)

### ЗАДАНИЕ

#### на выполнение выпускной квалификационной работы

В форме:

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

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ВМ92	Иштуганову Руслану Александровичу

Тема работы:

Разработка веб-приложения для формирования инвестиционного портфеля в соответствии с современной портфельной теорией	
Утверждена приказом	Приказ №34-36/с от 03.02.2021

Срок сдачи студентом выполненной работы:	03.06.2021
--	------------

### ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p><b>Исходные данные к работе</b></p> <p><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 являются способы и методы расчета и оценки показателей риска и доходности портфеля ценных бумаг, а также методы оптимизации портфеля по показателям риска и доходности.</p>
---	--

<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b></p> <p><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>	<ol style="list-style-type: none"> <li>1. Исследование предметной области.</li> <li>2. Обзор и анализ существующих решений, конкурентный анализ.</li> <li>3. Обзор математического аппарата.</li> <li>4. Выбор технологий разработки.</li> <li>5. Проектирование инфраструктуры.</li> <li>6. Разработка веб-приложения.</li> <li>7. Тестирование веб-приложения.</li> <li>8. Размещение веб-приложения.</li> </ol>
<p><b>Перечень графического материала</b> <i>(с точным указанием обязательных чертежей)</i></p>	<p>Примеры работы веб-приложения в виде снимков экрана, диаграмма архитектуры приложения, диаграмма облачной инфраструктуры, диаграмма бизнес-модели Остервальда.</p>

<p><b>Консультанты по разделам выпускной квалификационной работы</b> <i>(с указанием разделов)</i></p>	
<p><b>Раздел</b></p>	<p><b>Консультант</b></p>
<p>Основная часть</p>	<p>Доцент ОИТ ИШИТР, к.т.н., доцент Соколова В.В.</p>
<p>Концепция стартап-проекта</p>	<p>Доцент ШИП ИШИТР, к.э.н., доцент Павлова И. А.</p>
<p>Социальная ответственность</p>	<p>Доцент ООТ ШБИП, к.б.н., доцент Антоневич О.А.</p>
<p>Английский язык</p>	<p>Доцент ОИЯ ШБИП, к.ф.н., доцент Аксёнова Н. В.</p>
<p><b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b></p>	
<p>Раздел 8 Тестирование</p>	

<p><b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b></p>	<p>01.03.2021</p>
--	-------------------

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Соколова В. В.	к.т.н., доцент		01.03.2021

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ВМ92	Иштуганов Руслан Александрович		01.03.2021

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Информационных технологий и робототехники  
 Направление подготовки (специальность) 09.04.01 Информатика и вычислительная техника  
 Уровень образования Магистратура  
 Отделение школы (НОЦ) Информационных технологий  
 Период выполнения \_\_\_\_\_ (осенний / весенний семестр 2020 /2021 учебного года)

Форма представления работы:

Магистерская диссертация

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

### КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	03.06.2021
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
03.06.2021	Основная часть	70
03.06.2021	Концепция стартап-проекта	10
03.06.2021	Социальная ответственность	10
03.06.2021	Обязательное приложение на иностранном языке	10

#### СОСТАВИЛ:

##### Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Соколова В.В.	к.т.н., доцент		

#### СОГЛАСОВАНО:

##### Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Кочегурова Е. А.	к.т.н., доцент		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«КОНЦЕПЦИЯ СТАРТАП-ПРОЕКТА»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
<b>8ВМ92</b>	<b>Иштуганов Руслан Александрович</b>

<b>Школа</b>	<b>ИШИТР</b>	<b>Направление</b>	<b>09.04.01 Информатика и вычислительная техника</b>
<b>Уровень образования</b>	магистр		

**Перечень вопросов, подлежащих разработке:**

Проблема конечного потребителя, которую решает продукт, который создается в результате выполнения НИОКР (функциональное назначение, основные потребительские качества)	Анализ и оптимизация рисков инвестиционного портфеля
Способы защиты интеллектуальной собственности	Регистрация патента
Объем и емкость рынка	Объем рынка частных инвесторов в РФ – 8 миллионов человек
Современное состояние и перспективы отрасли, к которой принадлежит представленный в ВКР продукт	Взрывной рост количества частных инвесторов в РФ, при этом существующие средства анализа инвестиций сфокусированы только на доходности
Себестоимость продукта	Себестоимость продукта для одного пользователя – 2 433 руб/год
Конкурентные преимущества создаваемого продукта	Количественный анализ риска инвестиционного портфеля, оптимизация портфеля по параметру риска
Сравнение технико-экономических характеристик продукта с отечественными и мировыми аналогами	На основании конкурентных преимуществ
Целевые сегменты потребителей создаваемого продукта	Индивидуальные инвесторы, нефинансовые организации, занимающиеся инвестициями свободных денежных средств
Бизнес-модель проекта	Модель по А. Остервальду и И. Пинье
Производственный план	Увеличивающийся, с предварительным темпом роста 10% в год, начиная с 4 840 платных подписчиков
План продаж	Прямая розничная продажа продукта, фриум подписочная модель

**Перечень графического материала:**

При необходимости представить эскизные графические материалы (например, бизнес-модель)	Расчет экономической целесообразности проекта Расчет себестоимости продукта
--	--

<b>Дата выдачи задания для раздела по линейному графику</b>	16.03.2021
---	------------

Задание выдал консультант по разделу «Концепция стартап-проекта» (со-руководитель ВКР):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ШИП	Павлова И. А.	к.э.н		16.03.2021

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ92	Иштуганов Руслан Александрович		16.03.2021

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ВМ92	Иштуганов Руслан Александрович

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

Тема ВКР:

**Разработка веб-приложения для формирования инвестиционного портфеля в соответствии с современной портфельной теорией**

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

<p>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</p>	<p>Объект исследования: методы построения портфелей ценных бумаг, оптимизации таких портфелей по параметрам риска и доходности и реализация данных методов в виде компьютерных алгоритмов Область применения: Инвестиционные компании Рабочее место: офис инвестиционной компании</p>
---	---

**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>	<p>СП 2.4.3648-20. Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования ГОСТ 21889-76. Система «человек-машина». Кресло человека-оператора. Общие эргономические требования. ГОСТ 22269-76. Система «человек-машина». Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования. ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре. ГОСТ Р ИСО 9241-5-2009. Эргономические требования к проведению офисных работ с</p>
--	---

	использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора.
<b>2. Производственная безопасность:</b> 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Вредные факторы -аномальные микроклиматические параметры воздушной среды; - отсутствие или недостатки необходимого искусственного освещения; - физические статические перегрузки, связанные с рабочей позой; - умственное перенапряжение, в том числе вызванное информационной нагрузкой; - повышенный уровень и другие неблагоприятные характеристики шума. Опасные факторы - разница электрических потенциалов, под действие которого попадает работающий.
<b>3. Экологическая безопасность:</b>	Воздействие на литосферу: - Утилизация вычислительных машин, информационных носителей
<b>4. Безопасность в чрезвычайных ситуациях:</b>	Возможные ЧС: - Пожар; Наиболее типичная ЧС: - Пожар.

Дата выдачи задания для раздела по линейному графику	01.03.2021
--	------------

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООТ	Антоневич Ольга Алексеевна	к.б.н.		01.03.2021

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ВМ92	Иштуганов Руслан Александрович		

# Содержание

Реферат .....	13
Определения, обозначения, сокращения, нормативные ссылки .....	15
1. Введение.....	17
2. Задачи исследования .....	23
2.1    Объект исследования .....	23
2.2    Предмет исследования .....	23
2.3    Цели и задачи исследования.....	23
3. Анализ предметной области.....	25
4. Обзор математического аппарата .....	26
4.1    Традиционные портфельные теории .....	26
4.2    Современные теории .....	27
4.3    Выводы по главе .....	39
5. Описание используемых технологий .....	40
5.1    Серверная часть .....	40
5.2    Контроль версий .....	46
5.3    Клиентская часть .....	46
5.4    Выводы по главе .....	46
6. Инфраструктура веб-приложения.....	48
6.1    Вычислительные ресурсы.....	49
6.2    Выводы по главе .....	51
7. Реализация.....	53
7.1    Серверная часть .....	53
7.2    Клиентская часть .....	55
7.3    Выводы по главе .....	56
8. Тестирование .....	57
8.1    Модульное тестирование .....	59
8.2    Интеграционное тестирование .....	62
8.3    Системное тестирование.....	64
8.4    Непрерывная интеграция .....	65
8.5    Выводы по главе .....	70
9. Развертывание инфраструктуры .....	71
9.1    Заказ вычислительных ресурсов .....	71
9.2    Конфигурирование системы.....	72
9.3    Настройка внешних модулей.....	73
9.4    Непрерывное развертывание .....	74
9.5    Выводы по главе .....	77
10. Описание работы веб-приложения.....	78
10.1    Общая статистика .....	78
10.2    Модель Марковица.....	78

10.3	Бюджетирование риска .....	79
10.4	Модель Блэка-Литтермана.....	80
10.5	Выводы по главе .....	82
11.	<b>ВКР как стартап .....</b>	<b>84</b>
11.1	Актуальность.....	84
11.2	Конкурентный анализ рынка.....	85
11.3	Сравнение существующих решений.....	102
11.4	Анализ целевой аудитории .....	104
11.5	Бизнес-модель .....	107
11.6	Оценка экономической целесообразности бизнеса.....	110
11.7	Стратегия продвижения .....	116
11.8	Защита интеллектуальной собственности.....	118
11.9	Выводы по главе .....	119
12.	<b>Социальная ответственность .....</b>	<b>120</b>
12.1	Введение .....	120
12.2	Правовые и организационные вопросы обеспечения безопасности .....	120
12.3	Производственная безопасность .....	124
12.4	Отсутствие или недостаток необходимого естественного и искусственного освещения.....	126
12.5	Акустические колебания в производственной среде .....	129
12.6	Электромагнитные полями, неионизирующие ткани тела человека.....	130
12.7	Разница электрических потенциалов, под действие которого попадает работающий 131	
12.8	Физические статические перегрузки, связанные с рабочей позой. ....	132
12.9	Умственное перенапряжение .....	133
12.10	Экологическая безопасность .....	134
12.11	Безопасность в чрезвычайных ситуациях .....	134
12.12	Выводы по главе .....	136
13.	<b>Заключение .....</b>	<b>138</b>
14.	<b>Публикации и достижения .....</b>	<b>140</b>
15.	<b>Список использованных источников .....</b>	<b>141</b>
16.	<b>Приложение А .....</b>	<b>145</b>
16.1	Software Testing.....	146
16.2	Conclusion .....	158

# Реферат

Работа содержит 158 страниц, 59 рисунков, 8 таблиц и 45 использованных источников.

ПОРТФЕЛЬ, ЭФФЕКТИВНЫЙ ПОРТФЕЛЬ, ЛИНИЯ РЫНКА КАПИТАЛА, ЭФФЕКТИВНАЯ ГРАНИЦА, АКЦИЯ, ЦЕННАЯ БУМАГА, РИСК, ДОХОДНОСТЬ, МАРКОВИЦ, БЛЭК-ЛИТТЕРМАН.

Объектом исследования являются способы и методы расчета и оценки показателей риска и доходности портфеля ценных бумаг, а также методы оптимизации портфеля по показателям риска и доходности.

Предметом исследования являются компьютерные алгоритмы, реализующие рассматриваемые методы и модели, результаты работы таких алгоритмов в виде рекомендаций по ребалансировке портфеля для достижения задачи оптимизации портфеля по параметру риска.

Итоговой целью работы является создание веб-приложения, позволяющего пользователю выполнить оценку риска инвестиционного портфеля и выполнить оптимизацию портфеля с целью повышения доходности при заданном уровне риска или снижения риска при заданном уровне доходности.

Результатом ВКР является веб-приложение.

Веб-приложение реализовано согласно парадигме SaaS и доступно с любого устройства с выходом в сеть Интернет.

В настоящее время приложение находится на стадии закрытого бета-тестирования небольшой группой отобранных пользователей и ведется сбор и анализ отзывов и пожеланий пользователей.

Приложение планируется внедрить в коммерческую эксплуатацию в виде стартапа.

В будущем планируется сфокусироваться на реализации функционала оценки истинной стоимости акционерного капитала компании, который будет выражать ожидаемую доходность акции данной компании методами машинного

обучения и искусственного интеллекта. Данное исследование планируется выполнять в аспирантуре.

## **Определения, обозначения, сокращения, нормативные ссылки**

1. НАУФОР (сокр.) – Национальная Ассоциация Участников Фондового Рынка.
2. SaaS (Software As A Service) – программное обеспечения как услуга.
3. Инфраструктура как код – методология, а также набор инструментов и практик развертывания и конфигурирования вычислительных ресурсов с помощью кода, на том или ином языке программирования, зафиксированного в виде конфигурационных файлов, в отличии от использования интерактивных инструментов запуска и конфигурации провайдера облачных ресурсов.
4. CI (Continuous Integration) – непрерывная интеграция.
5. CD (Continuous Delivery) – непрерывное развертывание.
6. Инвестирование – размещения капитала с целью получения прибыли.
7. Инвестиция на фондовом рынке – правовое и экономическое соглашение, заключённое между двумя лицами, юридическим со стороны продавца и юридическим или физическим со стороны покупателя по поводу приобретения части прав на капитал юридического лица.
8. Риск – вероятность непредвиденного изменения цены.
9. Метод дисконтирования денежных потоков – анализ стоимости бизнес-собственности, основанный на оценке ожидаемых доходов от оцениваемых активов, приведенный на сегодняшний день по требуемой ставке дисконтирования.
10. MVCT (Model ViewController Template) – модель – представление – контроллер – шаблон.
11. SQL (Structured Query Language) – структурированный язык запросов.
12. СПТ – современная портфельная теория.
13. VaR (Value At Risk) – Стоимость под риском.
14. ORM (Object Relation Mapper) – объектно-реляционное отображение.
15. HTML (Hyper Text Markup Language) – язык гипертекстовой разметки.

16. CSS (Cascading Style Sheet) – каскадные страницы стилей.
17. JSON (Java Script Object Notation) – Объектная нотация языка JavaScript.

# 1. Введение

В современных условиях деньги являются неотъемлемым атрибутом хозяйственной жизни предприятия. Денежные средства являются единственным видом оборотных средств, обладающим абсолютной ликвидностью. Все сделки, связанные с поставкой материальных ценностей и оказанием услуг, завершаются денежными расчетами.

Сущность денег заключается в том, что это – специфический товарный вид, с натуральной формой которого сростается общественная функция всеобщего эквивалента.

Сущность денег проявляется в единстве трех свойств:

1. всеобщей непосредственной обмениваемости;
2. кристаллизации меновой стоимости;
3. материализации всеобщего рабочего времени.

Именно поэтому деньги в современном мире играют весьма важную роль. Без них общество не могло бы развиваться и функционировать. Несомненно, деньги пронизывают все сферы жизни общества.

Деньги играют эту роль не только в жизни общества в целом, но и в жизни каждого, отдельного взятого человека. Потребность в деньгах может вынуждать человека прибегать к определенным социальным связям и действиям. Хотя потребности в деньгах как таковой не существует. Под «потребностью в деньгах» можно понимать потребности, которые можно удовлетворить с помощью денег, т.е. можно сказать и так, что главная роль денег в жизни человека заключается в том, что человек с их помощью может ощутить свободу выбора. Причем чем больше будет денег у конкретного человека, тем большую свободу в своей жизни он способен ощутить, т.е., деньги открывают для любого человека финансовую свободу.

Причем с возрастом, потребность в деньгах у человека только растет. Ведь с течением времени способность зарабатывать деньги ухудшается в связи с процессом деградации как физических, так и умственных способностей, а

потребности, и в первую очередь медицинского характера в связи с теми же самыми процессами, растут.

Так, согласно оценке сайта Our World in Data [1], к 2100 году прирост населения в возрасте 60 лет и выше значительно превзойдет прирост населения в возрасте от 0 до 40 лет (Рисунок 1). Это означает, что прогнозируется глобальное старение населения. Поэтому можно предположить, что потребность и спрос на медицинские услуги будет только расти, так же, как и их стоимость.

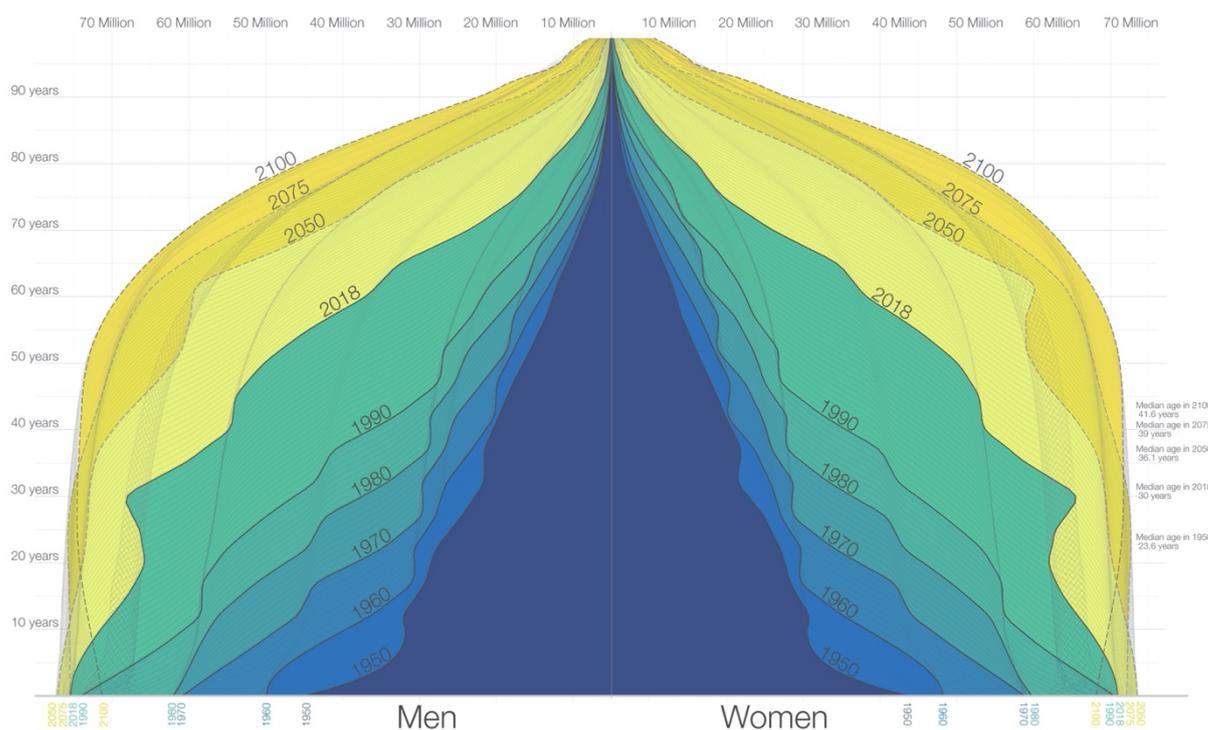


Рисунок 1 – Динамика изменения возраста населения Земли

Причем данная картина характерна как для всего мира в целом, так и для России в частности, где в ближайшие несколько лет также будет наблюдаться сдвиг среднего возраста населения к 50-60 годам (Рисунок 2).

# Демография России 1897–2036

Меньше людей  Больше людей

В 1984–1991 годах родилось **последнее большое поколение**.  
Это отзвук демографического эха **трагедий XX века**.

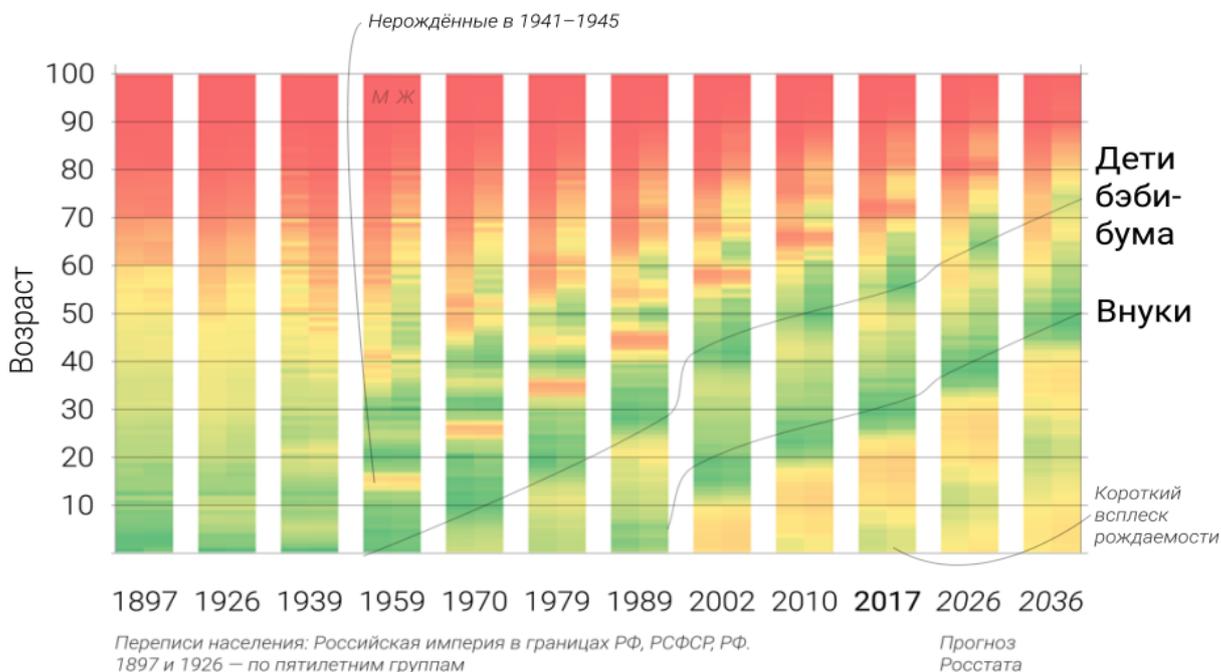


Рисунок 2 – Динамика изменения возраста населения РФ

Таким образом, можно сделать вывод, что вопрос зарабатывания, и, что не менее важно, сохранения и приумножения денег является чрезвычайно важной задачей в жизни человека.

Для физических лиц существует немного способов сохранить и приумножить свои сбережения, это:

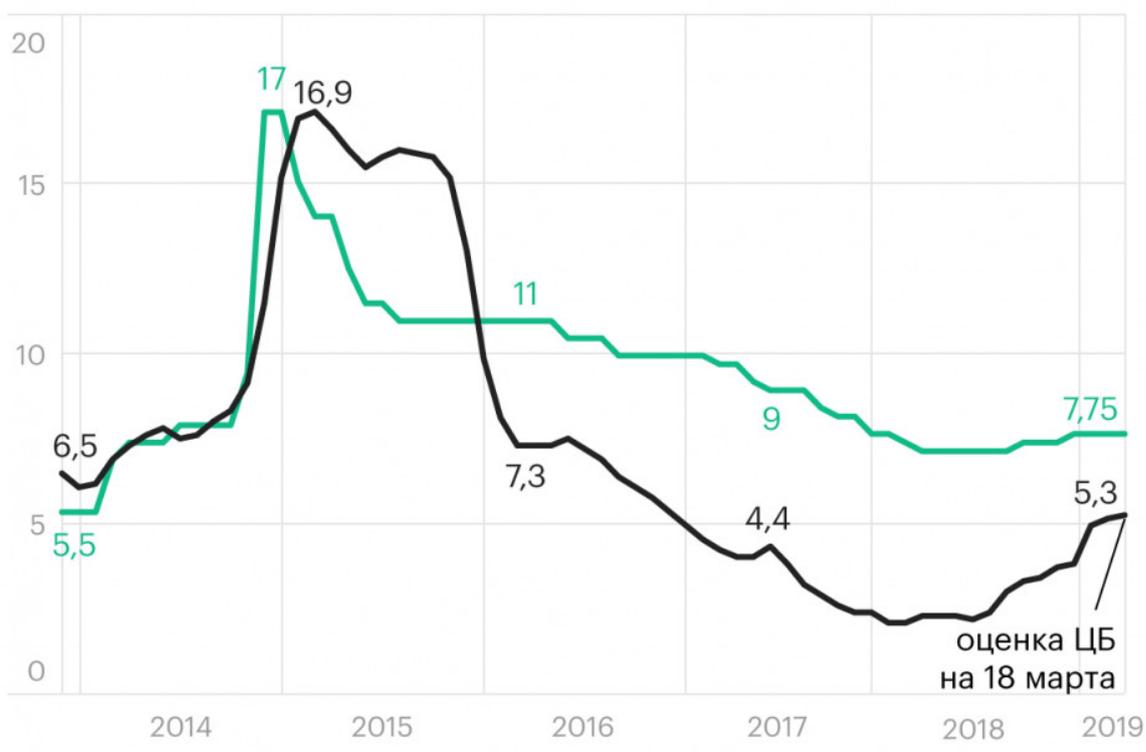
- банковский вклад;
- инвестиции в ценные бумаги.

Банковские вклады являются средством приумножения исключительно номинально, т.к. по любым депозитам процентная ставка будет либо равна темпу инфляции, либо быть ниже его (Рисунок 3).

## Как меняются ставка ЦБ и инфляция

— Ключевая ставка ЦБ, %

— Инфляция (г/г), %



Источник: Банк России, терминал Bloomberg

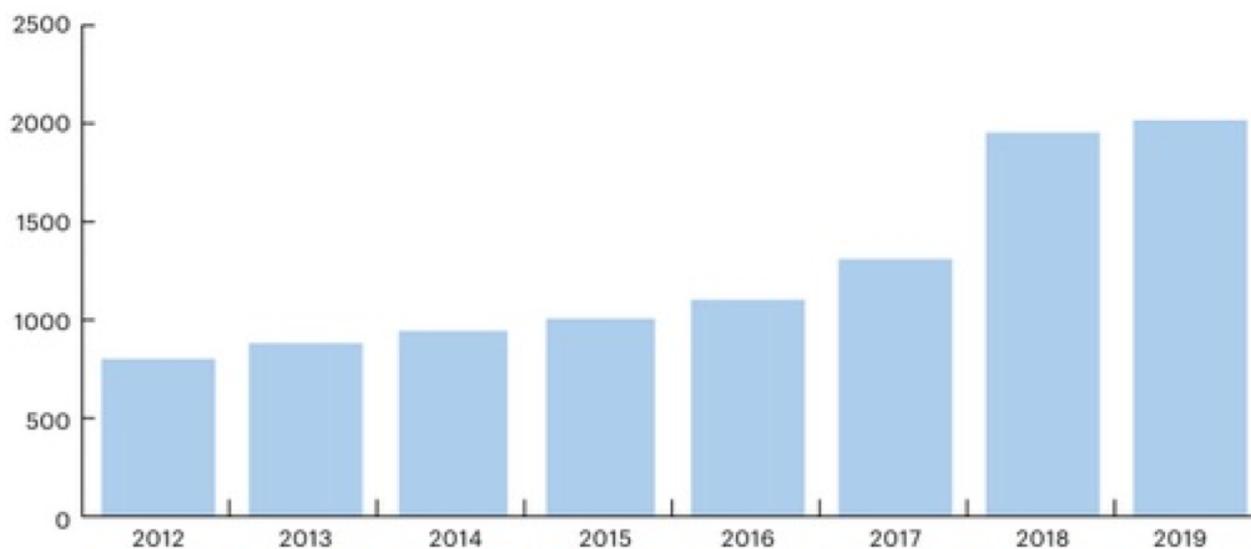
© РБК, 2019

Рисунок 3 – Темпы инфляции и ключевая ставка ЦБ в РФ

Поэтому именно инвестиции в ценные бумаги остаются единственным, доступным широкому кругу людей, способом получить доходность не только превышавшую темпы инфляции, т.е. не только сохранить деньги, но и приумножить их, зачастую более чем в 50 - 100 и более процентов.

По этой причине, что число россиян, вкладывающих деньги на фондовом рынке, стремительно растет. Так, по данным опроса НАУФОР, общая сумма средств, размещенная российскими гражданами на фондовом рынке, в течение 2018 года удвоилась и превысила 2 трлн руб. Это, конечно, пока еще на порядок меньше объема вкладов в банках (28,8 трлн руб.), но динамика впечатляет: банковские вклады за прошлый год выросли всего лишь на 9,5 %, и это с учетом начисленных процентов, т.е. реальный рост был близок к нулевому (Рисунок 4).

## Динамика количества индивидуальных инвесторов в России тыс. человек на конец года



Источник: НАУФОР

Рисунок 4 – Динамика количества индивидуальных инвесторов в РФ

Этот тренд вызван комплексом причин, главными из которых являются снижение доходности депозитов на фоне установившейся низкоинфляционной конъюнктуры, а также сокращение эффективности вложений в недвижимость.

Однако стоит отметить, что если история фондовых рынков в мире началась более 300 лет назад, то в России фондовому рынку – около 30 лет. Культура инвестирования и управления портфелем инвестиций в России находится в зачаточном состоянии. Принимая решение вложить сбережения в те или иные ценные бумаги, инвесторы зачастую руководствуются только рекламными проспектами, брошюрами и статьями, оперирующими только одним аспектом инвестирования – доходностью, который несомненно является основной причиной инвестиций. Однако, как будет показано в ходе настоящего исследования, доходность инвестиций напрямую связана с их риском, который в экономике и финансах понимается иначе по сравнению с общепринятым пониманием этого термина.

При этом даже в англоязычном мире, не говоря уже о России, отсутствуют инструменты, доступные широкому кругу частных инвесторов для оценки рискованности инвестиций и наглядной ее демонстрации, для

оптимизации портфеля ценных бумаг с целью достижения минимального риска или максимальной доходности.

Таким образом, существует острая необходимость в исследовании методов оценки рискованности инвестиций и разработки ПО для реализации того или иного метода в виде удобного пользователю приложения.

## **2. Задачи исследования**

### **2.1 Объект исследования**

Объектом настоящего исследования являются способы и методы расчета и оценки показателей риска и доходности портфеля ценных бумаг, а также методы оптимизации портфеля по показателям риска и доходности для достижения требуемого соотношения риска и доходности.

За всю историю развития фондового рынка было предложено несколько теорий, решающих данные задачи в той или иной мере. Условно, данные теории можно разделить на традиционные и современные:

1. Традиционные теории:
  - а. Теория Доу-Джонса.
  - б. Теория случайного блуждания.
  - в. Формульная теория.
2. Современные теории:
  - а. Модели оптимизации дисперсии среднего значения.
  - б. Модели бюджетирования.
  - в. Байесовские модели.

В настоящем исследовании будет выполнен сравнительный анализ данных теорий и предлагаемых ими способов для решения поставленных задач.

### **2.2 Предмет исследования**

Предметом исследования являются компьютерные алгоритмы, реализующие рассматриваемые методы и модели, результаты работы таких алгоритмов в виде рекомендаций по ребалансировке портфеля для достижения задачи оптимизации портфеля по параметру риска, а также графики, демонстрирующие те или иные аспекты работы реализуемых компьютерных алгоритмов.

### **2.3 Цели и задачи исследования**

Таким образом, целью данного исследования являются:

1. Реализация веб-приложения для оптимизации портфеля ценных бумаг на основе современной портфельной теории.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Выполнить обзор современных портфельных теорий.
2. Выполнить сравнительный анализ аналогичных веб-решений, обосновать актуальность разработки нового программного продукта (веб-приложения) и определить его архитектуру.
3. Проанализировать преимущества и недостатки существующих портфельных теорий.
4. Определить средства разработки веб-приложения, средства тестирования и дать их описания.
5. Спроектировать и разработать бизнес-логику веб-приложения на основе математического аппарата, реализованного в рамках выбранной теории.
6. Спроектировать инфраструктуру для данного веб-приложения и выполнить ее развертывание и конфигурирование.
7. Реализовать систему тестирования веб-приложения и настроить конвейер непрерывной интеграции и непрерывного развертывания.
8. Развернуть веб-приложение на вычислительных мощностях и запустить его в работу.

### **3. Анализ предметной области**

Детальный анализ предметной области приведен в Главе № 11.2 – «Конкурентный анализ рынка», однако результаты анализа текущего состояния рынка подобных решений и конкурентного анализа состоят в следующем: в настоящее время на рынке отсутствуют простые, доступные широкому кругу инвесторов, в особенности розничных, инструменты для анализа риска инвестиционного портфеля, для оптимизации портфеля по совокупности параметров доходности и риска, для наглядного отображения степени риска портфеля в зависимости от того, какие бумаги и в каких пропорциях он содержит, т.е. для анализа риска инвестиций так, как риск понимается в экономике, финансах и в соответствии с современной портфельной теорией.

Очевидно, что программная реализация веб-приложения по результатам данного исследования:

1. Имеет высокую практическую ценность, т.к. будет создан удобный и полезный инструмент для инвесторов.
2. Обладает значимой общественной ценностью, т.к. поможет значительно повысить финансовую грамотность инвесторов, настоящих и потенциальных, показав второй аспект инвестирования – риск.

## **4. Обзор математического аппарата**

В рамках ВКР математический аппарат, описывающий используемые в веб-приложении модели, представлен такими разделами математики как линейная алгебра и теория вероятностей. Ниже представлен обзор теорий и моделей, реализующих данный математический аппарат для решения прикладных задач, заявленных в целях и задачах настоящей ВКР.

Выделяются следующие портфельных теории на основании классификации их по времени возникновения:

1. Традиционные теории:
  - а. Теория Доу-Джонса.
  - б. Теория случайного блуждания.
  - в. Формульная теория.
2. Современные теории:
  - а. Модели минимизации дисперсии среднего значения.
  - б. Модели бюджетирования.
  - в. Байесовские модели.

### **4.1 Традиционные портфельные теории**

Традиционные портфельные теории не имели в своей основе математического аппарата, позволяющего формально выразить в первую очередь риск, а также связь риска с доходностью.

Теория Доу-Джонса, в первую очередь, описывает тенденции движения цен на активы и их цикличность, а в плане управления портфелем только формирует сигналы на открытие или закрытие позиций.

Теория случайного блуждания и близкая ей теория эффективного рынка описывает движения цен на активы как совершенно случайные события и в рамках настоящего дискурса не предоставляет никаких аналитических

инструментов<sup>1</sup>, а в плане управления портфелем рекомендует инвестиции только в пассивные индексные фонды рынка.

Формальная теория в целом устанавливает правила (формулы) распределение активов между двумя видами портфелей, а также содержит набор правил по открытию и закрытию позиций. В плане управления риском, данная теория оперирует только на уровне минимизации убытков портфеля (то, что называется «наивным» пониманием риска и это именно то понятие, которое интуитивно ассоциируется с риском у большинства населения).

## 4.2 Современные теории

Только в 1952 г. Марковицем Г. [2] был предложен формальный подход к определению риска, показана функциональная зависимость доходности от риска и предложена формальная методика выбора портфеля по соотношению риск/доходность. Эта теория в настоящее время известна как «Современная портфельная теория», которая дала начало применению аналитических методов при оценке портфелей финансовых активов.

### 4.2.1 Модели минимизации дисперсии среднего значения

Современная портфельная теория утверждает, что каждый инвестор стремится реализовать две цели: максимизировать доходность и минимизировать неопределённость, связанную с фактом получения такой доходности. Таким образом, в момент принятия решения о приобретении той или иной ценной бумаги или их набора (далее в тексте, любое количество ценных бумаг будет называть портфелем), необходимо сбалансировать эти две цели, что и помогает сделать предложенная теория.

Основные положения теории заключаются в следующем:

В момент  $t_0$  инвестор не знает каким будет уровень доходности потенциальных портфелей<sup>2</sup>, поэтому ожидаемые доходности стоит считать

---

<sup>1</sup> Хотя теория случайного блуждания и упоминает риск, говоря о том, что невозможно постоянно получить доходность выше рыночной с учетом риска, но само понятие риска аналитически не определено

<sup>2</sup> За единственным исключением приобретения Treasury Inflation-Protected Securities.

случайной величиной. Случайная величина среди прочего имеет две характеристики – среднее значение (математическое ожидание) и стандартное отклонение. Поэтому, инвестору необходимо оценить ожидаемые доходности потенциальных портфелей, которые в подавляющем большинстве случаев выражаются через средние исторические доходности за определенный период владения и стандартные отклонения или риски, которые также в подавляющем большинстве случаев выражаются как стандартное отклонение исторических доходностей за определенный период владения. И после этого выбрать лучший портфель с точки зрения максимальной доходности при минимально приемлемом риске.

Стоит отметить, что данная теория неявно делает два предположения:

1. О ненасыщаемости –при одинаковом уровне неопределенности инвестор всегда предпочтет большую доходность
2. Об избегании риска –при одинаковой доходности инвестор всегда предпочтет минимальную неопределенность.

Расчет доходности портфеля представлен в формуле ( 1 ):

$$r_p = \sum_{i=1}^N X_i r_i, \quad (1),$$

где  $r_p$  – ожидаемая доходность портфеля;  
 $X_i$  – процентная доля  $i$ -й ценной бумаги в портфеле;  
 $r_i$  – ожидаемая доходность  $i$ -й ценной бумаги;  
 $N$  – количество ценных бумаг в портфеле.

Расчет риска портфеля представлен в формуле ( 2 ):

$$\sigma_p = \sqrt{\sum_{i=1}^N \sum_{j=1}^N X_i X_j \sigma_{ij}} \quad (2),$$

где  $X_i$  – процентная доля  $i$ -й ценной бумаги в портфеле;  
 $X_j$  – процентная доля  $j$ -й ценной бумаги в портфеле;  
 $\bar{\sigma}_{ij}$  – ковариация доходностей  $i$ -й и  $j$ -й ценных бумаг.

На основании данных формул возможен расчет т.н. достижимого множества, т.е. комбинации всех возможных портфелей на данном рынке. Пример такого множества представлен на Рисунок 5.

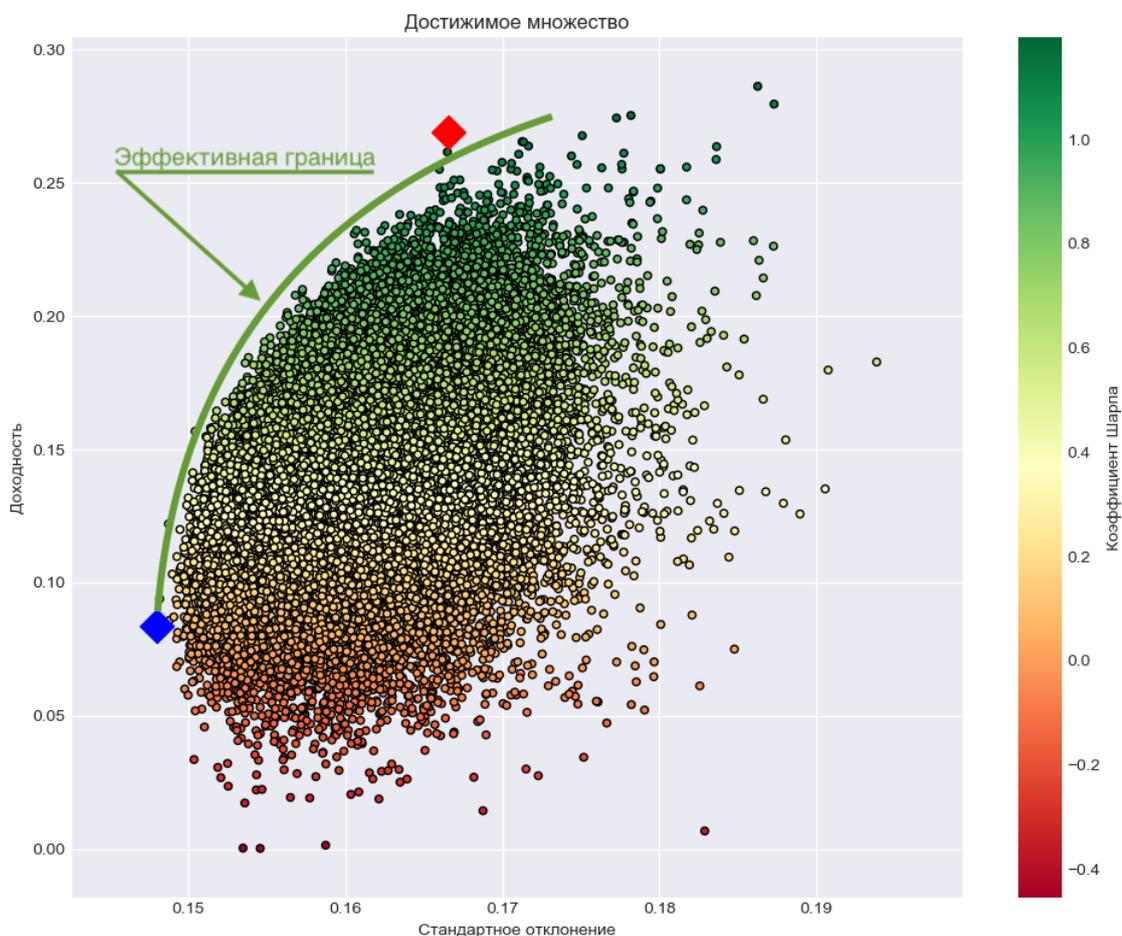


Рисунок 5 – Достижимое и эффективное множества

Однако инвестора должно интересовать только определенное подмножество портфелей из данного достижимого множества, а именно, множество портфелей, лежащих на левой границе достижимого множества портфелей, и известное как «эффективное множество». Это те портфели, которые обеспечивают максимальный уровень доходности при минимальном риске. Для того, чтобы подобрать оптимальный для себя портфель, инвестору необходимо построить кривые безразличия, характеризующие его/ее отношение к риску и выбрать портфель, находящийся в точке касания кривой безразличия эффективного множества.

Однако процесс построения кривых безразличия не тривиален, требует определения функции полезности для каждого индивидуального инвестора и затем определения степени принятия риска.

По этой причине более простой способ выбора портфеля был предложен У. Шарпом в 1966 году [3] и заключается в ранжировании портфелей по коэффициенту отношения доходности к риску, известному как коэффициент Шарпа.

Расчет данного коэффициента представлен на формуле ( 3 ):

$$SR = \frac{R_p - R_f}{\sigma_p} \quad ( 3 ),$$

где  $R_p$  – доходность портфеля;

$R_f$  – безрисковая ставка;

$\sigma_p$  – стандартное отклонение доходности портфеля.

Данная формула понимается следующим образом – вычисляемый коэффициент показывает сколько единиц рискованной доходности приходится на 1 единицу риска. Соответственно, инвестору следует выбирать портфель с наивысшим коэффициентом Шарпа. Другой выбор, который может сделать инвестор – это портфель с минимальным риском, т.е. с минимальным стандартным отклонением. На рисунке данные портфели помечены ромбами: синим для портфеля с минимальным риском и красным – для портфеля с максимальным коэффициентом Шарпа.

Стоит отметить, что исходя из рыночной модели<sup>3</sup>, риск портфеля состоит из двух компонентов, так же, как и риск отдельной ценной бумаги – это рыночный риск и собственный риск и может быть выражен формулой ( 4 ):

$$\sigma_p = \sqrt{\beta_{pI}^2 \sigma_I^2 + \sigma_{\varepsilon p}^2}, \quad ( 4 ),$$

где  $\beta_{pI}$  – коэффициент «бета», характеризующий степень зависимости доходности портфеля от доходности всего рынка, который в свою очередь является средневзвешенным коэффициентом бета всех ценных бумаг, находящихся в портфеле;

$\sigma_I^2$  – дисперсия доходности рынка;

---

<sup>3</sup> Данная модель предполагает, что доходность ценной бумаги неким образом связана с доходностью всего рынка, выраженного через какой-либо индекс.

$\sigma_{\varepsilon p}^2$  – дисперсия случайной погрешности доходности портфеля, которая в свою очередь является средневзвешенной случайной погрешностью доходностей ценных бумаг, входящих в портфель. Данный член уравнения «компенсирует» разницу между расчётной доходностью в соответствии с рыночной моделью и фактической.

Коэффициент «бета» портфеля в свою очередь рассчитывается согласно формуле ( 5 ):

$$\beta_{pI} = \sum_{i=1}^N X_i \frac{\sigma_{iI}}{\sigma_I^2} \quad ( 5 ),$$

где  $X_i$  – доля  $i$ -й ценной бумаги в портфеле;  
 $\sigma_{iI}$  – ковариация доходности  $i$ -й ценной бумаги в портфеле;  
 $\sigma_I$  – дисперсия доходности рынка.

Включение в портфель все большего количества ценных бумаг (или диверсификация) не оказывает значительного влияния на рыночный риск и ведет к его усреднению. Другими словами, это объясняется тем, что изменения на рынке в связи с теми или иными политическими или экономическими событиями приведут к соответствующему изменению цен всех ценных бумаг на рынке в той или иной мере.

Случайная погрешность портфеля рассчитывается согласно формуле ( 6 ):

$$\sigma_{\varepsilon p}^2 = \frac{1}{N} \left( \frac{\sigma_{\varepsilon 1}^2 + \sigma_{\varepsilon 2}^2 + \sigma_{\varepsilon N}^2}{N} \right) \quad ( 6 ),$$

где  $N$  – количество ценных бумаг в портфеле;  
 $\sigma_{\varepsilon i}^2$  – дисперсия доходности  $i$ -й ценной бумаги в портфеле.

Как следует из формулы ( 6 ), включение в портфель все большего количества ценных бумаг (или диверсификация) оказывает значительное влияние на собственный риск портфеля и ведет к его снижению. На интуитивном уровне это объясняется тем, что высокий риск одной ценной бумаги компенсируется низким риском другой ценной бумаги.

Влияние диверсификации портфеля на его риск обобщено на Рисунок 6.

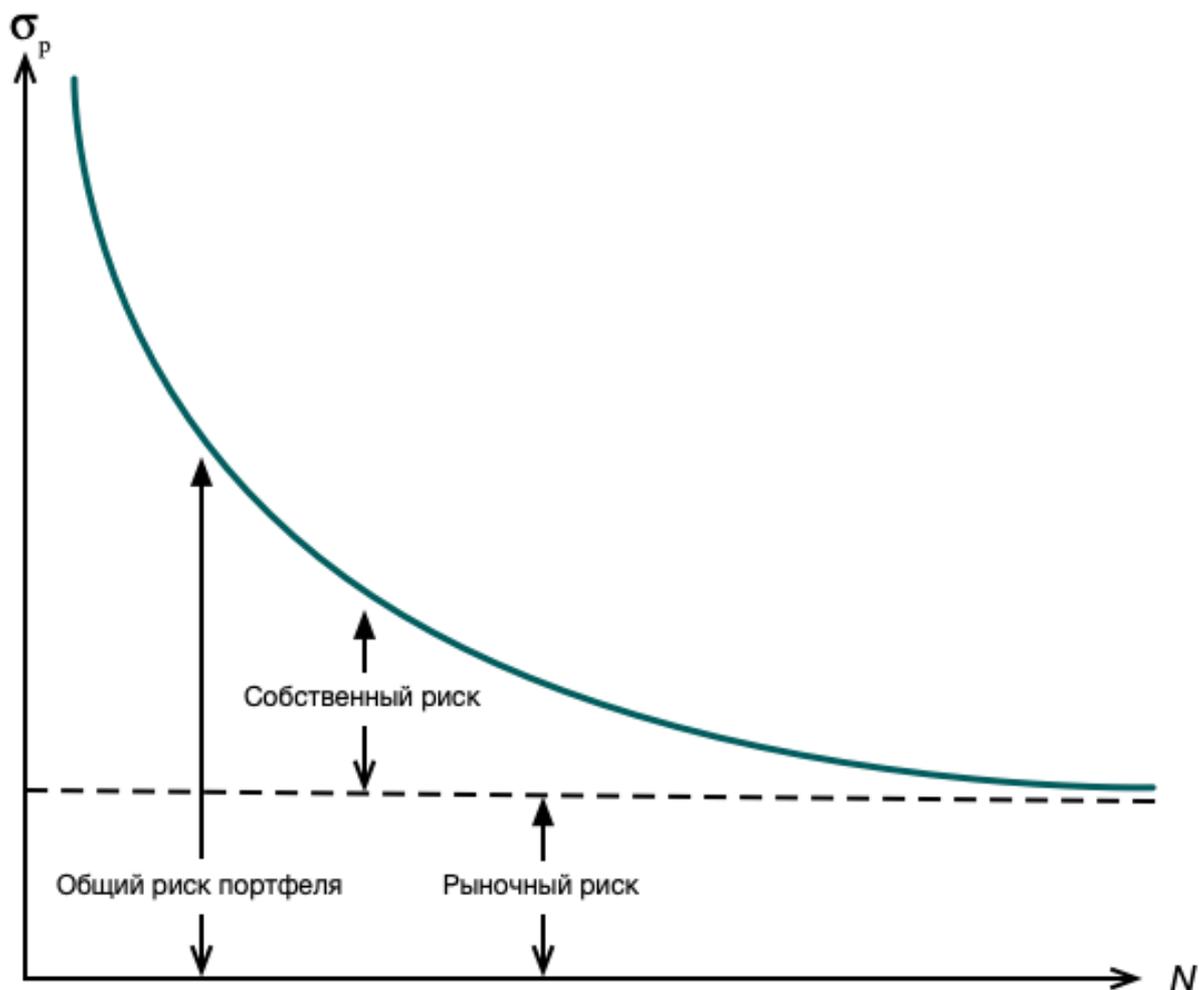


Рисунок 6 – Влияние диверсификации на риск

#### 4.2.2 Пост-современная портфельная теория

Данная теория, впервые упомянутая в прессе в 1993 году [4], и основанная на работах Х. Форси и Ф. Сортино [5] является скорее дополнением теории Марковица, а не совершенно новой теорией, призванной решить некоторые противоречия СПТ.

Основные ограничения СПТ состоят в том, что она подразумевает следующее:

1. Стандартное отклонение доходностей ценных бумаг портфеля является истинной мерой риска.
2. Доходности ценных бумаг портфеля могут быть адекватно представлены нормальным распределением.

СПТ использует такие меры риска и доходности, которые не всегда точно отражают объективную реальность. Так, использование нормального распределения подразумевает, что неопределенность получения дохода выше ожидаемого (или среднего, при использовании исторических доходностей) равна неопределенности получения дохода ниже ожидаемого. Также, СПТ оценивает ценные бумаги с доходностями, в которых количество положительных отклонений от среднего превышает количество отрицательных, как более рискованные чем они есть на самом деле. Обратное искажение характерно и для доходностей, где преобладают отрицательные доходности.

Сам Г. Марковиц отмечал [6], что в некоторых случаях СПТ не точно отражает поведение и предполагал, что модель с использованием полудисперсии будет более предпочтительной, но из-за сложности расчетов полудисперсии он все же взял за основу среднее и стандартное отклонение.

Также, в данной формуле вводится понятие «целевой доходности» для сравнения с фактической доходностью, что также более точно отражает объективную реальность.

Данной теорией предлагается мера риска, известная как «downside risk». Пример расчета данной меры риска представлен на формуле ( 7 ):

$$\sigma_{DR} = \sqrt{\int_{-\infty}^t (t - r)^2 f(r) dr} \quad (7),$$

где  $t$  – целевая требуемая доходность ценной бумаги или портфеля;  
 $r$  – средняя годовая доходность ценной бумаги и портфеля;  
 $f(r)$  – функция распределения доходности;

Данная формула учитывает вероятностно взвешенные квадраты доходностей ниже целевой. Возведение в квадрат экспоненциально «наказывает» «плохие» доходности, что более точно соответствует интуитивному пониманию риска [7].

Для ранжирования портфелей теория предлагает коэффициент аналогичный коэффициенту Шарпа – коэффициент Сортино [8] (см. формулу ( 8

)). Как уже отмечалась выше, коэффициент Шарпа, как и вся СПТ, исходят из предположения нормального распределения доходности. Соответственно, при рассмотрении доходностей, которые не отвечают критерию нормального распределения, а, например, скошены вправо (присутствуют положительные «сюрпризы», который инвестор будет оценивать как желаемые) коэффициент Шарпа будет оценивать как высокорискованные.

$$SrR = \frac{R_p - T_p}{\sigma_{DRp}} \quad (8),$$

где  $R_p$  – средняя годовая доходность ценной бумаги или портфеля;  
 $T_p$  – целевая требуемая доходность ценной бумаги и портфеля;  
 $\sigma_{DRp}$  – «Downside Risk» портфеля.

В остальном процедура выбора портфеля соответствует подходу, предложенному в СПТ – риск портфеля либо с минимальной значением значения «downside risk» или с максимальным значением коэффициента Сортино.

### 4.2.3 Бюджетирование риска

Кроме вышеперечисленных проблем, характерных для СПТ, другой проблемой, отмеченной в работе Р. Мертона [9] и [10], является высокая чувствительность модели к входным параметрам и, в первую очередь, к параметру ожидаемой доходности. Так, было показано, что при изменении доходности и риска на одинаковую величину, влияние доходности на состав портфеля оказывается в 10 раз более сильным, а ведь именно оценки доходности являются наиболее сложным процессом инвестиционной деятельности и характеризуются максимальной неопределенностью. Параметр риска в этом смысле гораздо более стабилен.

Поэтому были предложены методы оптимизации портфеля, которые бы не учитывали показатель доходности, а оперировали только риском. Построенные в ходе применения этих методов портфели получили название «портфели паритета риска».

Базовая идея метода состоит в том, чтобы подобрать активы в портфель таким образом, чтобы величина вклада каждого актива в общий риск портфеля была одинакова.

Можно использовать различные меры риска, например VaR, но «классический» подход – это использование стандартного отклонения исторических доходностей.

Определив стандартное отклонение портфеля по историческим данным, возможно рассчитать вклад каждого актива в общий риск портфеля:

$$RC_i = w_i \frac{\sum_{j=1}^N w_j Cov(X_i X_j)}{\sigma_p} \quad (9),$$

где  $w_i$  – вес  $i$ -й ценной бумаги в портфеле;  
 $N$  – количество ценных бумаг в портфеле;  
 $Cov(X_i X_j)$  – коэффициент ковариации доходностей  $i$ -й и  $j$ -й ценной бумаги;  
 $\sigma_p$  – стандартное отклонение доходности портфеля.

Данная формула показывает взвешенный коэффициент чувствительности риска актива к риску всего портфеля.

Получив данный коэффициент чувствительности для всех активов портфеля, необходимо их уравнивать так, чтобы выполнить нижеприведенное равенство:

$$RC_i = \frac{1}{N} \sigma_p \quad (10),$$

где  $N$  – количество ценных бумаг в портфеле;  
 $\sigma_p$  – стандартное отклонение доходности портфеля.

Аналитическое решение данной задачи существует только для случая диагональной матрицы дисперсии активов (т.е. при условии, что доходности активов не коррелированы между собой). Для всех остальных случаев необходимо использовать численные методы.

В настоящее время предложено большое количество алгоритмов на разных языках программирования (R, Python, Matlab) в которых применяются различные подходы к решению данной задачи – метод Ньютона, циклический покоординатный спуск и др.

В общем, можно сказать, что данный подход является частным случаем метода бюджетирования риска, в котором производится явное распределение риска в соответствии с профилем риска той или иной ценной бумаги. Другими словами, коэффициенты вклада риска каждой ценной бумаги в портфель не подразумеваются равными, а задаются явно и задача сводится к определению весов на основании уже имеющихся коэффициентов вклада риска.

#### **4.2.4 Байесовские методы**

Байесовская статистика – это теория в области статистики, основанная на байесовской интерпретации вероятности, где вероятность отражает степень доверия событию, которая может измениться, когда новая информация будет собрана, в отличие от фиксированного значения, основанного на частотном подходе. Степень доверия может основываться на априорных знаниях о событии, таких как результаты предыдущих экспериментов или личное доверие событию. Самым известной моделью, использующей данные методы для решения проблемы оптимизации портфеля ценных бумаг, является модель «Блэка-Литтермана».

##### **4.2.4.1 Модель Блэка Литтермана**

Данная модель была разработана Ф. Блэком и Р. Литтерманом во время их работы в банке «Голдман Сакс» и впервые опубликована в 1992 году [11]. Данная модель совмещает априорные оценки доходности с апостериорными оценками ожидаемых доходностей, но данная модель не только позволяет инвестору учесть свой прогноз относительно ожидаемых доходности той или иной ценной бумаги как в абсолютных значениях, так и относительных, но и степень уверенности в том или ином прогнозе. Расчет оптимального портфеля выполняется в несколько шагов.

На первом этапе необходимо рассчитать апостериорные значения. Оценка начинается с расчета рыночного риска согласно формуле ( 11 ):

$$\lambda = \frac{R - R_f}{\sigma^2} \quad ( 11 ),$$

где  $R$  – рыночная, равновесная доходность;  
 $R_f$  – безрисковая ставка;  
 $\sigma^2$  – дисперсия рынка.

Затем выполняется расчет рыночной или равновесной доходности портфеля согласно формуле ( 12 ):

$$\Pi = \lambda \Sigma w_{mkt} \quad ( 12 ),$$

где  $\Pi_{N \times 1}$  – вектор равновесных доходностей, где  $N$  – количество ценных бумаг в портфеле;  
 $\lambda$  – рыночный риск;  
 $\Sigma_{N \times N}$  – ковариационная матрица доходностей активов в портфеле;  
 $w_{mkt}$  – удельный вес актива в портфеле на основании его капитализации;

Если у инвестора нет никакого мнения относительно будущей доходности активов в портфеле, то вычисления прекращаются на данном этапе и, в таком случае модель Блэка-Литтермана, по сути, рекомендует держать рыночно нейтральный портфель ценных бумаг.

На втором этапе необходимо рассчитать апостериорные значения. Если у инвестора имеется мнение касательно будущих доходностей активов, то их необходимо выразить либо в абсолютных, либо в относительных значениях. Например, «Доходность акций компании А составит 10%» или «Акции компании В покажут доходность на 5% выше, чем акции компании С». Мнение инвестора формализуется с помощью трех матриц:

1. Матрица «Р» – отражает факт наличия или отсутствия мнения инвестора и тип мнения – абсолютный или относительный (Рисунок 7).

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix} \text{ или } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0,8 & -0,2 \end{bmatrix}$$

Рисунок 7 – Пример матрицы мнений

2. Матрица «Q» – отражает мнение инвестора относительно ожидаемой доходности активов (Рисунок 8).

$$\begin{bmatrix} 10 \\ 5 \end{bmatrix}$$

Рисунок 8 – Пример матрицы предполагаемых доходностей

3. Матрица «Ω» – матрица уверенности инвестора во мнении относительно ожидаемой доходности. В данной матрице на этапе расчётов

используются не непосредственные значения уверенности, а дисперсия ошибок (отличие предполагаемой доходности от фактической) в виде ковариационной матрицы ошибок. Существует несколько методов расчета данной матрицы, классическим, предложенный создателями модели, является использование априорной дисперсий (это можно объяснить так, что актив, который характеризовался высокой дисперсией имеет высокую степень неопределенности прогноза). Рассчитывается данная матрица согласно формуле ( 13 ):

$$\Omega = \text{diag}(P(\tau\Sigma)P^T) \quad ( 13 ),$$

где  $P$  – матрица мнений инвестора о доходности активов;  
 $\tau$  – весовой коэффициент, отражающий степень неуверенности во мнении;  
 $\Sigma_{N \times N}$  – ковариационная матрица доходностей активов в портфеле;

Такой подход учитывает и неуверенность мнения инвестора касательно доходности актива и априорную подразумеваемую неопределенность, выраженную дисперсией исторических доходностей.

При этом необходимо рассчитать коэффициент  $\tau$ , который характеризует степень неуверенности инвестора во мнении. Одним из подходов является использование формулы ( 14 ):

$$\tau = \frac{1}{T} \quad ( 14 ),$$

где  $T$  – количество наблюдений в выборке;

Либо можно использовать заранее определенное значение равное 0,05.

После этого можно рассчитать новый комбинированный вектор доходности, представленный формулой ( 15 ):

$$E_r = [(\tau\Sigma)^{-1} + P^T\Omega^{-1}P]^{-1}[(\tau\Sigma)^{-1}\Pi + P^T\Omega^{-1}Q] \quad ( 15 ),$$

где  $\tau$  – весовой коэффициент, отражающий степень неуверенности во мнении;  
 $\Sigma_{N \times N}$  – ковариационная матрица доходностей активов в портфеле;  
 $P_{K \times N}$  – матрица, отражающая факт наличие или отсутствие мнения инвестора и тип мнения, где  $K$  – количество прогнозов и  $N$  – количество ценных бумаг в портфеле;  
 $Q_{K \times 1}$  – матрица, отражает мнение инвестора относительно ожидаемой доходности активов, где  $K$  – количество прогнозов;

$\Omega_{K \times K}$  – матрица, характеризующая уверенности инвестора во мнении относительно ожидаемой доходности актива, где  $K$  – количество прогнозов;

$P_{N \times 1}$  – вектор равновесных доходностей, где  $N$  – количество ценных бумаг в портфеле;

Данная формула рассчитывает средневзвешенную сумму вектора подразумеваемой доходности актива и вектора прогнозной доходности, где весами выступают подразумеваемая неопределённость доходностей, неопределённость мнений и степени неопределенности мнений.

Получив ожидаемый вектор доходностей, его необходимо подставить в формулу ( 12 ) и решить для каждого компонента вектора, получив тем самым веса активов в целевом портфеле.

### **4.3 Выводы по главе**

В ходе выполнения работы, был определен подход к классификации портфельных теорий, применяющихся на практике в современном портфельном управлении, а именно на основе математического аппарата, походов и методов, лежащих в основе той или иной портфельной теории. В рамках каждой математической модели была рассмотрена наиболее популярная теория ее представляющая. В следующих главах работы будут рассмотрены вопросы создания компьютерных алгоритмов, реализующих каждую теорию для решения практической задачи составления эффективного портфеля ценных бумаг.

## 5. Описание используемых технологий

### 5.1 Серверная часть

#### 5.1.1 Язык программирования Python

Для написания серверной части веб-приложения, а также для манипуляции данными выбран язык программирования Python, который в настоящее время является самым популярным языком для обработки данных (Рисунок 9) [12].

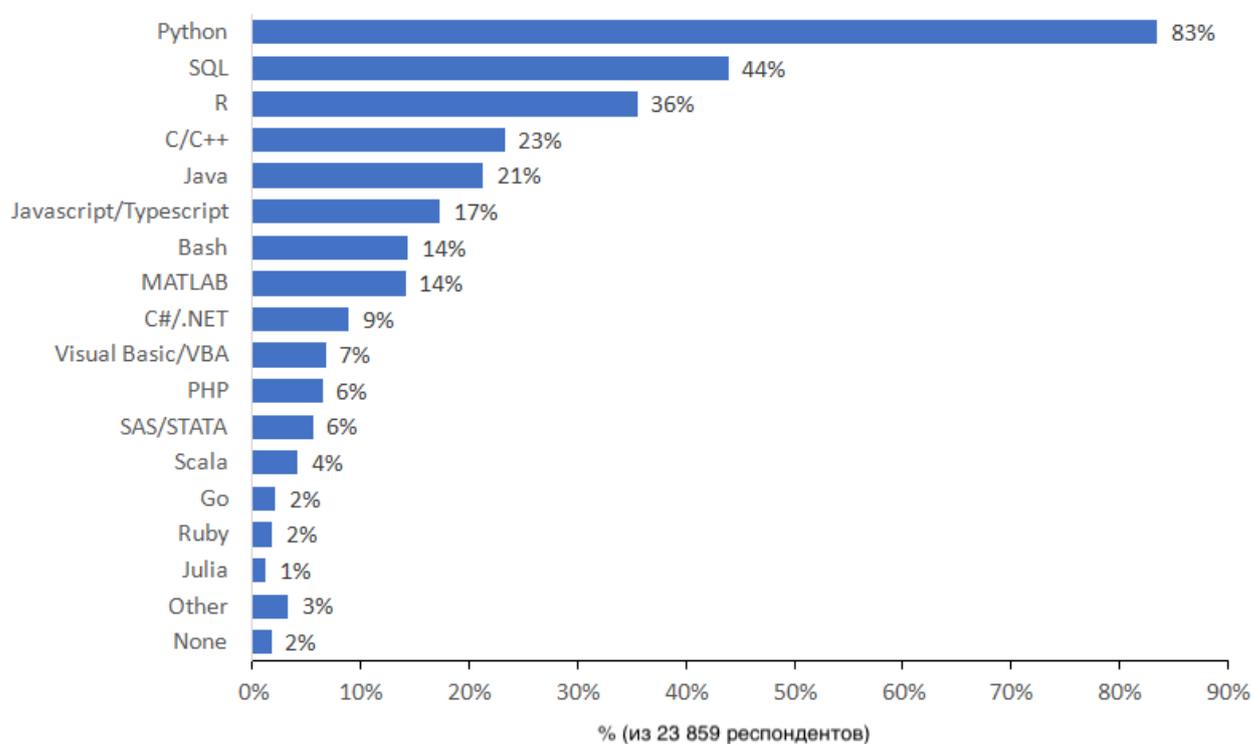


Рисунок 9 – Наиболее популярные языки программирования для обработки данных

#### 5.1.2 Библиотеки

Язык программирования Python является языком высокого уровня и, следовательно, поддерживает все математические операции – операции интегрального и дифференциального исчисления, статистического анализа, линейной алгебры и т.д. Однако данная особенность является и недостатком данного языка, т.к. все операции выполняются в виртуальной машине Python, где происходит перекомпиляция файлов в байт-код и их выполнение. Но для высоконагруженных приложений чрезвычайно важно добиться максимальной

производительности, что, в рамках настоящего обсуждения, подразумевает выполнение вычислений как можно «ближе к железу», т.е. в машинном коде. Для достижения наивысшей производительности математические вычисления будут выполняться с помощью сторонних библиотек.

### **5.1.2.1 Фреймворк Django**

Django – веб-фреймворк, написанный на языке программирования Python, предназначенный для создания веб-сайтов и веб-приложений.

Архитектура Django реализована на основе классической модели MVC, с добавлением компонента «Т» – template, ответственного за разметку представления в виде веб-страницы.

Данный фреймворк применяется с целью быстрого прототипирования веб-сайтов, т.к. включает большое количество готовых компонентов веб-приложений, абстрагирует целые функциональные блоки, а также поддерживает достаточно мощный ORM, который абстрагирует язык запросов SQL, позволяя писать запросы на языке программирования Python.

### **5.1.2.2 Numpy**

Numpy – библиотека с открытым исходным кодом для Python, является реализацией спецификаций BLAS и LAPACK для задач линейной алгебры, поддерживает JIT-компиляция в машинный код. Она оптимизирована для работы с многомерными массивами и поддерживает высокоуровневые математические функции, предназначенные для работы с такими массивами. Часть библиотеки, ответственная за обработку данных, написана на компилируемом языке C, что значительно повышает скорость выполнения вычислений.

### **5.1.2.3 Pandas**

Pandas – это программная библиотека на языке Python для обработки и анализа данных, которая выступает своего рода высокоуровневыми API для библиотеки Numpy, являющейся инструментом более низкого уровня. Pandas

оперирует специальными структурами данных, называемыми «Датафреймы», которые состоят из рядов данных, в том числе и временных, и предоставляет в распоряжение разработчику большой набор операций из области статистического анализа, реализованных в виде вызываемых функций, что позволяет абстрагировать математический аппарат и сосредоточиться на анализе данных.

#### **5.1.2.4 Matplotlib**

Matplotlib – это библиотека на языке программирования Python для визуализации данных двумерной и трехмерной графикой. Получаемые изображения могут быть сохранены в любой файл изображения, поддерживаемый операционной системой. Matplotlib является гибким, легко конфигурируемым пакетом, который вместе с Numpy, SciPy и IPython предоставляет возможности, подобные «MATLAB».

#### **5.1.2.5 Dash**

Dash – это библиотека, разработанная компанией Plotly, для создания интерактивных веб-приложений с помощью следующего стека технологий:

- Серверная часть – Plotly, который в свою очередь написан на Python;
- Клиентская часть – ReactJS.

В результате пользователь получает полностью интерактивное веб-приложение для визуализации данных, которое поддерживает следующие виды взаимодействия:

- взаимодействие с графиком;
- взаимодействие с элементом графика;
- динамическая связь данные – пользователь;
- обновление графика в режиме реального времени без перезагрузки страницы / перерисовки графика.

Библиотека позволяет строить графики по данным, поступающим в режиме реального времени с приборов, сенсоров и т. д. Данные должны быть записаны в текстовый файл любого формата, и затем при любом изменении

файла, данные на графике обновляются без перезагрузки страницы. Частота обновления ограничена лишь вычислительными ресурсами компьютера.

Также, библиотека поддерживает каскадные страницы стилей, что позволяет создавать удобные для восприятия дашборды, которые значительно облегчают проведение разведочного анализа данных.

#### **5.1.2.6 Redis**

Redis – система управления базами данных типа NoSQL с открытым исходным кодом, которая работает со структурами данных типа «ключ – значение». Ориентирована на достижение максимальной производительности на атомарных операциях, т.к. все данные располагаются в оперативной памяти вычислительной машины. Используется как для баз данных, так и для реализации кэшей, брокеров сообщений.

В данном случае система используется как кэш.

#### **5.1.2.7 Celery**

Python долгое время оставался синхронным языком программирования, не позволяющим выполнять задачи в асинхронном режиме. Для решения локальных бизнес задач это не представляло больших проблем, т.к. данный факт принимался как одно из условий использования данного языка программирования и преимущества его использования оказывались выше данного недостатка. Однако при использовании Python для веб-разработки, данный недостаток становится гораздо более существенным т.к. программная блокировка взаимодействия пользователя с продуктом до момента окончания продолжительного вычислительного процесса, является одним из самых серьезным факторов, наиболее сильно снижающих степень удовлетворенности пользователя продуктом.

Библиотека Celery позволяет организовывать асинхронное выполнение функций языка Python, представляя собой систему управления очередями задач, которая может брать задачи из очереди и выполнять задачи по расписанию. В качестве брокера очередей обычно используются «RabbitMQ» или «Redis». В

очереди помещаются задачи, которые потом извлекаются и выполняются исполнительными узлами Celery.

Это значительно повышает UX продукта, т.к. клиентская часть никогда не теряет «отзывчивости» даже если результат вычисления будет доступен через несколько минут.

### 5.1.2.8 Dask

Важным является вопрос масштабирования ресурсов. Горизонтальное масштабирование является гораздо более гибким инструментом как экономической, так и с практической точки зрения, позволяя динамически вводить и выводить дополнительные вычислительные ресурсы в зависимости от нагрузки и требуемого объема вычислений. Физически, ввод и вывод ресурсов будет выполняться автобалансировщиком нагрузки, а распределение вычислений по узлам и контроль будет осуществляться с помощью библиотеки Dask.

Dask – это библиотека с открытым исходным кодом, написанная на языке Python, для параллельных вычислений и может может распараллеливать фреймы данных Pandas и структуры данных Numpy. Dask можно запускать либо на локальном компьютере, либо в кластере. Таким образом, код пишется всего один раз, а затем необходимо только выбрать – запускать код на локальном компьютере, либо развертывать в кластере из множества узлов, используя для всего этого самый обычный синтаксис Python.

В Dask предлагается три параллельные коллекции, в которых можно хранить объем данных, превышающий по размеру оперативной памяти: Dataframes, Arrays и Bags. В каждом из этих типов коллекций можно хранить данные, сегментировав их между RAM и жестким диском, а также распределять данные по множеству узлов в кластере.

- Dask DataFrame состоит из уменьшенных датафреймов, таких, как в Pandas, поэтому позволяет использовать подмножество возможностей из синтаксиса запросов Pandas.

- Структура данных Dask Arrays поддерживает массивы в стиле NumPy и подмножество операций NumPy над массивами.

- В Dask Bag можно хранить и обрабатывать различные коллекции объектов Python, не уместяющихся в памяти.

Преимущества использования Dask:

- Нет необходимости изучать другие языки программирования и фреймворки, т.к. Dask использует Python.

- Нет необходимости трансформировать данные в другие форматы, т.к. Dask нативно поддерживает основные виды представления данных в Python, используемые для математических расчетов – массивы и датафреймы.

- Простота в установке и настройке – для установки используются менеджеры пакетов «pip» или «conda», при этом минимальная рабочая конфигурация настраивается одним аргументом команды, запускающей узел кластера в работу.

- Возможность разворачивать узел кластера на практически любом компьютере, поддерживающем Python версии 3.

### **5.1.3 Базы данных**

#### **5.1.3.1 PostgreSQL**

PostgreSQL – бесплатно распространяемая объектно-реляционная система управления базами данных. Данная СУБД имеет встроенные высокопроизводительные и надёжные механизмы транзакций и репликации, нативно поддерживает работу с языком программирования Python, имеет встроенную поддержку слабоструктурированных данных в формате JSON с возможностью их индексации

Данная СУБД используется для организации хранения всех пользовательских данных.

### **5.1.3.2 TimescaleDB**

TimescaleDB – специализированная система управления базами данных для хранения данных временных рядов или темпоральная база данных, основанная на PostgreSQL. Полностью поддерживает стандарт SQL, в то же время сочетает в себе качества реляционных и NoSQL баз данных. Благодаря тому, что TimescaleDB основана на PostgreSQL, она может быть интегрирована с основными программами экосистемы PostgreSQL: Grafana, Prometheus, Hibernate, Kafka и т. д.

## **5.2 Контроль версий**

Для контроля версий выбрана система Git.

Данная система нативно поддерживается всеми операционными системами, базирующимися на ядре Unix, такими как OSX и Linux. В качестве веб-сервиса для хостинга кода выбран сервис GitHub, который представляет широкие возможности для применения практик «Непрерывная интеграция» и «Непрерывное развертывание».

## **5.3 Клиентская часть**

Интерфейс веб-приложения реализован с использованием следующего стека фронтэнд технологий – HTML, CSS и Java Script.

### **5.3.1 Dash**

Библиотека Dash используется как на серверной стороне приложения, так и на клиентской. Серверная часть приложения отвечает за создания графиков, а клиентская часть – за рендеринг графиков в браузере клиента. Клиентская часть приложения реализована с использованием фреймворка ReactJS и реализует возможность интерактивного взаимодействия с графиком в браузере.

## **5.4 Выводы по главе**

В результате обзора существующих языков программирования и возможностей ими предоставляемых для написания серверной части продукта

была выбрана экосистема, базирующаяся на языке программирования высокого уровня Python.

Для манипуляции данными используется библиотеки Pandas и Numpy.

Для вывода графической информации на экран используются библиотеки Dash и Matplotlib&

С целью распараллеливания вычислений используется библиотека Dask.

Для обеспечения асинхронности вычислений используется библиотека Celery.

Для написания клиентской части продукта был выбран традиционный набор средств HTML/CSS/JavaScript.

## 6. Инфраструктура веб-приложения

Данное приложение будет реализовано в соответствии с парадигмой SaaS, необходимо уделить внимание планированию инфраструктуры приложения для обеспечения бесперебойной работы сервиса, а также предусмотреть возможности как вертикального, так и горизонтального масштабирования при резком увеличении нагрузки на сервис.

Также, важной задачей является задача оптимизации выполнения математических расчетов, которые не слишком сложны с математической точки зрения (т.к., по сути, задача сводится к задаче оптимизации по  $N$  параметрам), однако оперируют большими объемами данных. Решение данной задачи также достигается за счет возможности масштабирования ресурсов.

Принципиальная схема коммуникации между компонентами приложения и внешними системами представлена на Рисунок 10 и в Таблица 1.

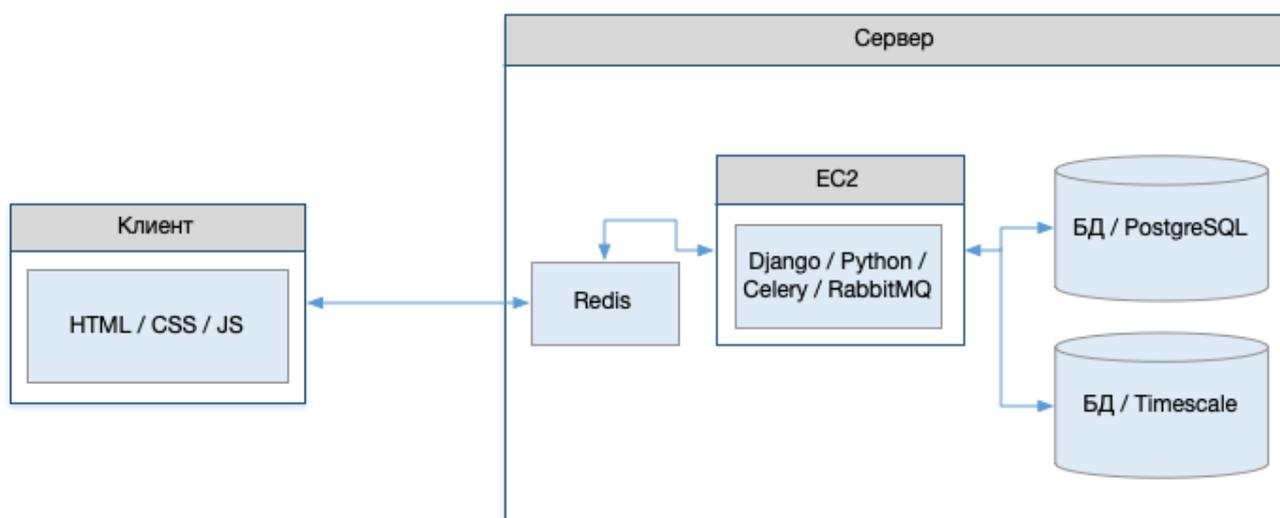


Рисунок 10 – Принципиальная схема коммуникации

Таблица 1. Описание элементов схемы коммуникации

Элемент схемы	Описание
Серверная часть	
База данных	База данных для хранения данных о пользователях и их портфелях ценных бумаг.
База данных для временных рядов	База данных, оптимизированная для хранения временных рядов – данных о ценах закрытия ценных бумаг.
EC2	Выделенная виртуальная машина, содержащий настроенный веб сервер и бизнес-логику приложения.
Redis	Выделенная виртуальная машина с сервисом кэширования.

Элемент схемы	Описание
Клиентская часть	
Клиент	Клиентская часть приложения для визуализации информации.

## 6.1 Вычислительные ресурсы

Принципиальная схема физической реализации инфраструктуры приложения показана на Рисунок 11 и в Таблица 2.

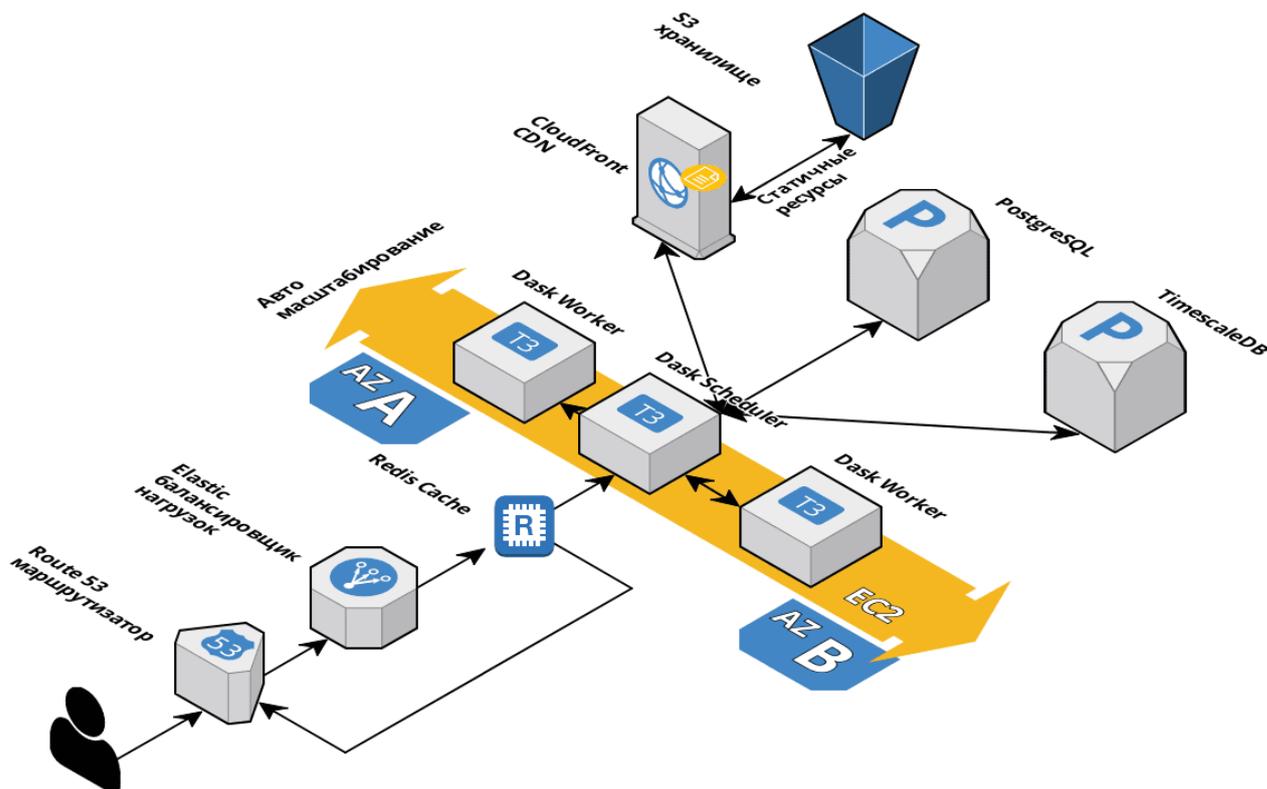


Рисунок 11 – Принципиальная схема физической инфраструктуры

Таблица 2. Описание элементов инфраструктуры

Элемент схемы	Описание
Маршрутизатор	Маршрутизатор AWS Route 53 для маршрутизации запросов и ответов по кратчайшему маршруту между сервером и пользователем.
Балансировщик нагрузок	Балансировщик нагрузок AWS Elastic Load Balancer для автоматического запуска и уничтожения VM в зависимости от объема поступающих запросов.
Redis Cache	Виртуальная машина с системой кэширования данных.
VM AWS EC2	Виртуальная машина с заказанными вычислительными мощностями с поддержкой функции автоматического масштабирования для запуска и уничтожения VM в зависимости от объема исполняемых вычислений.
Сеть доставки контента	Сеть доставки контента AWS CloudFront для кэширования и загрузки неизменяемых файлов.
Хранилище S3	Хранилище AWS S3 для хранения неизменяемых пользовательских и системных файлов.

Элемент схемы	Описание
PostgreSQL RDS	СУБД PostgreSQL RDS для хранения данных о пользователях и их портфелях.
TimescaleDB	База данных, оптимизированная для хранения данных временных рядов – данных об итогах торгов ценными бумагами.

Такая конфигурация позволит максимально оптимизировать работу приложения в случае высоких нагрузок, как в терминах количества запросов, так и в терминах объема исполняемых вычислений.

### 6.1.1 Инфраструктура как код

При заказе, развертывании и настройке облачных вычислительных ресурсов применяется подход «Инфраструктура как код».

Данный подход обладает многими преимуществами, как-то:

1. Скорость развертывания – файл конфигураций описывает заказываемые ресурсы формальным языком со своим синтаксисом и грамматикой, при этом многие задачи администрирования серверов абстрагированы и описываются одной или двумя командами. Таким образом, описание процесса развертывания и настройки вычислительных ресурсов сокращается до нескольких часов.

2. Консистентность – достаточно написать файл конфигураций один раз и затем дублирование либо перенос вычислительных ресурсов сокращается от дней и часов до нескольких минут, при этом исключаются ошибки неправильной конфигурации новых ресурсов.

3. Масштабируемость – данный подход позволяет легко масштабировать вычислительные ресурсы, начиная от количества серверов и заканчивая настройкой балансировщиков нагрузок. При этом многие действия уже абстрагированы в шаблонах того или иного инструмента.

4. Истинность – т.к. код имеет декларативный стиль, то текущая версия кода всегда однозначно отражает текущее состояние инфраструктуры.

Таким образом, применение данного подхода позволит значительно сэкономить время и повысить эффективность управления вычислительными ресурсами. С этой целью будет использоваться несколько инструментов.

#### **6.1.1.1 Terraform**

Terraform – открытое программное обеспечение компании HashiCorp, позволяющее разработчику описывать и разворачивать вычислительную инфраструктуру в облачных сервисах с помощью высокоуровневого языка конфигурации.

Это декларативный язык конфигураций, таким образом достаточно написать конфигурацию, в которой будет изложено, какую инфраструктуру и с какими характеристиками необходимо развернуть. Такая конфигурация создается в человеко-читаемом текстовом формате. Terraform работает напрямую с поставщиком облачных ресурсов через его API и приводит инфраструктуру в соответствие с конфигурацией, указанной в файле.

#### **6.1.1.2 Ansible**

Ansible – система управления конфигурациями виртуальных машин в целом и операционных систем в частности, написанная на языке программирования Python, с использованием декларативного языка разметки для описания данных конфигураций. Используется для автоматизации настройки и развертывания программного обеспечения.

Преимущества применения данного инструмента:

- Безагентность – т.е. нет необходимости предварительно устанавливать какое-либо ПО на удаленном сервере.
- Идемпотентность – вне зависимости от того, сколько раз вызывается операция, результат не меняется.

## **6.2 Выводы по главе**

В результате планирования инфраструктуры для развертывания веб-приложения принято решение использовать подход «Инфраструктура как код»,

что не только ускоряет заказ и настройку вычислительных ресурсов, но и является масштабируемым решением, позволяя в считанные минуты увеличивать или уменьшать количество заказанных вычислительных ресурсов, что вместе с почасовой оплатой за их использование, позволяет существенно сократить затраты на ее содержание.

В качестве облачного провайдера был выбран сервис AWS, что однако не принципиально т.к. подход «Инфраструктура как код» позволяет выполнять перенос инфраструктуры за считанные минуты.

## 7. Реализация

В качестве интегрированной среды разработки использовалась профессиональная версия IDE PyCharm компании JetBrains, которая предоставляется бесплатно всем студентам аккредитованных учебных заведений.

### 7.1 Серверная часть

Серверная часть приложения отвечает за логику обработки пользовательских запросов и их роутинг, формирование представлений, взаимодействие с БД и обработку данных.

#### 7.1.1 Сервер

Серверная часть приложения написана на языке Python с использованием фреймворка Django. В Django та или иная единица функциональности веб-приложения реализована в виде класса, который может быть наследован, а атрибуты которого могут быть переопределены. Вообще, наследование классов используется очень широко, целые блоки стандартной функциональности веб-приложения (например, проверка факта авторизации пользователя при попытке получить доступ к ограниченному контенту) уже реализован в «стандартной» библиотеке и может быть наследована при написании представления. Поддерживается множественное наследование, что позволяет создавать достаточно сложные представления с минимальным количеством кода.

#### 7.1.2 Обработка данных

Обработка данных реализована на языке программирования Python с использованием библиотек Pandas, Numpy и Scipy. В основном используется библиотека Pandas, предоставляющая наиболее высокий уровень абстракции для выполнения математических расчётов. На Рисунок 12 представлен расчет эксцесса серии данных о доходности ценной бумаги с использованием функций Pandas.

```
def get_kurtosis(returns_series, ddof=0):
```

```

standard_deviation = returns_series.std(ddof=ddof)
mean_of_series = ((returns_series-returns_series.mean())**4).mean()
return mean_of_series/standard_deviation**4

```

Рисунок 12 – Расчет эксцесса серии данных

Высокий уровень абстракции предполагает выполнение кода «далеко» от железа, что сказывается на скорости вычислений. Поэтому, при выполнении некоторых операций над массивами данных, используется библиотека NumPy, которая хоть и не обладает столь богатым функционалом манипуляции данных как Pandas, но значительно выигрывает в скорости, т.к. многие операции, в частности итерирование реализованы на языке C. На Рисунок 13 представлен пример расчета волатильности портфеля ценных бумаг с использованием функции умножения массивов данных.

```

def get_portfolio_volatility(weights, covariance_matrix):
    return np.dot(np.dot(weights.T, covariance_matrix), weights)**0.5

```

Рисунок 13 – Расчет волатильности портфеля ценных бумаг

Для решения задач, связанных с нахождением минимума функций, используется библиотека Scipy, в которой реализовано большое количество различных численных методов.

При реализации программного продукта использовался алгоритм последовательного квадратичного программирования методом наименьших квадратов, который является одним из наиболее распространённых и эффективных оптимизационных алгоритмов общего назначения.

На Рисунок 14 представлен фрагмент кода с использованием данного алгоритма для нахождения эффективного множества.

```

results = minimize(
    get_portfolio_volatility, initial_guess,
    args=(covariance_matrix,),
    method="SLSQP",
    options={"disp": False},
    constraints=(return_equals_target, weights_sum_to_one),
    bounds=weight_constraints
)
return results.x

```

Рисунок 14 – Алгоритм SLSQP

### 7.1.3 Распараллеливание вычислений

Распараллеливание вычислений организовано с помощью библиотеки Dask. После установки данной библиотеки, на центральном узле запускается планировщик задач, а на всех остальных узлах – исполнители. Планировщик задач отвечает за распределение вычислений по узлам, сбор результатов и дальнейшую диспетчеризацию результата в соответствии с поступившим запросом.

В библиотеке Dask реализован большой (но не весь) функционал Pandas и NumPy, поэтому при выполнении вычислений, если используемый метод поддерживается библиотекой, его можно вызвать через метод Dask, тем самым распараллелив данное вычисление. На Рисунок 15 приведен фрагмент кода по распараллеливанию процесса вычисления корреляционной матрицы.

```
import dask.dataframe as dd
from dask.distributed import Client
client = Client('tcp://192.168.1.102:8786')

dask_df = dd.from_pandas(portfolio_df, npartitions=20, sort=False)
dask_corr = dask_df.corr().compute()
```

Рисунок 15 – Распараллеливание вычислений в Dask

## 7.2 Клиентская часть

Клиентской частью являются сформированные сервером представления, которые отправляются в ответ на запрос клиента. Клиентская часть написана с использованием стандартного для таких случаев стека технологий:

- Язык разметки – HTML;
- Язык стилей – CSS;
- Язык скриптов – JavaScript.

Также используется язык шаблонизатора Django для внедрения в страницу динамического контента, поступающего либо из базы данных, либо полученного в результате манипуляций над данными.

### **7.3 Выводы по главе**

В результате выполнения работы было разработано и написано программное обеспечение в соответствии с клиент – серверной архитектурой, шаблоном MVCT и принципами ООП.

Также написаны модули обработки данных, выполняющих вычисления требуемых показателей. Используются методы распараллеливания вычислений для их ускорения.

## 8. Тестирование

Тестирование программного продукта – процесс изучения работы данного программного продукта с целью получения информации о его качестве. Качество программного продукта, в свою очередь, характеризует насколько полно он отвечает требованиям заинтересованных сторон – в первую очередь заказчика, но также и подрядчика как в лице компании, так и в лице ее отделов и специалистов – дизайнеров, программистов, маркетологов и т.д. Каждый из участников может иметь различное представление о продукте, его качествах и об идеальном состоянии продукта. Таким образом, постановка задачи обеспечения качества продукта подразумевает под собой определение заинтересованных лиц, их критериев качества и затем нахождения оптимального решения, удовлетворяющего этим критериям. Тестирование является одним из наиболее устоявшихся способов обеспечения качества разработки программного обеспечения и входит в набор эффективных средств современной системы обеспечения качества программного продукта.

С технической точки зрения тестирование заключается в выполнении приложения на некотором множестве исходных данных и сверке получаемых результатов с заранее известными эталонными данными с целью установить соответствие различных свойств и характеристик приложения заказанным свойствам. Ведь если программа не содержит синтаксических ошибок, прошла трансляцию и может быть выполнена на компьютере, она обязательно вычисляет какую-либо функцию, осуществляющую отображение входных данных в выходные. Это означает, что компьютер на своих ресурсах доопределяет частично определенную программой функцию до тотальной определенности. Следовательно, судить о правильности или неправильности результатов выполнения программы можно, только сравнивая спецификацию желаемой функции с результатами ее вычисления, что и осуществляется в процессе тестирования.

Стоит отметить, что тестирование – это самая дорогая стадия разработки программного продукта для заказчика с точки зрения стоимости устранения ошибок в коде. Так, согласно [13], в среднем в США затраты на исправление одной ошибки в коде на этапе тестирования составляют более 7 000 долларов, по сравнению с 1000 долларов на этапе реализации (см. Рисунок 16). Однако затраты на этапе эксплуатации возрастают в два раза и превышают 14 000 долларов. При этом, если затраты подрядчика подсчитать достаточно легко – это могут быть штрафные санкции за срыв сроков поставки продукта, рост трудозатрат на поиск и устранение ошибок в коде, то затраты заказчика на этапе эксплуатации, в частности, упущенную выгоду, подсчитать гораздо сложнее, а оценить точно – невозможно. Именно поэтому в 2000-х годах наблюдался значительный рост интереса как со стороны научного сообщества, так и со стороны бизнеса к проблематике тестирования программного обеспечения.



Рисунок 16 – Стоимость устранения ошибки в коде на стадиях разработки

К настоящему времени разрозненные методы и подходы к тестированию ПО сформировались в отдельную отрасль научного знания – QA или Quality Assurance (англ. Обеспечение Качества). Данная отрасль выработала свою

методологию, методы и инструменты, а также теоретическую основу, с целью повышения качества тестирования, снижения затрат на него и как можно более полного соответствия постоянно меняющимся бизнес-требованиям и контекстам.

Для решения задачи тестирования веб-приложения, выбран метод тестирования по уровням детализации:

1. Модульное тестирование.
2. Интеграционное тестирование.
3. Системное тестирование.

## **8.1 Модульное тестирование**

### **8.1.1 Определение**

Модульное тестирование – тестирование минимальных блоков кода, которые выполняют атомарные функции. В терминах ООП – это тестирование программных функций, методов и атрибутов класса и т.д.

### **8.1.2 Задачи**

Основной задачей модульного тестирования является обеспечение устойчивого роста масштаба программного продукта. Хотя написание тестов требует определенных трудозатрат, и на начальном этапе проекта это может быть не столь очевидным, но с ростом его сложности, модификация проекта без наличия качественных модульных тестов будет становиться все более и более сложной (см. Рисунок 17) [14] и в конечном итоге – невозможной из-за наличия эффекта энтропии кода (его неупорядоченности), которая увеличивается с каждым новым изменением.

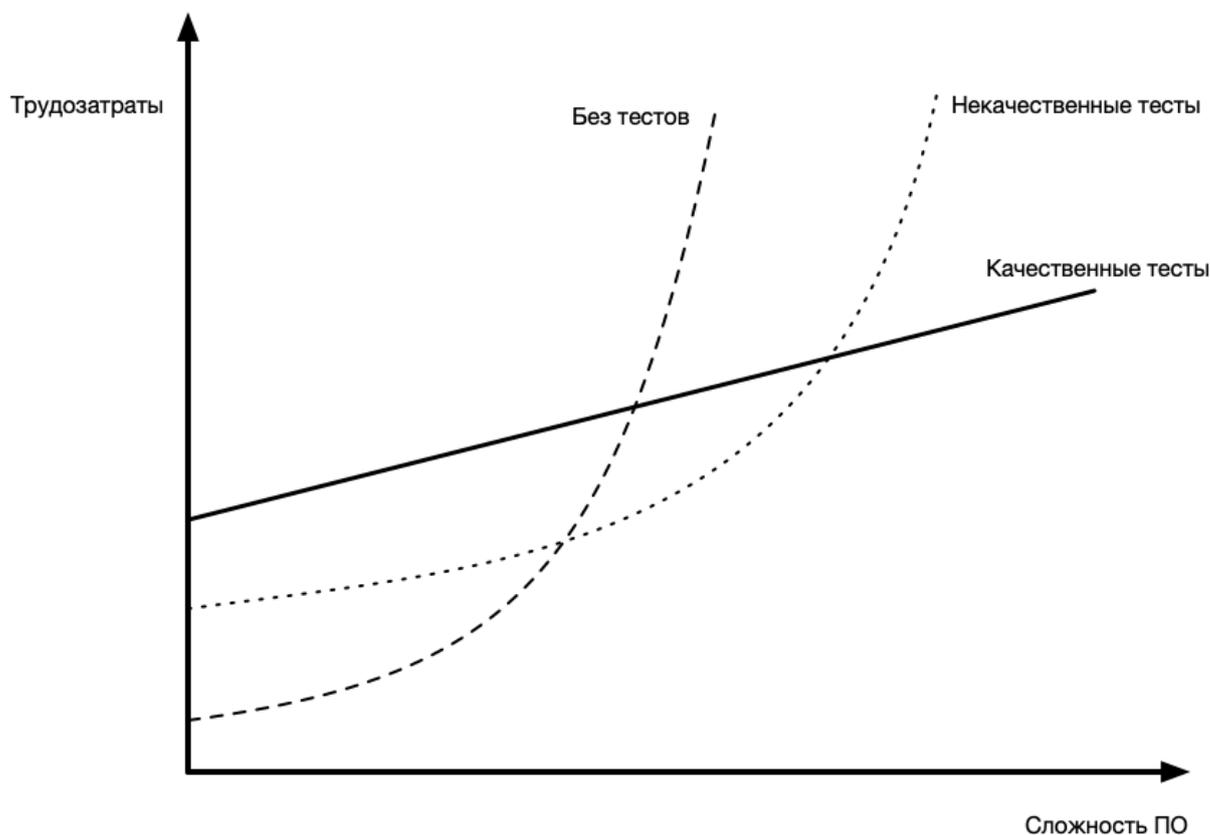


Рисунок 17 – Зависимость трудозатрат от наличия тестов с ростом сложности

### 8.1.3 Оценка тестов

Модульные тесты обычно оцениваются по коэффициенту покрытия. Коэффициент покрытия – это отношение количества строк кода, исполненного тестом к общему количеству строк кода в данном программном продукте. Обычно стараются достигнуть 100% коэффициента покрытия, однако стоит помнить, что хотя данный коэффициент является сильным отрицательным индикатором – т.е. низкий коэффициент покрытия явно говорит о том, что качество кода низко, но в то же время коэффициент покрытия является слабым положительным индикатором – т.е. даже 100% коэффициент покрытия ничего не говорит о высоком качестве кода.

### 8.1.4 Тестирование

Фреймворк «Джанго» обеспечивает полную поддержку тестирования приложения на всех трех уровнях детализации. Для этого используются стандартная библиотека языка Python – Unittest и ее подклассы SimpleTestCase и TestCase.

Для расчета процента покрытия кода модульными тестами используется сторонняя библиотек «Coverage».

Пример модульного теста с использованием библиотеки `unittest.TestCase` приведен на Рисунок 18.

```
class HomePageViewTest(TestCase):
    def test_view_url_exists_at_proper_location(self):
        resp = self.client.get('/')
        self.assertEqual(resp.status_code, 200)
    def test_view_url_by_name(self):
        resp = self.client.get(reverse('home'))
        self.assertEqual(resp.status_code, 200)
    def test_view_uses_correct_template(self):
        resp = self.client.get(reverse('home'))
        self.assertEqual(resp.status_code, 200)
        self.assertTemplateUsed(resp, 'home.html')
```

Рисунок 18 – Пример модульного теста

Данный модульный тест проверяет следующее:

1. Наличие домашней страницы по требуемому адресу.
2. Псевдоним страницы «Домашняя страница», используемый в программном коде, открывает на домашнюю страницу.
3. Домашняя страница использует требуемый шаблон представления.

Результаты работы теста представлен на Рисунок 19.

```
coverage run --source='.' manage.py test pages
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...
```

-----  
Ran 3 tests in 0.094s

OK

```
Destroying test database for alias 'default'...
```

Рисунок 19 – Пример работы модульного теста

Библиотека «Coverage» выводит отчет о коэффициенте покрытия кода тестами. Пример отчета представлен на Рисунок 20.

```
coverage report
Name                               Stmts  Miss  Cover
-----
manage.py                           12      2   83%
med/__init__.py                      0      0  100%
med/asgi.py                           4      4    0%
med/settings.py                       24      0  100%
med/urls.py                            3      0  100%
med/wsgi.py                            4      4    0%
pages/__init__.py                     0      0  100%
pages/admin.py                         1      0  100%
```

pages/apps.py	3	3	0%
pages/migrations/__init__.py	0	0	100%
pages/models.py	1	0	100%
pages/tests.py	14	0	100%
pages/urls.py	3	0	100%
pages/views.py	6	0	100%
users/__init__.py	0	0	100%
users/admin.py	10	0	100%
users/apps.py	3	3	0%
users/forms.py	11	0	100%
users/migrations/0001_initial.py	8	0	100%
users/migrations/__init__.py	0	0	100%
users/models.py	7	0	100%
users/tests.py	1	1	0%
users/views.py	1	1	0%
-----			
TOTAL	116	18	84%

Рисунок 20 – Пример отчета о покрытии кода тестами

## 8.2 Интеграционное тестирование

### 8.2.1 Задачи

Результатом тестирования и верификации отдельных модулей, составляющих программную систему, должно быть заключение о том, что эти модули являются внутренне непротиворечивыми и соответствуют требованиям. Однако отдельные модули редко функционируют сами по себе, поэтому следующая задача после тестирования отдельных модулей – тестирование корректности взаимодействия нескольких модулей, объединенных в единое целое. Такое тестирование называют интеграционным. Интеграционные тесты предназначены для верификации взаимодействия модулей системы друг с другом, а также системы с вне-процессными зависимостями. В первую очередь, это базы данных, файловые системы, почтовые сервера и так далее.

Необходимо помнить, что интеграционные тесты гораздо более высокочатратны как с точки зрения их написания и поддержки, так и с точки зрения из использования – они требуют гораздо больше времени и задействуют гораздо больше ресурсов по сравнению с модульными тестами. Поэтому рекомендуется покрывать как можно больше критических сценариев с помощью модульных тестов и покрывать основной бизнес-сценарий с помощью интеграционных тестов [14]. Таким образом, это позволяет максимально сократить трудозатраты и в то же время убедиться, что система работоспособна.

В результате проведения интеграционного тестирования и устранения всех выявленных дефектов получается согласованная и целостная архитектура программной системы, т.е. можно считать, что интеграционное тестирование – это тестирование архитектуры и низкоуровневых функциональных требований.

## 8.2.2 Тестирование

Фреймворк «Джанго» обеспечивает полную поддержку тестирования приложения на всех трех уровнях детализации. Для этого используются стандартная библиотека языка Python – unittest и ее подкласс TestCase, который поддерживает интеграционные тесты. Дополнительно, используется библиотека «Factory Boy», которая позволяет генерировать динамические данные, отвечающие различным требованиям – уникальности, соответствия региональным форматам данных (например, телефонные номера) и т.д., а также автоматизирует процесс создания и уничтожения тестовой базы данных.

Пример интеграционного теста с использованием библиотек unittest.TestCase и Factory Boy приведен на Рисунок 21 и Рисунок 22.

```
class UserFactory(factory.Factory):
    class Meta:
        model = CustomUser

class UserTestCase(TestCase):
    def setUp(self):
        self.test_user = UserFactory.create(
            passport='12345t',
            username='test_user',
            email='test_user@gmail.com',
            first_name='test_first',
            last_name='test_last',
            type='staff')

    def test_user_created(self):
        self.assertEqual(self.test_user.passport, '12345t')
        self.assertEqual(self.test_user.username, 'test_user')
        self.assertEqual(self.test_user.email, 'test_user@gmail.com')
        self.assertEqual(self.test_user.first_name, 'test_first')
        self.assertEqual(self.test_user.last_name, 'test_last')
        self.assertEqual(self.test_user.type, 'staff')
        self.assertNotEqual(self.test_user.type, 'patient')
```

Рисунок 21 – Пример интеграционного теста

```
Creating test database for alias 'default'...
```

```
System check identified no issues (0 silenced).
```

```
.
```

```
-----  
Ran 1 test in 0.015s
```

```
OK
```

```
Destroying test database for alias 'default'...
```

Рисунок 22 – Результаты интеграционного теста

## 8.3 Системное тестирование

### 8.3.1 Задачи

После завершения интеграционного тестирования все модули системы являются функционально согласованными. Начиная с этого момента можно переходить к тестированию системы в целом как единого объекта тестирования – к системному тестированию. Как правило, для системного тестирования применяется подход черного ящика, при этом в качестве входных и выходных данных используются реальные данные, с которыми работает система, или данные, подобные им.

### 8.3.2 Тестирование

Для выполнения и автоматизации системного тестирования применяется фреймворк «Селениум», предназначенный, в частности, для автоматического тестирования интерфейсов веб-приложений. Пример тестов представлен на Рисунок 23 и Рисунок 24.

```
class CustomerListTestCase(LiveServerTestCase):  
    def setUp(self):  
        self.client.login(username="alant", password="alan123")  
        self.selenium = webdriver.Firefox()  
        self.selenium.maximize_window()  
        self.selenium.get('http://127.0.0.1:8000/')  
        super(CustomerListTestCase, self).setUp()  
  
    def tearDown(self):  
        self.selenium.quit()  
        super(CustomerListTestCase, self).tearDown()  
  
    def test_succesful_login(self, username="alant", password="alan123",  
next_url=None):  
        self.selenium.get('http://127.0.0.1:8000/users/login/')  
        username_el = self.selenium.find_element_by_id("id_username")  
        username_el.send_keys(username)  
        password_el = self.selenium.find_element_by_id("id_password")
```

```

password_el.send_keys(password)
submit = self.selenium.find_element_by_id('loginButton')
submit.send_keys(Keys.RETURN)
time.sleep(3)
assert 'Здравствуйте, Alan Turing.' in self.selenium.page_source

def test_unsuccesful_login(self, username="alant", password="alan1234",
next_url=None):
self.selenium.get('http://127.0.0.1:8000/users/login/')
username_el = self.selenium.find_element_by_id("id_username")
username_el.send_keys(username)
password_el = self.selenium.find_element_by_id("id_password")
password_el.send_keys(password)
submit = self.selenium.find_element_by_id('loginButton')
submit.send_keys(Keys.RETURN)
time.sleep(3)
assert 'Please enter a correct username and password.' in
self.selenium.page_source

```

Рисунок 23 – Пример системного тестирования - тестирование интерфейса

```

Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..

```

-----

Ran 2 tests in 25.114s

OK

```

Destroying test database for alias 'default'...

```

Рисунок 24 – Результат системного тестирования

## 8.4 Непрерывная интеграция

Непрерывная интеграция (англ. Continuous Integration, CI) – это практика разработки программного обеспечения, которая заключается в выполнении частых автоматизированных тестирований и сборок проекта в различных средах выполнения для скорейшего выявления и решения интеграционных проблем. Обычно сборка проекта осуществляется после внесения разработчиком изменения в код, либо через определенные промежутки времени.

К преимуществам использования CI можно отнести то, что разработчик в любой момент времени имеет достоверную информацию о состоянии исходных кодов в системе. Это важно при командной разработке проекта с использованием центрального репозитория. В случае если последняя сборка совершилась с ошибкой, то, соответственно, не имеет смысла получать актуальные исходные файлы из репозитория, т.к. они не работают. В этом случае происходит оповещение разработчиков об ошибке, что позволяет сразу приступить к ее

поиску и устранению. Также часто частью процесса сборки является расчет различных метрик качества кода, как-то: покрытие тестами, статический анализ, поиск дублированного кода и так далее.

Обычно процесс CI состоит из следующих шагов:

1. Срабатывание триггера. В подавляющем большинстве случаев триггером является изменение в системе контроля версий, но также это может быть изменение в файловой системе, определенный момент времени, сборка другого проекта.

2. Обновление файлов с исходным кодом из репозитория. На данном этапе CI сервер делает обновление своей локальной копии исходного кода проекта. В процессе выясняются изменения в коде, произошедшие с последней интеграции. Выяснение изменений необходимо для того, чтобы в случае сбоя можно было легко выяснить причину.

3. Анализ кода. После получения файлов проекта из системы контроля версий можно провести статический анализ кода. Существует множество автоматических средств, для различных языков программирования, позволяющих провести такой анализ. Обычно измеряются следующие характеристики кода: наличие типичных ошибок, статические характеристики кода (сложность, размер и т. д.), соответствие принятым стандартам кодирования и другое. Данный этап является необязательным для процесса непрерывной интеграции, но в случае его наличия можно получить дополнительные преимущества от введения практики в виде метрик по коду. Данный этап подразумевает не только получение статических характеристик кода, но и их включение в отчеты, создаваемые сервером интеграции.

4. Выполнение тестирования – модульного, интеграционного, системного и так далее.

5. Сборка – это может быть компиляция файлов, создание инсталлятора, сборка Докер-контейнера.

6. Генерация отчетов.

Среди обозначенных этапов можно выделить критические, ошибка на которых означает сбой сборки – это, в первую очередь, этап тестирования. Иногда к критическим также относят этап статического анализа. Например, сборка считается проваленной, если оформление кода не соответствует принятому стандарту.

### **8.4.1 Настройки процесса CI**

Для реализации процесса CI используется программный продукт TravisCI, поставляемый по модели SaaS.

TravisCI – распределённый веб-сервис для сборки и тестирования программного обеспечения, использующий GitHub в качестве хостинга исходного кода. Веб-сервис поддерживает сборку проектов на множестве языков, включая C, C++, D, JavaScript, Java, PHP, Python и Ruby. Разные проекты с открытым исходным кодом используют Travis CI для непрерывной интеграции кода, например Ruby, Ruby on Rails, Node.js.

В качестве триггера тестирования и сборки проекта используется изменение в одной из отслеживаемых сервисом веток на хостинге GitHub.

Диаграмма состояний кода приложения в процессе CI представлена на Рисунок 25.

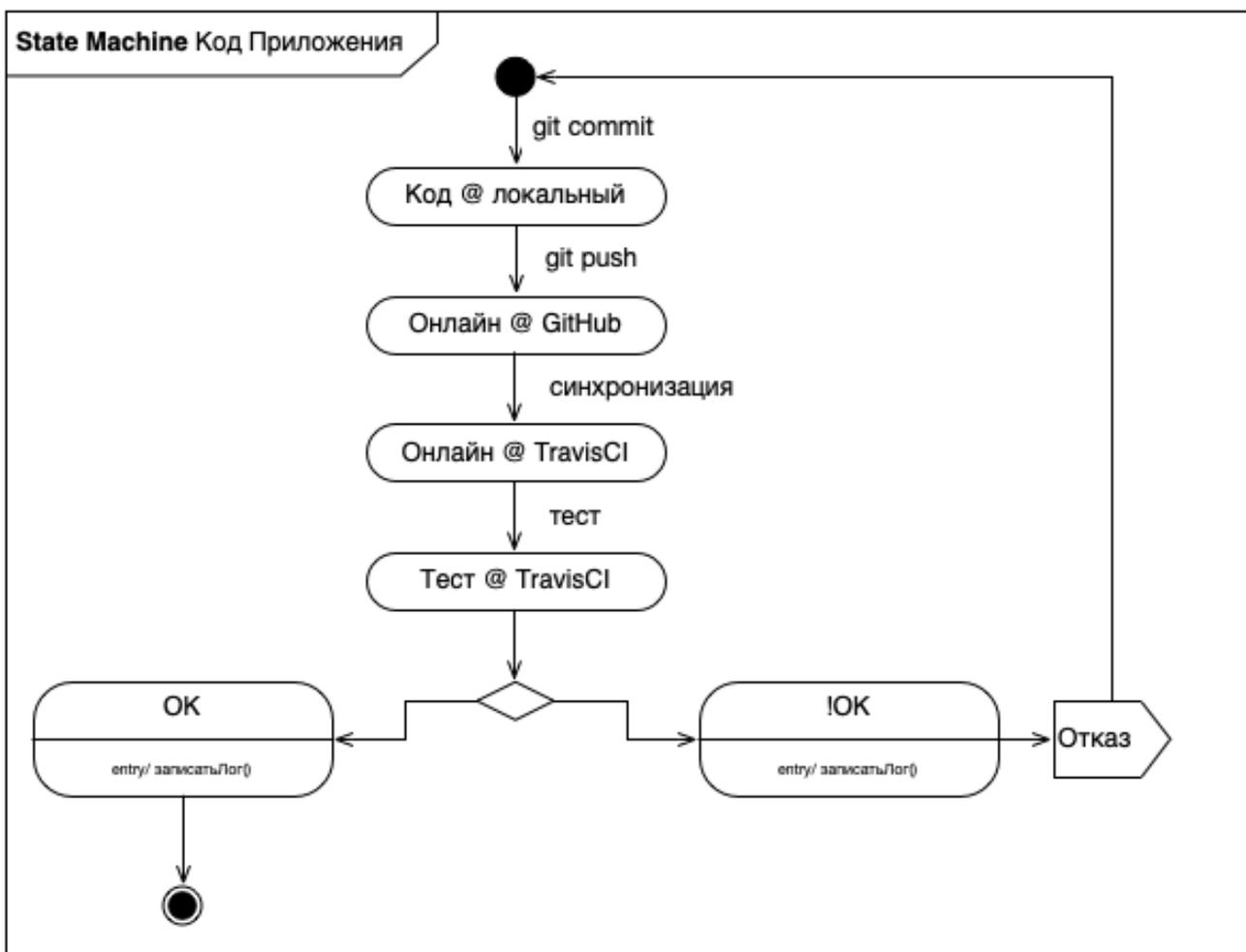


Рисунок 25 – Диаграмма состояний кода приложения в процессе CI

Сервис TravisCI поддерживает тестирования в различных версиях используемых фреймворков, языков программирования, сред исполнения, операционных систем и так далее. Все требуемые настройки указывается в конфигурационном файле, размещаемом в корневой папке приложения. Стоит обратить внимание, что сервис поддерживает не только модульное и интеграционное тестирование, но и тестирование интерфейса с использованием фреймворка Selenium и браузера Firefox работающего в «headless» режиме, т.е. без устройства графического вывода.

После выполнения тестирования и сборки проекта на странице сервиса отображается состояние проекта. Если состояние тестирования и сборки ошибочное (см. Рисунок 26), то в окне вывода лога тестирования и сборки можно выяснить причину ошибки (см. Рисунок 27).

**✘ vkr Add failing test**

- Commit cd89cf0
- Compare d5e9f35...cd89cf0
- Branch vkr
- Ruslan

---

Python: 3.9  
AMD64

Рисунок 26 – Сообщение об ошибке процесса CI

```

283
284 tests/test_risk.py::TestRisk::test_get_conditional_historic_var FAILED [ 16%]
285 tests/test_risk.py::TestRisk::test_get_drawdown PASSED [ 33%]
286 tests/test_risk.py::TestRisk::test_get_gaussian_var PASSED [ 50%]
287 tests/test_risk.py::TestRisk::test_get_historic_var PASSED [ 66%]
288 tests/test_risk.py::TestRisk::test_get_maximum_drawdown_and_date PASSED [ 83%]
289 tests/test_risk.py::TestRisk::test_get_semideviation PASSED [100%]
290
291 ===== FAILURES =====
292 _____ TestRisk.test_get_conditional_historic_var _____
293
294 self = <tests.test_risk.TestRisk testMethod=test_get_conditional_historic_var>
295
296 def test_get_conditional_historic_var(self):
297     self.assertAlmostEqual(0.036550, risk.get_conditional_historic_var(self.hfiportfolios['Convertible Arbitrage']),
298                             places=5)
299 > self.assertAlmostEqual(0.9054841264, risk.get_conditional_historic_var(self.hfiportfolios['CTA Global']), places=5)
300 E AssertionError: 0.9054841264 != 0.04126428571428569 within 5 places (0.8642198406857143 difference)
301
302 /home/travis/build/ruslaniv/vltl/tests/test_risk.py:54: AssertionError
303 ===== short test summary info =====
304 FAILED tests/test_risk.py::TestRisk::test_get_conditional_historic_var - Asse...
305 ===== 1 failed, 5 passed in 0.72s =====
306 The command "pytest -v" exited with 1.
307
308
309 Done. Your build exited with 1.

```

Рисунок 27 – Лог процесса CI

В случае отсутствия ошибок в процессе тестирования и сборки, выводится сообщение об успешном завершении процесса (см. Рисунок 28) и сопутствующий лог.

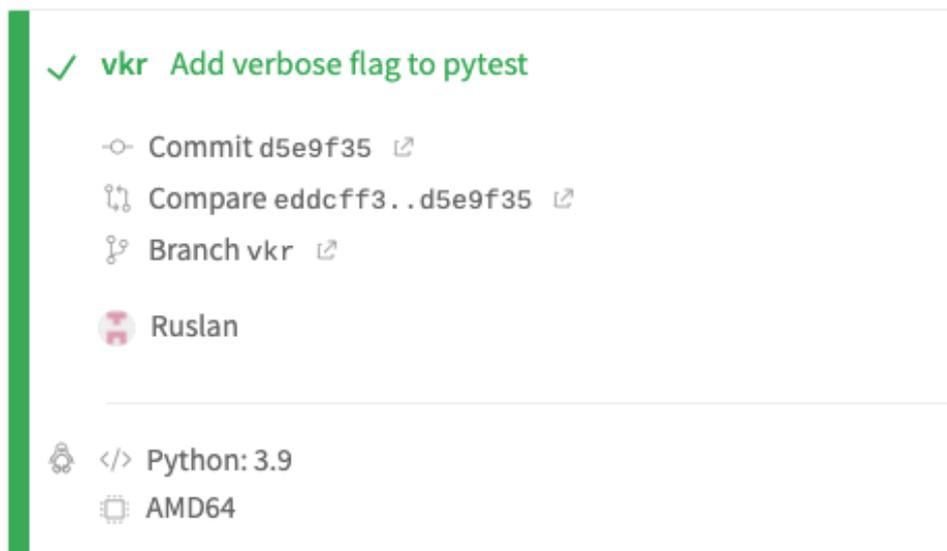


Рисунок 28 – Сообщение об успешном завершении процесса CI

## 8.5 Выводы по главе

В результате выполнения работы по данной главе, были разработаны модульные, интеграционные и системные тесты и выполнено тестирование веб-приложения по всем трем уровням детализации.

Настроен сервис непрерывной интеграции Travis CI и выполнена настройка репозитория кода веб-приложения для использования данного сервиса, который обеспечивает тестирование веб-приложения при любом изменении в кодовой базе.

## 9. Развертывание инфраструктуры

Развёртывание программного обеспечения – это все действия, которые делают программную систему готовой к использованию. Данный процесс является частью жизненного цикла программного обеспечения [15].

На Рисунок 29 представлена диаграмма развертывания на физической инфраструктуре, которая в графическом виде отображает взаимосвязь кода, инфраструктуры и их соединений. Диаграмма развертывания объединяет компоненты, артефакты и узлы для определения физической архитектуры системы.

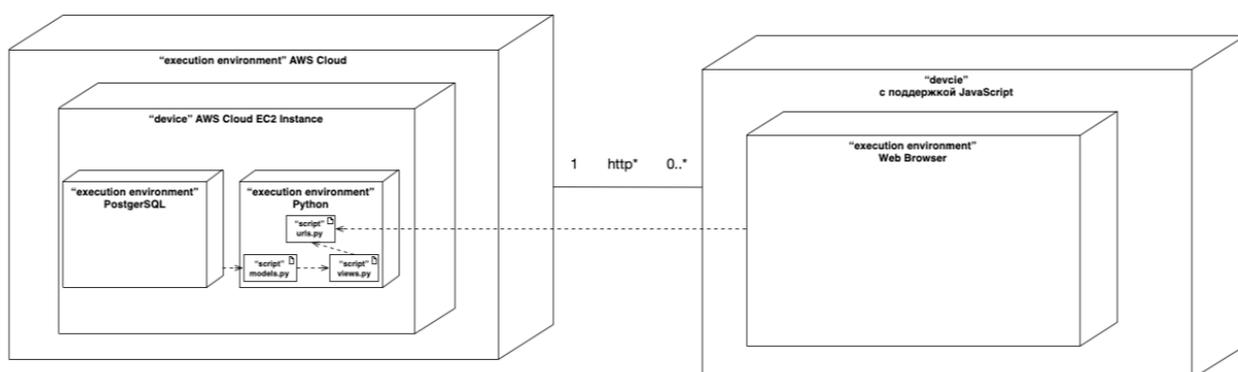


Рисунок 29 – Диаграмма развертывания

### 9.1 Заказ вычислительных ресурсов

Заказ вычислительных ресурсов осуществляется на мощностях облачного провайдера Amazon Web Services, но благодаря использованию языка обеспечения служб Terraform, это не существенно. Идентичную конфигурацию можно развернуть на любом другом облачном ресурсе, поддерживающим библиотеку Terraform.

На Рисунок 30 представлен пример файла, описываемый требуемую конфигурацию ресурсов.

```
resource "aws_autoscaling_group" "http" {
  name                = "vltl-autoscaling-group"
  max_size            = var.autoscaling_http["max_size"]
  min_size            = var.autoscaling_http["min_size"]
  desired_capacity    = var.autoscaling_http["desired_capacity"]
  health_check_type  = "ELB"
  default_cooldown   = "30"
  health_check_grace_period = "120"
```

```

force_delete           = true
launch_configuration   = aws_launch_configuration.http.name
load_balancers         = [aws_elb.http.name]
termination_policies  = ["OldestLaunchConfiguration"]
vpc_zone_identifier    = [aws_subnet.http.id]
}
resource "aws_launch_configuration" "http" {
  name_prefix   = "http"
  image_id      = data.aws_ami.centos.id
  instance_type = "t2.small"
  key_name      = aws_key_pair.user_key.key_name
  security_groups = [
    aws_security_group.administration.id,
    aws_security_group.web.id,
  ]
  user_data = file("scripts/first-boot-http.sh")
}

```

Рисунок 30 – Заказ вычислительных ресурсов с помощью Terraform

## 9.2 Конфигурирование системы

Конфигурация системы – как то установка требуемых пакетов, настройка брандмауэра, создание пользователей, пользовательских групп и разграничение прав доступа осуществляются с помощью библиотеки Ansible и языка описания конфигураций им поддерживаемым.

На Рисунок 31 представлен фрагмент кода по настройке ОС – создание пользовательских групп, настройка брандмауэра и установка пакетов, требуемых для работы программного продукта.

```

- name: create a non-root group
  group:
    name: "{{ deploy_group }}"
    state: present

- name: create non-root user
  user:
    name: "{{ deploy_user }}"
    group: "{{ deploy_group }}"
    shell: "/bin/bash"
    state: present

- name: add authorized_key to non-root user
  authorized_key:
    user: "{{ deploy_user }}"
    state: present
    key: "{{ lookup('file', ssh_dir + ssh_key_name + '.pub') }}"

- name: add non-root group to sudo privileges
  lineinfile:
    dest: /etc/sudoers
    state: present
    regexp: "^{{ deploy_group }}"

```

```

    line: "%{{ deploy_group }} ALL=(ALL) NOPASSWD: ALL"
    validate: visudo -cf %s

- name: ensure Python packages are installed
  dnf:
    name: [ "Python3-pip", "Python3-devel", "fail2ban", "ufw", "net-tools" ]
    update_cache: yes
  become: yes

```

Рисунок 31 – Конфигурация системы с помощью Ansible

## 9.3 Настройка внешних модулей

К внешним модулям относятся модули, не определенные в коде разрабатываемого продукта и, следовательно, требующие отдельной установки и настройки. В данном случае – это СУБД PostgreSQL и кэш redis.

Библиотека Terraform поддерживает описание и заказ таких модулей с высоким уровнем детализации.

На рисунке 32 и рисунке 33 представлены фрагменты кода по спецификации и заказу СУБД PostgreSQL и службы кэша ElastiCache AWS являющегося полным, т.н. drop-in replacement, системы управления кэшем redis.

```

module "db" {
  source = "terraform-aws-modules/rds-aurora/aws"
  version = "~> 3.0"

  name           = "vlt1-aurora-db-postgres96"
  engine         = "aurora-postgresql"
  engine_version = "11.9"
  instance_type  = "db.t2.small"

  vpc_id = [aws_subnet.http.id]
  subnet_ids = module.vpc.private_subnets

  replica_count          = 1
  allowed_security_groups = [
    aws_security_group.administration.id,
    aws_security_group.web.id,
  ]
  allowed_cidr_blocks    = ["10.20.0.0/20"]

  storage_encrypted = true
  apply_immediately = true
  monitoring_interval = 10

  db_parameter_group_name      = "default"
  db_cluster_parameter_group_name = "default"

  enabled_cloudwatch_logs_exports = ["postgresql"]

  tags = {
    Environment = "prod"
  }

```

```

    Terraform    = "true"
  }
}

```

Рисунок 32 – Заказ и конфигурация СУБД

```

module "redis" {
  source = " terraform-aws-modules /elasticache-redis/aws"
  version = "~> 1.5.0"

  name_prefix          = "vltl "
  number_cache_clusters = 2
  node_type            = "cache.t1.small"

  engine_version      = "5.0.6"
  port                = 6379
  maintenance_window  = "mon:03:00-mon:04:00"
  snapshot_window     = "04:00-06:00"
  snapshot_retention_limit = 7

  automatic_failover_enabled = true

  at_rest_encryption_enabled = true
  transit_encryption_enabled = true
  auth_token                  = [auth_token]

  apply_immediately = true
  family            = "redis5.0"
  description       = "vltl elasticache redis."

  subnet_ids = module.vpc.private_subnets
  vpc_id     = [aws_subnet.http.id]

  ingress_cidr_blocks = ["0.0.0.0/0"]

  parameter = [
    {
      name = "repl-backlog-size"
      value = "16384"
    }
  ]

  tags = {
    Project = "vltl"
  }
}

```

Рисунок 33 – Заказ и конфигурация кэша

## 9.4 Непрерывное развертывание

Непрерывное развертывание – это практика разработки программного обеспечения, когда при любых изменениях в программном коде выполняется автоматическая сборка, тестирование и развертывание кода на рабочий сервер. Непрерывное развертывание является одним из основополагающих принципов разработки современных приложений, поскольку расширяет практику

непрерывной интеграции за счет того, что все изменения кода после стадии сборки развертываются в тестовой и/или в рабочей среде. Таким образом, у разработчика всегда будет готовый к развертыванию собранный экземпляр ПО, прошедший стандартизированную процедуру тестирования. Диаграмма состояний кода при использовании принципов CI/CD (англ. Continuous Integration / Continuous Deployment) представлена на Рисунок 34.

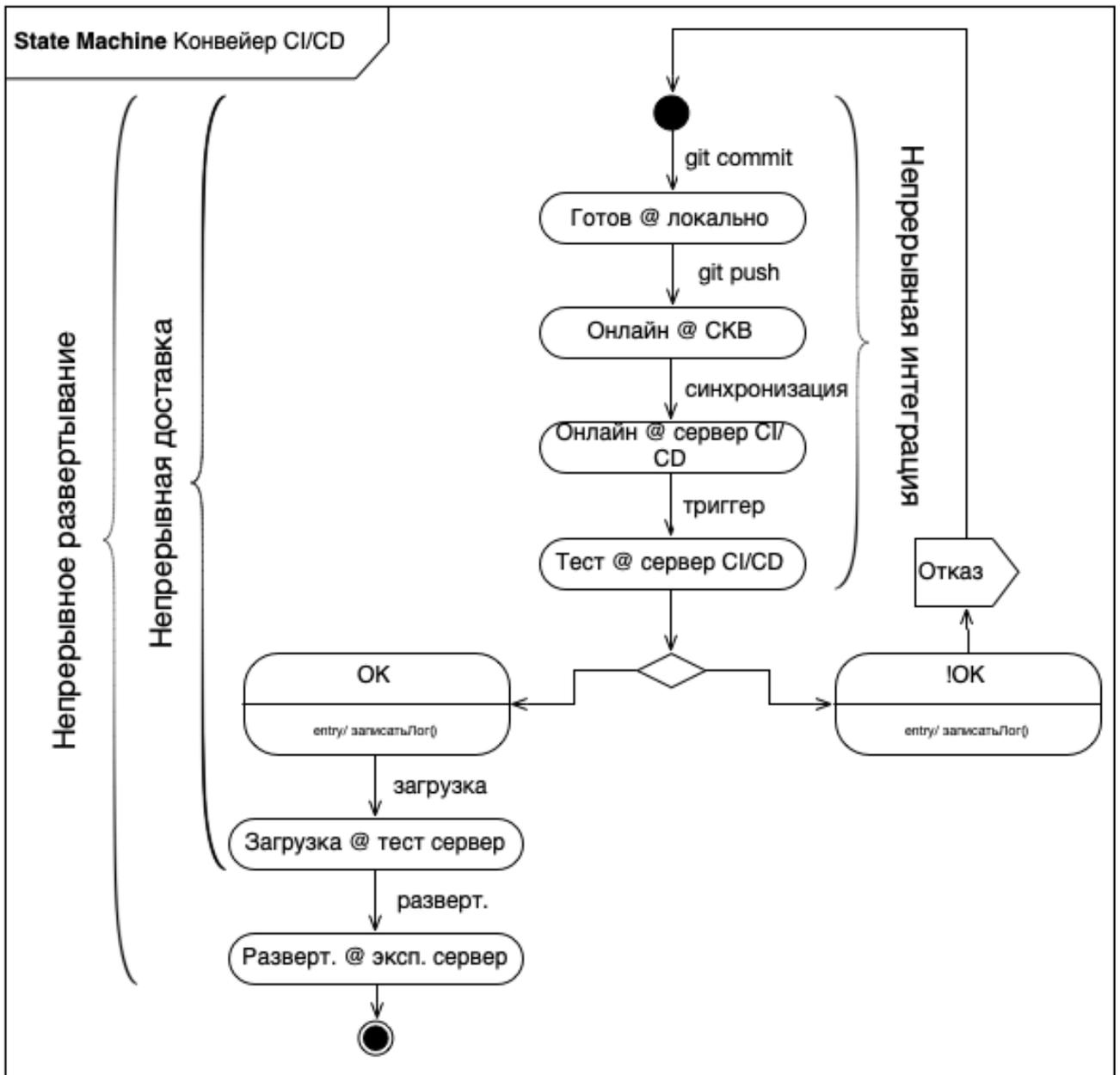


Рисунок 34 – Диаграмма состояний кода при CI/CD

Весь процесс можно представить как последовательное выполнение следующих этапов:

1. Разработчик отправляет свои изменения в систему контроля версий;

2. сервер CI/CD синхронизирует код;

3. выполняется сборка и тестирование поступивших изменений;

4. после успешной интеграции, собранный пакет выкладывается на тестовый сервер;

5. заинтересованные лица получают уведомления о выкладке новой версии ПО на тестовый сервер. Начинается вторая фаза тестирования – запускаются системные и системные интеграционные тесты;

6. в зависимости от настроек политики развертывания кода, развертывание в эксплуатацию происходит либо вручную после получения разрешения от ответственного лица, либо автоматически.

Основные преимущества использования CI/CD:

- Скорость вывода новой функциональности от запроса клиента до запуска в эксплуатацию, CI/CD позволяет запускать обновления за считанные дни или недели по сравнению с целым календарным годом при классическом «waterfall»-подходе;

- возможность выбора оптимального варианта за счет оперативного тестирования и большего числа итераций;

- качество итогового результата выше: автоматическое тестирование охватывает все аспекты продукта, что труднореализуемо при стандартном порелизном подходе. Все ошибки и тонкие места выявляются и удаляются ещё на ранних этапах разработки.

Сервисы GitHub и TravisCI совместно со службой AWS CodeDeploy полностью поддерживают конвейер непрерывной интеграции и непрерывного развёртывания. Для настройки конвейера необходимо описать параметры развёртывания и параметры провайдера на мощностях которого необходимо выполнить развёртывание протестированного и собранного продукта.

На Рисунок 35 представлен пример конфигурационного файла для сервиса Travis CI.

```
language: Python
Python:
- '3.9'
cache:
```

```

directories:
  - venv
deploy:
- provider: s3
  access_key_id: $AWS_ACCESS_KEY
  secret_access_key: $AWS_SECRET_KEY
  local_dir: project_upload
  skip_cleanup: true
  on: &2
  repo: ruslaniv/vl1l
  bucket: vl1l_s3_bucket
  region: eu-central-1
- provider: codedeploy
  access_key_id: $AWS_ACCESS_KEY
  secret_access_key: $AWS_SECRET_KEY
  bucket: vl1l_s3_bucket
  key: latest.zip
  bundle_type: zip
  application: vl1l
  deployment_group: deployers
  region: eu-central-1
  on: *2
script:
  - zip -r latest *
  - mkdir -p project_upload
  - mv latest.zip project_upload /latest.zip

```

Рисунок 35 – Пример файла конфигурации CI/CD

## 9.5 Выводы по главе

В результате выполнения работ по данной главе, выполнено описание требуемых вычислительных ресурсов с помощью декларативного языка описаний библиотеки «Terraform», а также выполнено описание конфигурации системы с помощью декларативного языка описаний библиотеки «Ansible».

Выполнена настройка конвейера непрерывной интеграции и непрерывного развертывания с помощью сервисов «Travis CI» и «Amazon Code Deploy».

В результате, получена готовая инфраструктура, соответствующая заданной спецификации и настроено автоматическое развертывания кода на данной инфраструктуре.

## 10. Описание работы веб-приложения

Веб-приложение реализовано в соответствии с клиент-серверной архитектурой и может быть развернута на любом сервере с запущенной службой веб-сервера. Доступ к программе осуществляется через браузер клиента.

Веб-приложение принимает в качестве ввода информацию о ценных бумагах пользователя – количестве каждого наименования ценной бумаги, дату и цену приобретения каждого наименования ценной бумаги и выводит на экран соответствующие графики.

### 10.1 Общая статистика

Веб-приложение рассчитывает общую статистическую информацию по всему портфелю в целом, а также по каждой ценной бумаге в портфеле. Пример представлен на Рисунок 36.

Статистика по портфелю								
Доходность (годовая)	Риск (годовой)	Скошенность	Эксцесс	Cornish Fischer VaR (5%)	Исторический cVaR (5%)	Коэффициент Шарпа	Максимальная просадка	
9.79 %	22.51 %	-0.29	8.55	2.22 %	3.23 %	38.81 %	-40.69 %	

Статистика по ценным бумагам портфеля								
ЦБ	Доходность (годовая)	Риск (годовой)	Скошенность	Эксцесс	Cornish Fischer VaR (5%)	Исторический cVaR (5%)	Коэффициент Шарпа	Максимальная просадка
SBER	11.76 %	32.60 %	-0.27	8.89	3.20 %	4.71 %	26.09 %	-51.69 %
GAZP	1.52 %	26.76 %	0.36	10.11	2.32 %	3.56 %	-5.37 %	-56.05 %
LKOH	11.40 %	26.91 %	-0.20	13.61	2.45 %	3.72 %	30.33 %	-44.11 %
GMKN	16.57 %	30.10 %	-0.14	10.32	2.82 %	4.31 %	43.76 %	-47.49 %
VTBR	-3.06 %	29.77 %	-0.20	9.75	2.91 %	4.23 %	-19.78 %	-74.68 %
ROSN	6.12 %	28.03 %	-0.04	9.44	2.64 %	3.81 %	10.81 %	-54.30 %
NVTK	20.12 %	31.02 %	0.07	8.90	2.83 %	4.22 %	53.60 %	-48.13 %

Рисунок 36 – Пример расчета статистических показателей

### 10.2 Модель Марковица

На данной вкладке, веб-приложение выводит на экран эффективное множество портфелей, а также два наилучших портфеля, рекомендуемых к покупке:

1. Портфель, обеспечивающий максимальную доходность на единицу риска.

## 2. Портфель, обеспечивающий минимальный риск.

Пример представлен на Рисунок 37.

### Эффективное множество портфелей

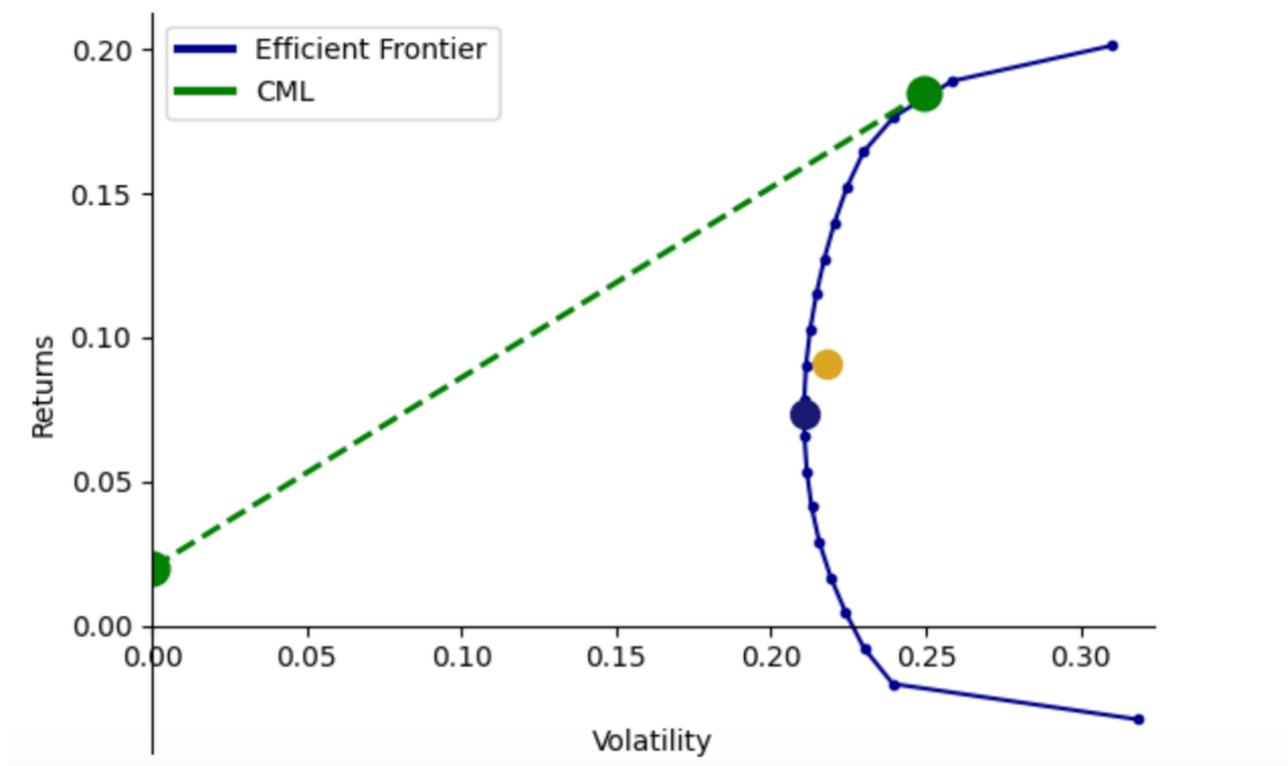


Рисунок 37 – Эффективное множество

Также, модель рассчитывает веса, позволяющие получить данные портфели и выводит их в табличном виде. Пример представлен на Рисунок 38.

Портфель с минимальным риском								
GAZP	LKOH	GMKN	VTBR	ROSN	NVTK	NLMK	CHMF	HYDR
16 %	19 %	17 %	11 %	7 %	6 %	3 %	6 %	15 %
Портфель с максимальным коэффициентом Шарпа								
GMKN			NVTK			CHMF		
28 %			49 %			23 %		

Рисунок 38 – Веса по Марковицу

## 10.3 Бюджетирование риска

Метод бюджетирования риска позволяет не учитывать параметр доходности как чрезвычайно трудный в прогнозировании и оперирует только параметром риска. Метод реализован в двух модификациях:

- Паритет риска.
- Бюджетирование риска.

### 10.3.1 Паритет риска

В данной модели вклад доли влияния риска в общий риск портфеля берется равным для всех бумаг в портфеле. Программа рассчитывает веса, требуемые для достижения такого состояния. Пример представлен на Рисунок 39.

Портфель паритета риска											
	SBER	GAZP	LKOH	GMKN	VTBR	ROSN	NVTK	NLMK	TATN	CHMF	HYDR
Оптимальные веса	8.06	9.62	9.75	10.08	9.26	9.18	8.91	8.50	8.01	8.81	9.81
RC	1.97	1.97	1.97	1.97	1.97	1.97	1.97	1.97	1.97	1.97	1.97
RC (нормированный)	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09

Рисунок 39 – Портфель паритета риска

### 10.3.2 Бюджет риска

Также, можно непосредственно задать бюджеты риска, явно указав, какова должна быть доля влияния риска каждой ценной бумаги портфеля на риск портфеля. Программа подбирает требуемые веса для достижения данного состояния. Пример представлен на Рисунок 40.

Портфель с бюджетирование риска											
	SBER	GAZP	LKOH	GMKN	VTBR	ROSN	NVTK	NLMK	TATN	CHMF	HYDR
Оптимальные веса	4.67	10.55	15.19	19.78	10.39	14.44	7.02	5.12	6.99	2.16	3.68
RC	1.08	2.16	3.25	4.33	2.16	3.25	1.51	1.08	1.73	0.43	0.65
RC (нормированный)	5.00	10.00	15.00	20.00	10.00	15.00	7.00	5.00	8.00	2.00	3.00

Рисунок 40 – Бюджет риска

## 10.4 Модель Блэка-Литтермана

Модель Блэка-Литтермана реализует Байесовскую интерпретацию вероятности, которая учитывает как историческое данные о доходности ценной бумаги и ее риски, так и мнение о будущей доходности и риски ценной бумаги, который выражен через неуверенность в таком мнении.

В первую очередь программа рассчитывает историческую доходность ценных бумаг портфеля, скорректированных на их риск, а также риск этих

ценных бумаг, выраженный ковариационной матрицей. Пример представлен на Рисунок 41.

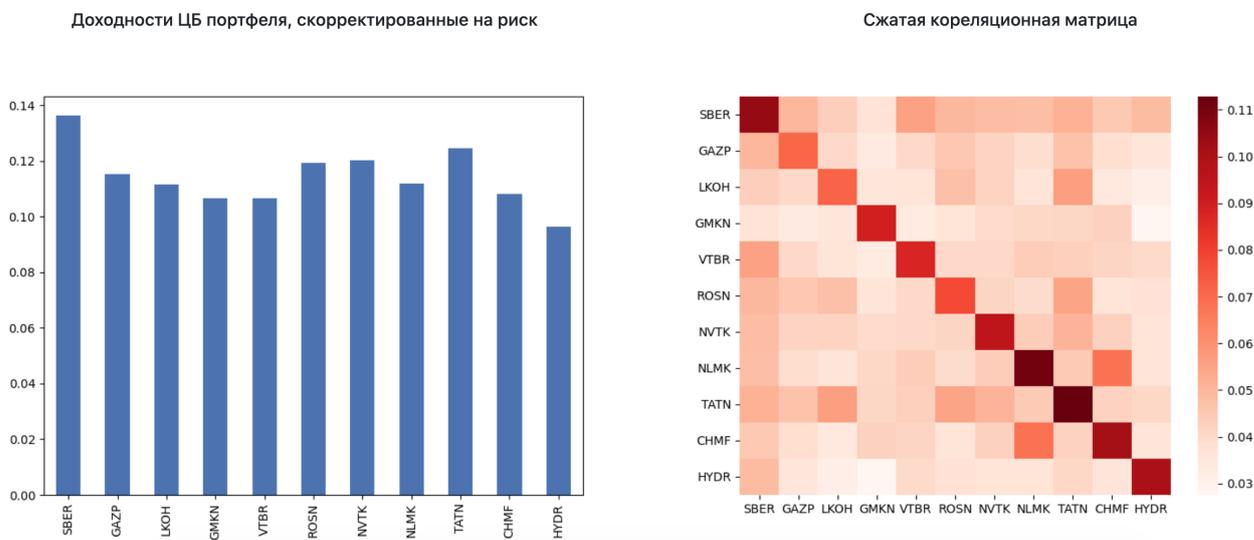


Рисунок 41 – Априорные параметры модели

Затем необходимо сформировать мнение относительно ожидаемой доходности ценных бумаг портфеля. Исследованию методов определения ожидаемой доходности ценных бумаг будет посвящена научно-исследовательская работа, планируемая к выполнению в рамках аспирантуры.

Эти мнения могут быть выражены как в абсолютной, так и в относительной форме. Также, необходимо выразить степень неуверенности в таком мнении для каждой ценной бумаги, что и будет являться ее риском. Пример представлен на Рисунок 42.

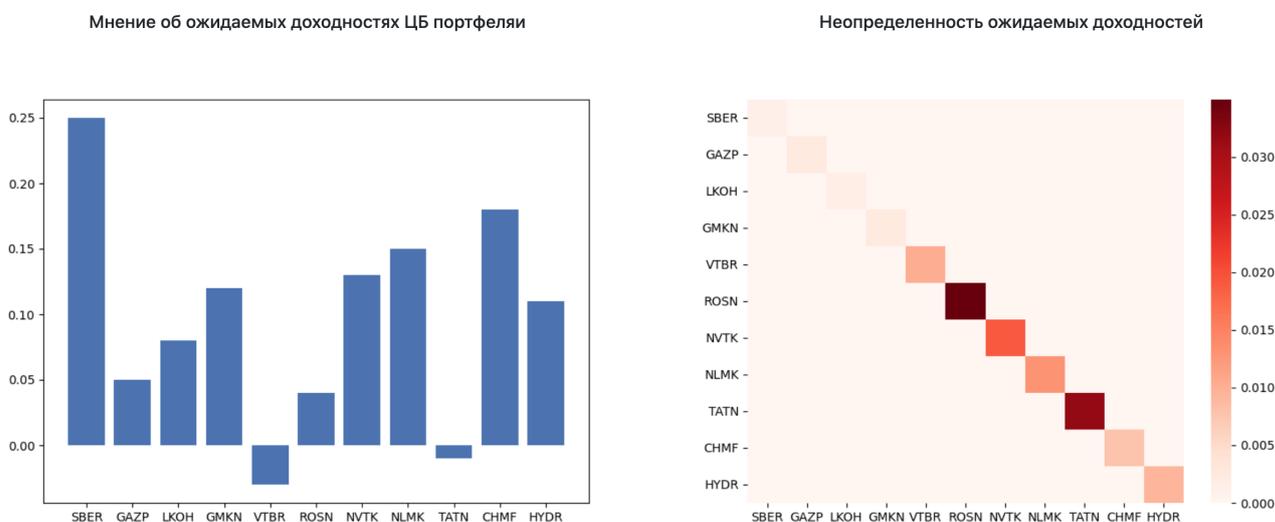


Рисунок 42 – Апостериорные параметры модели

Получив эти данные, алгоритм отображает ожидаемые доходности, которые представляют собой взвешенную сумму априорных и апостериорных данных. Пример представлен на Рисунок 43.

Модель Блэка-Литтермана

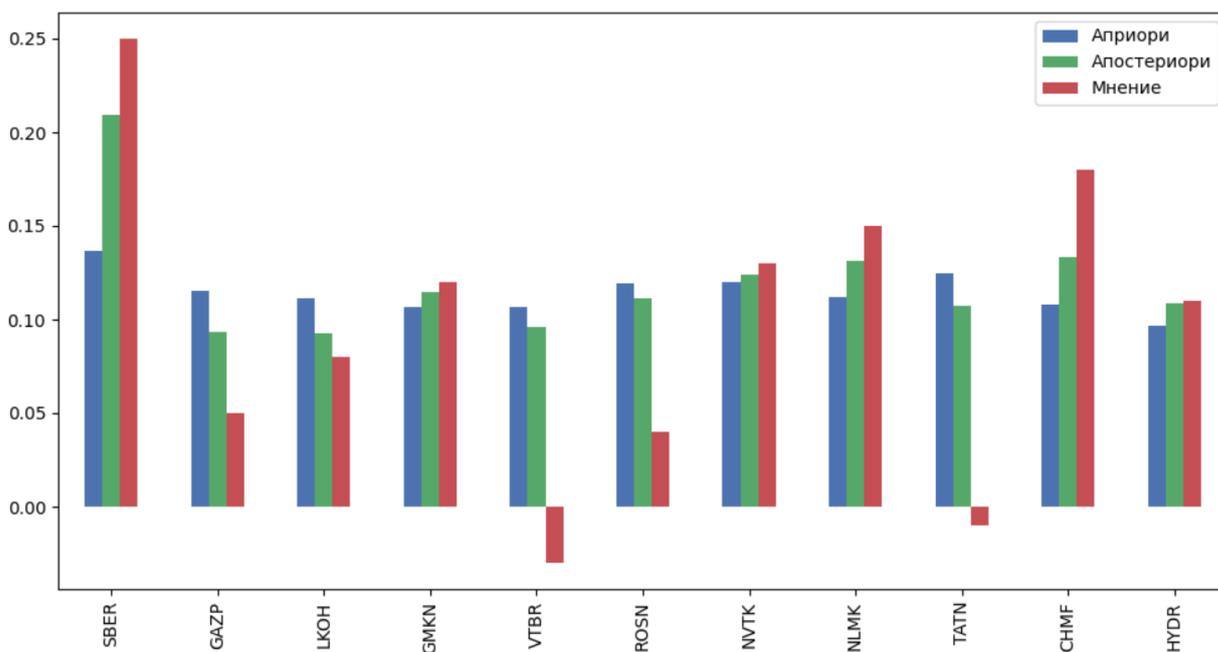


Рисунок 43 – Прогноз модели

В заключении модель отображает веса для каждой ценной бумаги портфеля, позволяющие получить максимальную доходность при минимальном уровне риска. Пример представлен на Рисунок 44.

Новые веса портфеля

SBER	GAZP	LKOH	GMKN	VTBR	ROSN	NVTK	NLMK	TATN	CHMF	HYDR
20.35	5.75	5.88	8.92	5.88	8.05	9.54	10.15	6.87	10.66	7.94

Рисунок 44 – Новые веса модели

## 10.5 Выводы по главе

В данном разделе продемонстрирована работа веб-приложения по оптимизации портфеля ценных бумаг по параметру риска.

Представлены примеры оценки как всего портфеля, так и каждой ценной бумаги статистическими методами.

Представлен пример оценки риска портфеля в соответствии с моделью Марковица, приведены рекомендации по ребалансировке портфеля для достижения расчетных показателей.

Представлен пример работы двух модификаций модели бюджетирования – создание портфеля паритета риска и портфеля с бюджетами риска.

Представлен пример работы модели Блэка-Литтермана на каждом этапе ее работы – расчет исторических значений риска и доходности и оценка ожидаемых значений и, затем, расчет выходных параметров модели – ожидаемой доходности активов портфеля.

# 11. ВКР как стартап

## 11.1 Актуальность

Проблема современной инвестиционной культуры в целом и диалога между финансовыми институтами и частными инвесторами в частности состоит в том, фокус дискурса всегда смещен в сторону доходности. Достаточно посмотреть любой рекламный проспект инвестиционной, брокерской компании или банка и убедиться, что фокус сделан на получение инвестором доходности. О рисках обычно пишут мелким шрифтом, на последней странице и только то, что он существует. Как его измерить, в каких единицах, как сравнить, как минимизировать?

Если рассмотреть многочисленные мобильные и прочие приложения, предоставляемые в пользование банками (Сбербанк Инвестиции, Робот-Советник ВТБ, Тинькофф и прочие), то вся работа с риском инвестиций сводится к определению «толерантности» пользователя к риску (чтобы это ни значило, т.к. неизвестно на каких теориях и/или принципах базируются данные оценки и методы при этом применяемые – эта информация не банками не раскрывается). Затем пользователю предлагается инвестировать в готовый модельный портфель с тем или иным распределением средств между «высоко-рискованными» и «низко-рискованными» инструментами (обычно это акции и облигации), без какой-либо количественной оценки риска такого портфеля.

Самостоятельное выполнение подобных расчетов является хотя и возможной, но относительно сложной задачей, т.к. требует междисциплинарных знаний, на пересечении математики, статистик, теории вероятности и др.

Предлагаемый продукт абстрагирует весь математический аппарат, предлагая пользователю простой и понятный интерфейс в виде таблиц и графиков.

## **11.2 Конкурентный анализ рынка**

### **11.2.1 Общие сведения**

В соответствии с определением инвестирования, люди занимались им с момента появления денег или даже раньше, если под капиталом понимать не только деньги, но и материальные ценности. Однако в данном исследовании будет рассмотрена одна из форм инвестирования, которая появилась в XV – XVI веках и представляет собой организованную, регулируемую законом форму размещения капитала в ценных бумагах компании, доступную как профессионалам, так и простым гражданам. Таким образом, можно вывести определение понятия «инвестирования на фондовом рынке», т.к., именно, в рамках этого процесса будет проводиться исследование методов решения задач.

Специально организованный для совершения сделок рынок называется биржей, а посредник – брокером, который занимается регистрацией сделок. Задача биржи – предоставить площадку для заключения сделок, а также оперативно довести до всех участников торгов текущую стоимость права на капитал той или иной компании, которое называется акцией. В свою очередь задача брокера – зафиксировать факт сделки купли-продажи акции. Через брокеров заключаются все сделки розничных, непрофессиональных инвесторов, а также большая часть сделок профессиональных инвесторов. На бирже могут работать только крупные юридические лица, с большим собственным капиталом, гарантирующим оплату по всем сделкам заключенным при посредничестве данного лица.

Данный процесс чрезвычайно важен и выгоден как для экономике страны в целом, так и для участников торгов в частности.

Выгоды для экономики:

- Увеличивается оборачиваемость денег, что ведет к росту ВВП без увеличения денежной массы, т.е. к росту деловой активности, росту благосостояния населения при нулевой инфляции.

Выгода для продавца акций:

- Продавец получает дешевое финансирование на развитие бизнеса, т.к. дивидендные ставки на порядок ниже ставок кредитования.

Выгода для покупателя акций:

- Покупатель не только сохраняет деньги от обесценивания, но и может получить прибыль от инвестиции, т.к. помимо дивидендной доходности, инвестор может получить курсовую разницу при продаже акции.

Именно возможность получения прибыли, которая может исчисляться десятками и даже сотнями процентов, делает инвестиции одним из самых привлекательных видов вложений. А для непрофессиональных инвесторов – пожалуй единственным видом вложений, который потенциально может обеспечить столь высокую доходность.

Однако даже на бытовом, наивном уровне каждый инвестор понимает, что инвестиция связана с определенной долей риска. И, в первую очередь, инвесторы ассоциируют понятие инвестиционного риска с возможностью потери вложенных средств.

Следует отметить, что, в данном случае, неважно в какую сторону изменяется цена, в положительную для инвестора сторону или нет. Сама вероятность того, что цена будет изменяться непредвиденно, не т.к. планирует инвестор, включая на большую чем планировалось величину, является риском ценной бумаги. Естественно, непрофессиональные инвесторы не рассматривают непредвиденное изменение цены в свою пользу как риск и часто приписывают такие изменения удаче или своему профессиональному опыту. Однако такое изменение цены в первую очередь характеризует волатильность ценной бумаги, т.е. степень изменчивости ее цены. Естественно, цена бумаги может измениться и в невыгодную для инвестора сторону, создавая нереализованные убытки, которые могут привести к вполне реальным убыткам.

Таким образом, риск – это мера волатильности, изменчивости стоимости ценной бумаги. И для инвестора очень важно уметь измерять эту метрику или иметь инструменты для ее измерения и анализа. Ведь только зная степень риска ценной бумаги, инвестор может оценить – дает ли ему / ей данная бумага

достаточную доходность для того, чтобы оправдать принимаемый риск. А также, сравнить риск одной ценной бумаги с другой и выбрать ту, которая характеризуется меньшим риском при такой же ожидаемой доходности.

В разделе 11.2.2 будут рассмотрены представленные на рынке инструменты, предназначенные для инвестиционного анализа, и будет проведен их сравнительный анализ с целью оценки – имеют ли данные инструменты необходимый функционал для определения степени риска ценных бумаг.

## **11.2.2 Обзор существующих решений**

В настоящее время на рынке представлено большое количество инструментов в виде сайтов, приложений и специализированного программного обеспечения, призванного помочь инвесторам в принятии решений о покупке или продаже ценных бумаг. В обзоре рассмотрены наиболее популярные решения, представленные как на международном, так и на российском рынке.

### **11.2.2.1 Веб-приложения**

Веб приложение в контексте данного анализа – это сайт или веб-приложение, размещенное в сети Интернет, доступное через любой браузер и с любой ОС и предоставляемое пользователю по модели SaaS. Веб-приложение может быть бесплатным или платным. Во втором случае доступ к приложению может предоставляться непосредственно брокером, после заключение договора на брокерское обслуживание, или же третьей стороной для отслеживания состояния инвестиционного портфеля.

#### **11.2.2.2 Yahoo finance**

«Yahoo Finance» – это экономический портал, размещенный по адресу [finance.yahoo.com](http://finance.yahoo.com) и принадлежащий компании «Yahoo Inc». По данным рейтингового агентства «Alexa» [16] данный портал является самым крупным сайтом в сети Интернет предоставляющим финансово-экономическую информацию и занимает 11 место среди всех сайтов сети Интернет по количеству посетителей.

На главной странице сайта отражаются основные рыночные показатели по рынку США, а также экономические новости (Рисунок 45).

The screenshot displays the Yahoo Finance website interface. At the top, there is a navigation bar with links for Entertainment, Search, Mobile, and More. Below this is the Yahoo Finance logo and a search bar. The main content area features a row of market indices: S&P 500 (3,240.14, -6.14 (-0.19%)), Dow 30 (28,606.12, -97.26 (-0.34%)), Nasdaq (9,085.42, +13.96 (+0.15%)), Russell 2000 (1,660.02, -3.24 (-0.19%)), Crude Oil (62.58, -0.69 (-1.09%)), and Gold (1,572.70, +3.90 (+0.25%)). A central video player shows a 'Yahoo Finance LIVE' broadcast with the title '>> A LOT OF ROOM TO IMPROVE WHEN IT COUPLES TO DANCE'. Below the video are several news snippets: 'Stocks fall, Dow sheds over 100 points', 'What slow population growth means for Congress', 'New phone covered in 24-karat gold, leather', 'Planet Fitness CEO: We're ready for explosive growth', and 'New North American trade deal advances'. On the right side, there is a 'Quote Lookup' section and a 'My Portfolio & Markets' section with a table of stock prices and changes.

Symbol	Last Price	Change	% Change
AAPL	299.86	+0.06	+0.02%
C	78.88	-0.57	-0.72%
GE	11.97	-0.18	-1.44%
GOOG	1,399.67	+5.46	+0.39%
WMT	116.24	-1.41	-1.20%
XOM	69.69	-1.18	-1.66%

Рисунок 45 – Главная страница сайта Yahoo Finance

Непосредственно анализ портфеля ценных бумаг выполняется на отдельной вкладке «My Portfolio», на которой отображаются сводная информация по портфелю ценных бумаг инвестора (Рисунок 46).

При использовании бесплатной версии портала, пользователю доступны следующие вкладки:

- «Summary» – сводная информация по портфелю.
- «My Holdings» – показатели эффективности портфеля в разрезе прибыли или убытков.
- «Fundamentals» – фундаментальная информация по акциям в портфеле, т.е. основные финансовые показатели деятельности компании.

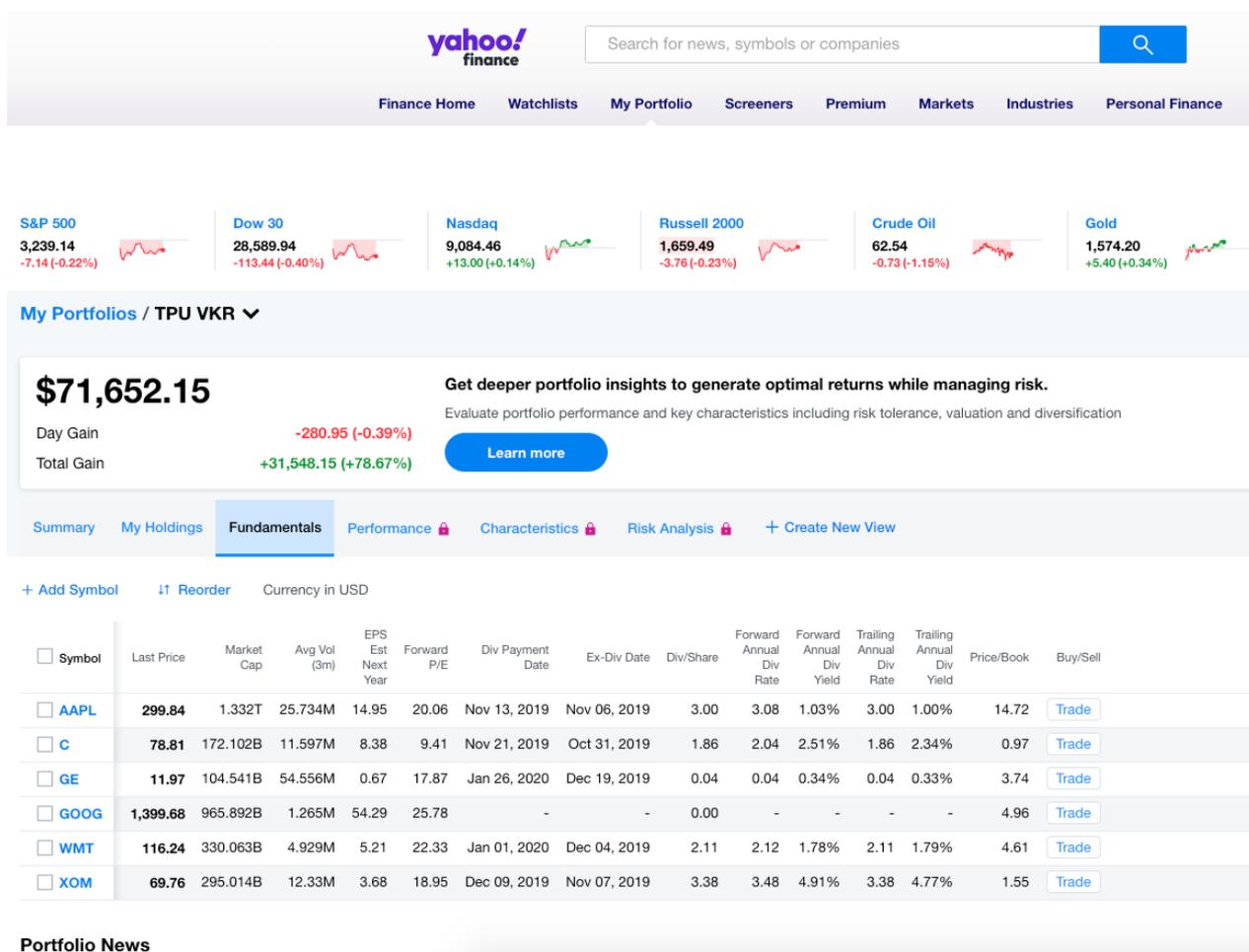


Рисунок 46 – Сводный анализ портфеля ЦБ на сайте Yahoo Finance

Остальные вкладки и, в частности, вкладка «Risk Analysis», в которой представлены инструменты анализа рисков портфеля, доступна только платным подписчикам сервиса, причем цена подписки составляет 349,99 долларов США в год. Если для жителя США это составляет около 9% от средней месячной заработной платы, то для жителя РФ – около 80%. Поэтому для частного российского инвестора, для которого инвестирование не является профессиональной деятельностью, данный сервис вряд ли можно назвать доступным.

Однако даже после приобретения подписки выясняется, что сервис предоставляет минимальный функционал для оценки риска портфеля. Для портфеля рассчитываются только несколько базовых показателей, такие как корреляционно-регрессионная матрица ценных бумаг портфеля и бета коэффициенты как для индивидуальных ценных бумаг, так и для всего портфеля. Инструменты для оптимизации портфеля по параметру риска отсутствуют.

### 11.2.2.3 МФД-ИнфоЦентр

Портал «МФД-ИнфоЦентр», размещенный в сети Интернет по адресу mfd.ru, принадлежит информационному агентству «МФД-ИнфоЦентр», которое специализируется на финансовой информации и создании современных высокотехнологичных программных продуктов для банков, инвестиционных компаний, корпоративных и индивидуальных инвесторов. На сайте представлены новости и аналитика, анонсы публикаций, курсы валют и их архивы, котировки и технический анализ ценных бумаг и индексов. Сайт является одним из крупнейших финансовых порталов в российском сегменте сети Интернет из занимает 2 207 место по числу посетителей в доменной зоне «ru».

На домашней странице сайта представлена сводная экономическая по российскому фондовому рынку, результаты торгов, а также экономические новости (Рисунок 47).

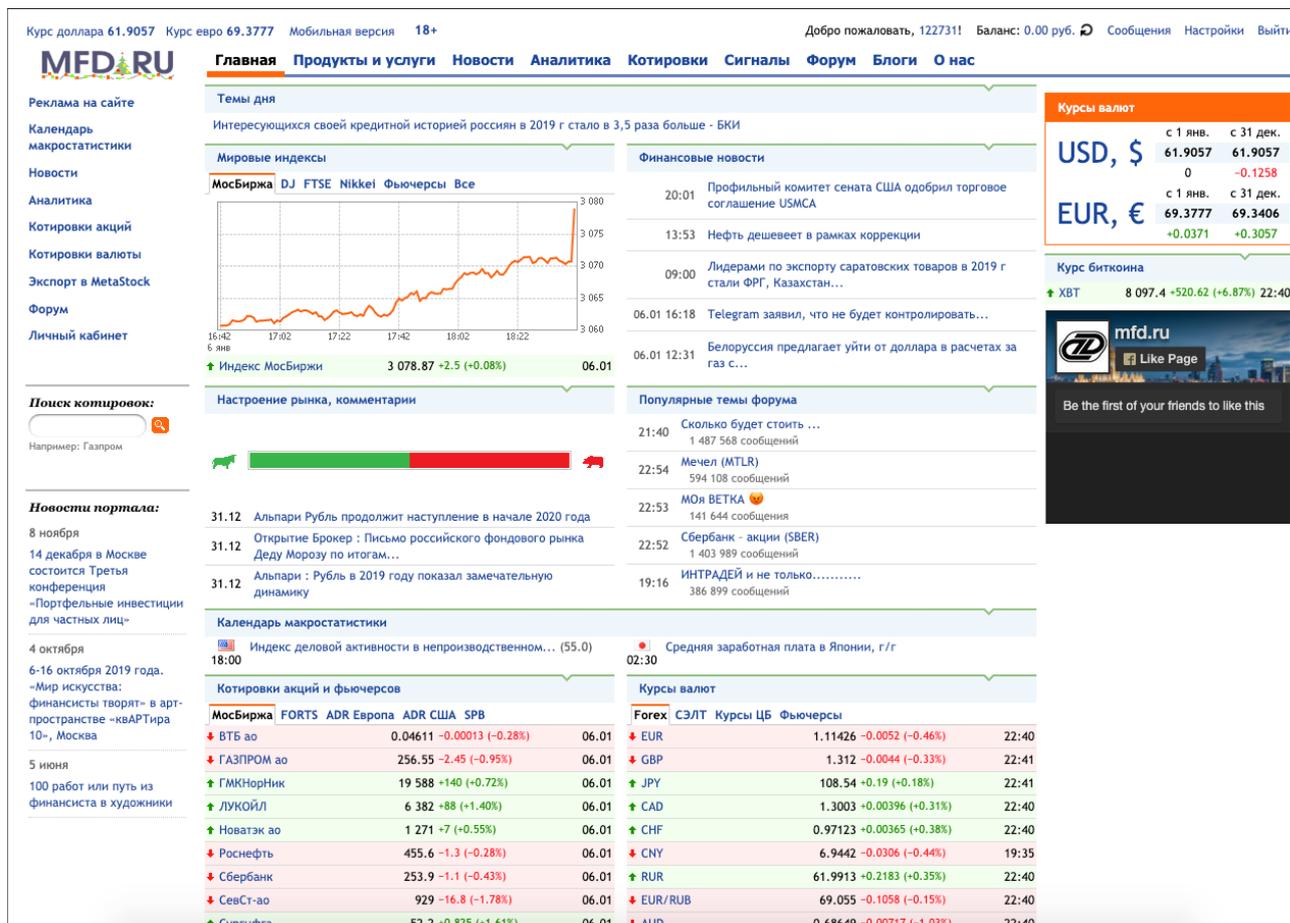


Рисунок 47 – Главная страница сайта МФД

В отличие от сайта Yahoo Finance, на сайте МФД отсутствует возможность отслеживать показатели эффективности модельного портфеля даже в терминах доходности. Присутствует возможность анализа рисков, которая доступна на вкладке «Аналитика» → «Анализ рисков». Однако, данный инструментарий позволяет проводить только самый примитивный анализ риска единичной ценной бумаги, с расчетом исторической волатильности, коэффициента корреляции относительно индекса ММВБ и  $\beta$ -коэффициента ценной бумаги (Рисунок 48). Анализ риска и оптимизации модельного портфеля не предусмотрен.

[Главная](#) [Продукты и услуги](#) [Новости](#) [Аналитика](#) [Котировки](#) [Сигналы](#) [Форум](#) [Блоги](#) [О нас](#)

mfd.ru → Аналитика → Анализ рисков

### Анализ рисков

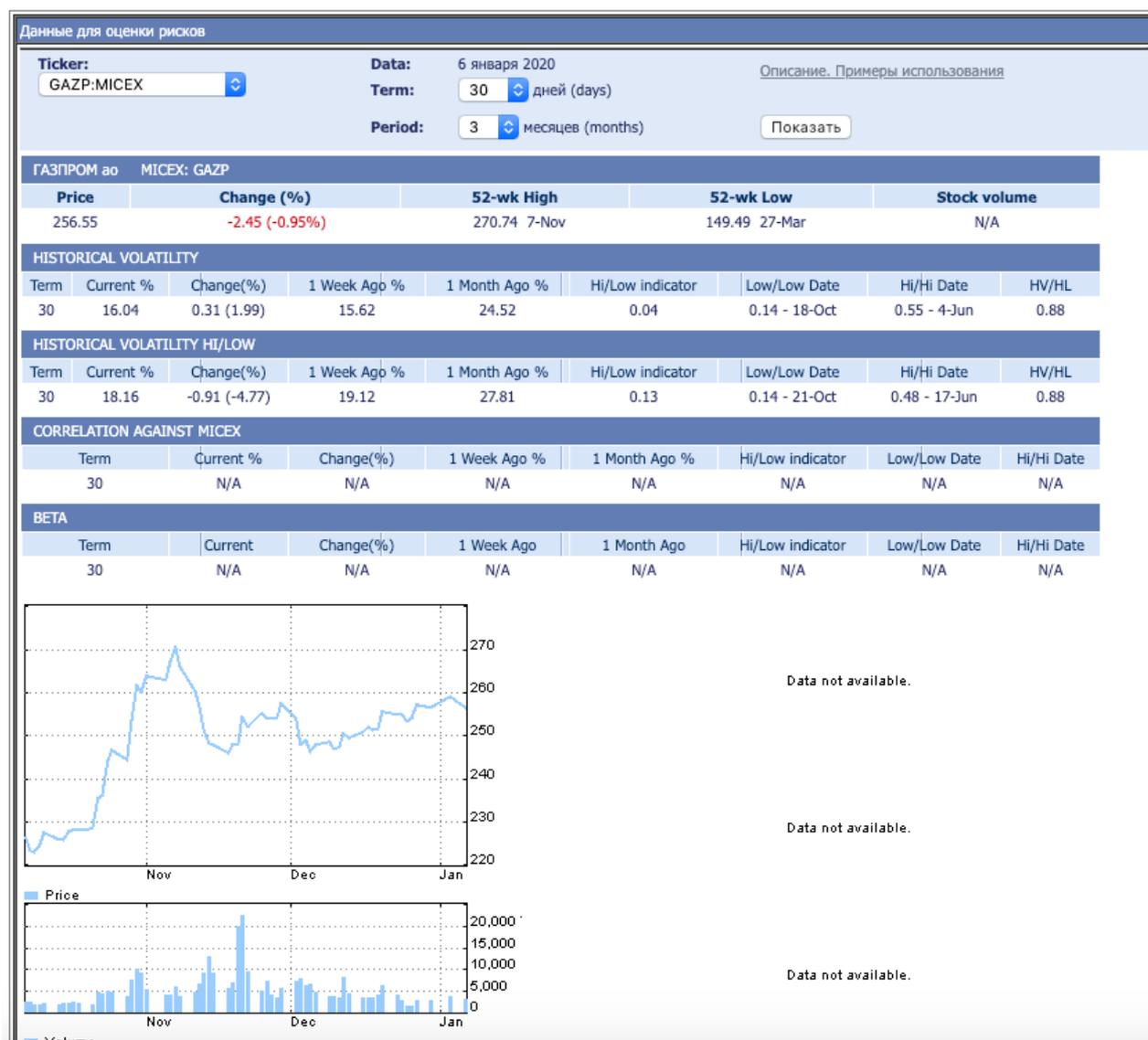


Рисунок 48 – Анализ рисков ЦБ на сайте МФД

### 11.2.2.4 АйТи Капитал

ITI Capital (АО ИК «Ай Ти Инвест») – это современная финансовая компания, предоставляющая брокерские услуги на российском рынке с 2000 года. После заключения договора на брокерское обслуживание, инвестор получает логин и пароль для доступа к личному кабинету, реализованному в виде веб-приложения. В личном кабинете отражается текущее состояние портфеля инвестора, а также присутствует возможность отправлять заявки по покупке или продажу ЦБ в систему электронных торгов биржи и следить за ходом торгов, получая данные о текущих котировках ЦБ в режиме минимум 5 секундной задержки. Пример страницы личного кабинета инвестора представлен ниже (Рисунок 49).

The screenshot displays the investor's account page for account number BP4555-MS-01 on the MOEX market. The interface includes a top navigation bar with the ITI Capital logo, a menu icon, and a 'Помощь' (Help) button. Below the navigation bar, there are buttons for 'Открыть ИИС' (Open IIS) and 'М Е Н Ю' (Menu). The main content area shows the account name 'Иштуганов Руслан Александрович' and the account number 'Счет № BP4555-MS-01'. It also displays the tariff plan 'Универсальный / Депо №3' and the portfolio status 'Ограниченный (авто)'. A table provides details on portfolio liquidity, trading availability, and costs. Below this, a section titled 'Мой портфель' (My Portfolio) shows a table with columns for instrument, ticker, type, platform, position, current position, buy orders, sell orders, average price, current price, and unrealized profit. A green box contains a warning icon and text explaining the meaning of trading availability and transferable funds.

Инструмент	Тикер	Тип инструмента	Площадка	Позиция	Текущие позиции**)	В заявках на покупку	В заявках на продажу	Ср.взв. цена	Текущая цена	Нереализ. прибыль
------------	-------	-----------------	----------	---------	--------------------	----------------------	----------------------	--------------	--------------	-------------------

\*) Доступные для торговли - средства, на которые можно купить немаржинальные бумаги. Отрицательное число означает заемные средства.  
Доступные к переводу - средства, которые могут быть переведены с данного портфеля на другой. Для портфелей МО-01 (MS-01) равны сумме, которая может быть выведена с портфеля на банковский счет.

Рисунок 49 – Страница ЛК с отображением данных о состоянии портфеля АйТи Капитал

На странице, отображающей текущее состояние портфеля инвестора, присутствует возможность расчета риска портфеля, но это риск в разрезе понятия «маржинальная торговля». В случае покупки ЦБ на заемные средства, предоставляемые брокером, или продажи ЦБ без покрытия, которые также предоставляются займы брокером, у инвестора возникает задолженность перед брокером, которую он / она обязаны погасить. Риск в данном случае рассчитывается как возможность погасить текущую задолженность при продаже всех ценных бумаг на счете инвестора по текущей цене спроса. Т.е., другими словами, риск портфеля в данном случае – это коэффициент покрытия долга. Других инструментов расчета риска и оптимизации портфеля по показателю риск/доходность не предусмотрено.

### **11.2.3 Приложения**

Приложение в контексте данного анализа – это нативное приложение для определенной платформы и/или ОС, предоставляемое пользователю брокером после заключения договора на брокерское обслуживание, для отслеживания состояния инвестиционного портфеля в реальном времени и подачи заявок на покупку или продажу ценных бумаг.

#### **11.2.3.1 Сбербанк Инвестор**

Сбербанк-Инвестор – это приложение для платформ iOS и Android, предоставляемое в бесплатное пользование инвесторам, заключившим договор брокерского обслуживания с ПАО «Сбербанк». По результатам заключения договора, клиент получает имя пользователя и пароль, с помощью которых можно войти в приложение, предварительно загрузив его в «Google Play Market» или «Apple AppStore». На главном экране приложения отображается состояние инвестиционного портфеля, но только в терминах доходности (Рисунок 50 и Рисунок 51).

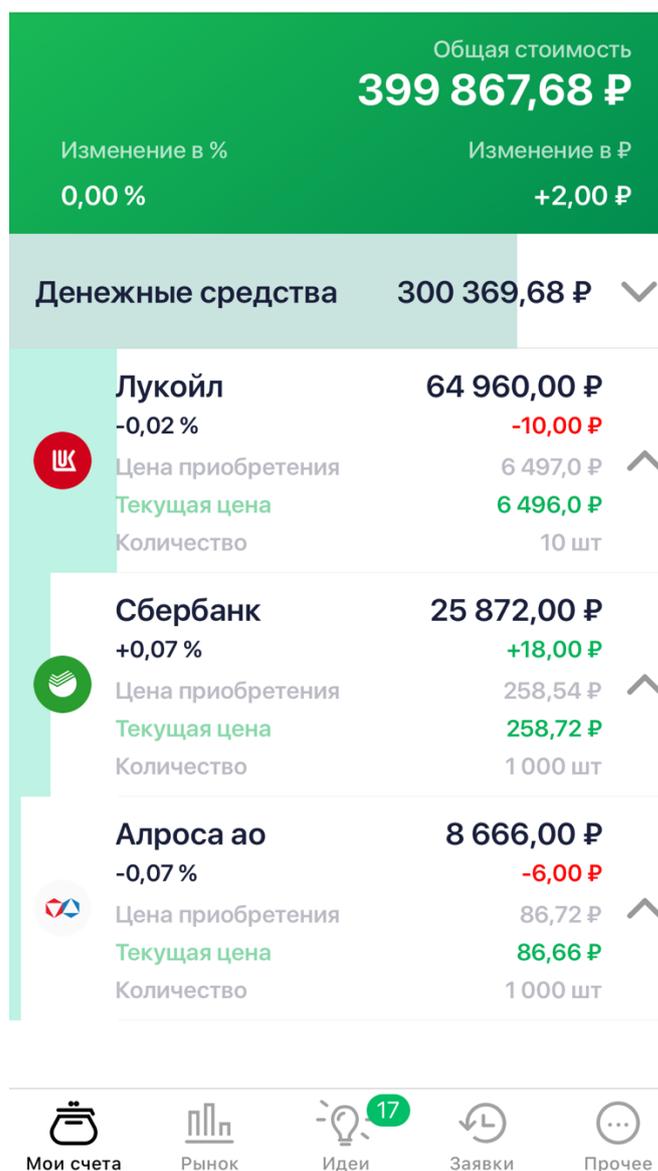


Рисунок 50 – Главный экран приложения Сбербанк Инвестор

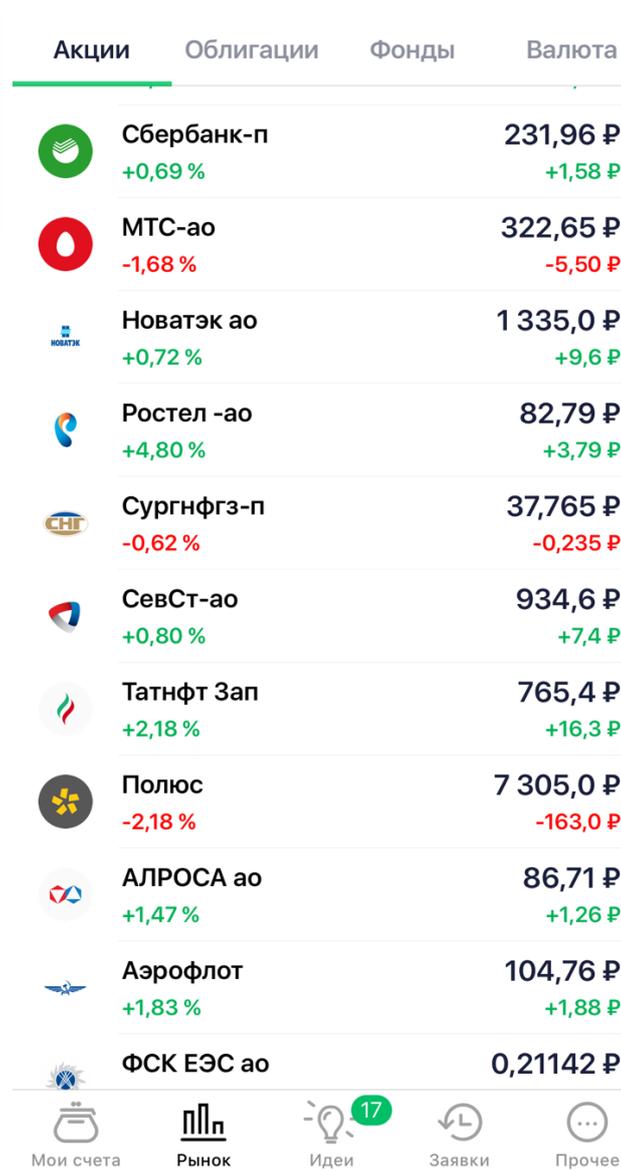


Рисунок 51 – Экран торговли приложения Сбербанк Инвестор

Также, в приложении можно отдавать торговые поручения на покупку или продажу ЦБ. Аналитический инструментарий в приложении отсутствует.

### 11.2.4 Торговые терминалы

Торговый терминал – это нативное приложение для определенной платформы и/или ОС, профессиональное рабочее место биржевого трейдера. Терминал предоставляется в пользование инвестору брокером после заключения договора на брокерское обслуживание и позволяет отправлять заявки на покупку или продажу ценных бумаг непосредственно в электронную торговую систему

биржевых торгов, получать информацию о ходе биржевых торгов в режиме реального времени, с отслеживанием не только текущих котировок, но и баланса спроса и предложения, например количество и объем заявок на покупку и продажу и т.д. Также, терминал позволяет отслеживать состояние инвестиционного портфеля в реальном времени.

### 11.2.4.1 Финам

АО «Инвестиционный холдинг ФИНАМ» – крупнейший в России розничный брокер, а также инвестиционная группа, специализирующаяся на оказании трейдерских, инвестиционных банковских услуг, доверительном управлении денежными средствами и ценными бумагами, инвестировании на валютном рынке Forex. После заключения договора на брокерское обслуживание, инвестор получает доступ к торговому терминалу Transaq (Рисунок 52).

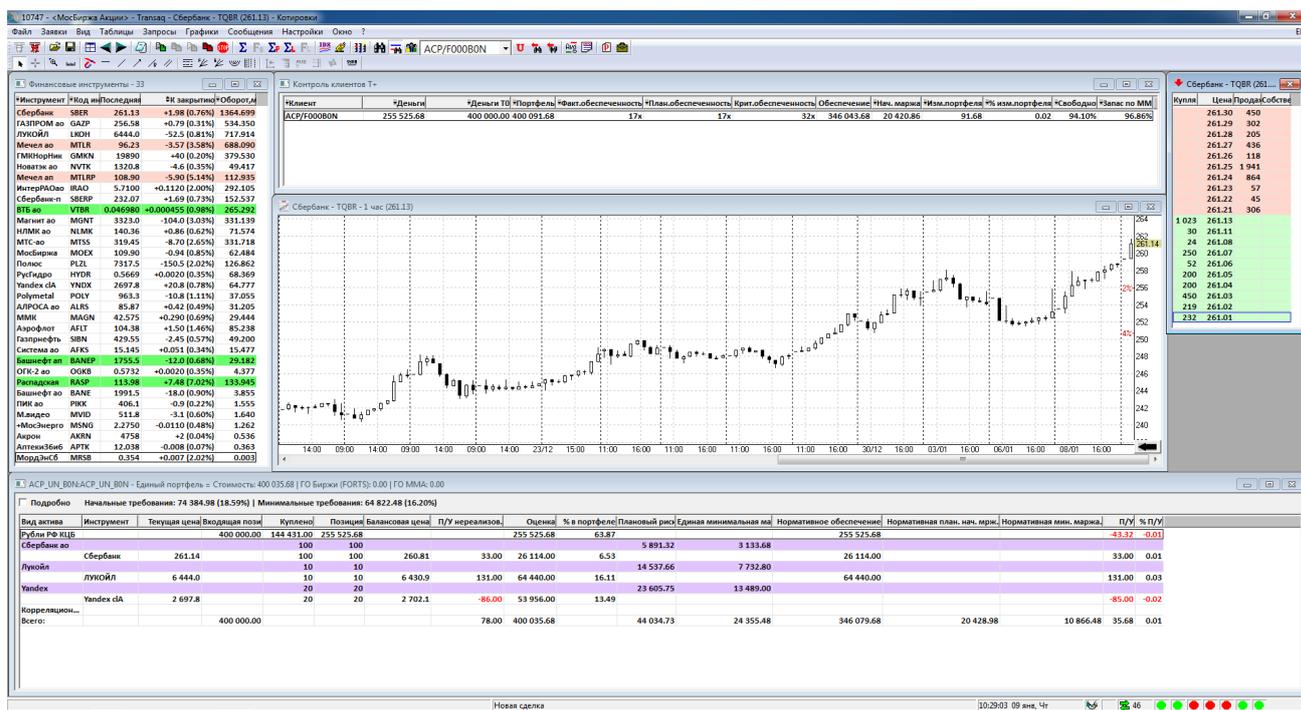


Рисунок 52 – Торговый терминал Transaq компании Финам

В терминале присутствует функция отображения состояния портфеля ЦБ в режиме реального времени (Рисунок 53).

ACP_UN_BON\ACP_UN_BON - Единый портфель - Стоимость: 400 035.68   ГО Биржи (FORTS): 0.00   ГО MMA: 0.00																	
[ Подробнее Начальные требования: 74 384.98 (18.59%)   Минимальные требования: 64 822.48 (16.20%)																	
Вид актива	Инструмент	Текущая цена	Входящая позиция	Куплено	Позиция	Балансовая цена	П/У unrealizov.	Оценка	% в портфеле	Плановый риск	Единая минимальная марж.	Нормативное обеспечение	Нормативная план. нач. марж.	Нормативная мин. маржа.	П/У	% П/У	
Рубли РФ КЦБ			400 000.00	144 431.00	255 525.68			255 525.68	63.87			255 525.68				-43.32	-0.01
Сбербанк ао				100	100					5 891.32	3 133.68						
	Сбербанк	261.14		100	100	260.81	33.00	26 114.00	6.53	14 537.66	7 732.80	26 114.00				33.00	0.01
	Лукойл			10	10												
	ЛУКОЙЛ	6 444.0		10	10	6 430.9	131.00	64 440.00	16.11			64 440.00				131.00	0.03
	Yandex			20	20					23 605.75	13 489.00						
	Yandex clA	2 697.8		20	20	2 702.1	-86.00	53 956.00	13.49							-85.00	-0.02
Корреляцион...																	
Всего:			400 000.00				78.00	400 035.68		44 034.73	24 355.48	346 079.68	20 428.98	10 866.48	35.68	0.01	

Рисунок 53 – Отображение состояния портфеля в торговом терминале Transaq

Данная модель отражает риск портфеля инвестора, но и в этом случае это риск в разрезе понятия «маржинальная торговля». Других инструментов расчета риска и оптимизации портфеля по показателю риск/доходность не предусмотрено.

## 11.2.5 Доверительное управление

Доверительное управление – это управление денежными средствами и ценными бумагами инвестора, которое осуществляется третьим лицом, после заключения договора доверительного управления. Инвестор не принимает самостоятельных решений по поводу покупки или продажи тех или иных ценных бумаг, а лишь высказывает свои пожелания относительно целей и задач, которые он / она решает, инвестируя денежные средства. На основе этих данных управляющий принимает инвестиционные решения. Это более «элитный» вид инвестирования, т.к. минимальная сумма инвестиций обычно ограничена довольно высокой планкой, также берется комиссия за управление как процент от средств инвестора находящихся в управлении, так и процент от чистой прибыли, полученной от управления активами в течении определенного периода времени, обычно одного календарного года.

### 11.2.5.1 «Простые инвестиции»

«Простые инвестиции» – это инвестиционный сервис, предоставляемый компаниями ПАО «Сбербанк» и ООО УК «ФинЭкс Плюс» (Рисунок 54). Данный сервис предлагает услугу открытие и ведения индивидуального инвестиционного счета [17].



или волатильность доходности) или, что эквивалентно, минимизирует риск при заданной ожидаемой доходности» [18].

Но несмотря на то, что данный подход соответствует одному из подходов, используемых в предлагаемом продукте, непосредственно процесс выбора портфеля, его оптимизации по показателям риска и доходности остается вне компетенции инвестора. Ему / ей лишь предлагается пройти небольшое анкетирование, на основании которого доверительный управляющий сможет составить свое мнение о «толерантности риску» того или иного инвестора и подобрать оптимальный по его / ее мнению портфель ЦБ. Также, после создания портфеля, инвестор не сможет самостоятельно отслеживать эффективность портфеля в режиме реального времени, проводить его ребалансировку или ликвидацию – все это может сделать только доверительный управляющий, причем оперативность его / ее действий достаточно ограничена.

Более детальная критика теорий, базирующихся на модели MVO, представлена ниже, в разделе «Робо-советники».

### **11.2.6 Робо-советники**

Робо-советник – это программный продукт, интегрированный в решение инвестиционной компании по торговле ценными бумагами, и представляющий собой сервис рекомендаций по покупке и продаже ценных бумаг и формированию инвестиционного портфеля. Пример экранов работы робо-советника представлен ниже (Рисунок 55).

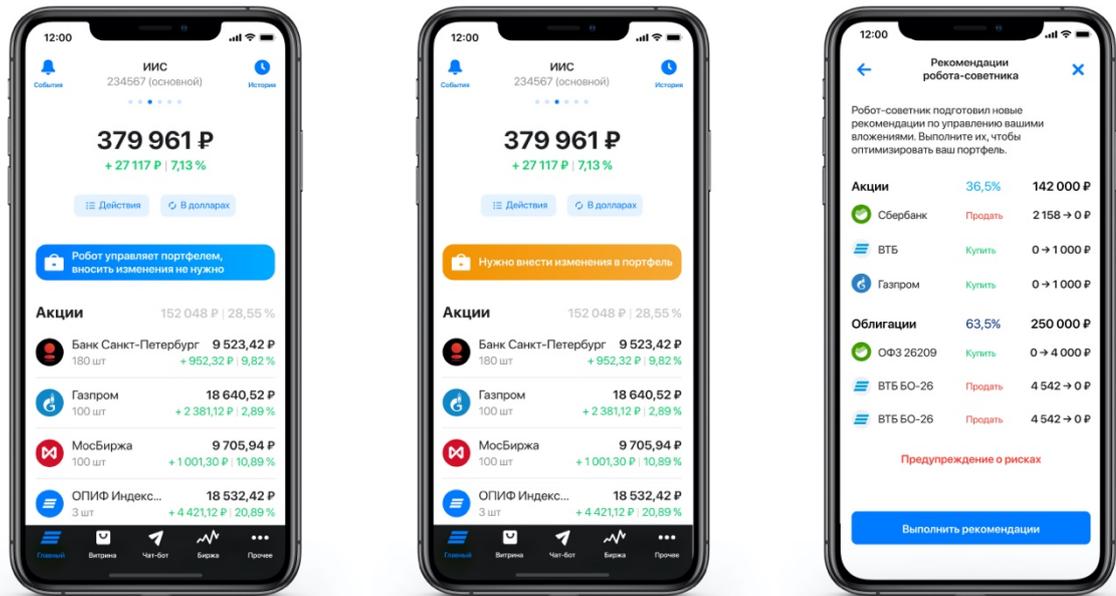


Рисунок 55 – Робо-советник «ВТБ-Инвестиции»

Перед началом использования робо-советника, пользователю необходимо ответить на несколько вопросов, на основе которых формируется метрика толерантности инвестора к риску. Хотя такая метрика является вполне зрелой и широко используемой мерой, алгоритм ее построения не раскрывается ни одной компанией. Вызывает скепсис факт того, что на основании ответов на 4-5 вопросов возможно построить адекватную данному инвестору функцию полезности, на основе ее построить кривые безразличия, характеризующие отношение инвестора к риску. Более того, если бы такой алгоритм применялся данными компаниям, то значение толерантности инвестора к риску было бы вполне конкретным, дискретным и выражалось цифрой, т.к. кривая безразличия касалась бы эффективного множества портфелей в одной единственной точке.

Предполагается, что качественные оценки толерантности риска, как-то «высокая» или «низкая» не несут большой практической ценности, т.к. не позволяют вести адекватный анализ портфелей. Как сравнить портфель с «высоким» и с «чуть более высоким» риском? Как сформировать портфель со степенью риска «на 24% выше среднего»? Где проходит граница между «средним» и «высоким» уровнем риска? В целом предполагается, что качественным характеристикам нет места в финансовом анализе, который

основан на таких областях научных знаний как статистика, теория вероятностей и математический анализ.

Кроме того, практически ни одна компания не упоминает на каких теориях или математических аппаратах базируются методы, применяемые робо-советниками, для оптимизации инвестиционных портфелей. Только на странице брокера ВТБ можно прочесть следующее «Стратегии робота созданы на основе классической теории по управлению портфелем от нобелевского лауреата Гарри Марковица» [19].

Данная теория (Современная Портфельная Теория) основана на методе «Оптимизация дисперсии среднего» или MVO (см. «Простые инвестиции»). Хотя данная теория и явилась фундаментом, на котором был построен весь современный инвестиционный анализ, в настоящее время данная теория применяется в основном или в научных кругах, или же как самый первый, «черновой» анализ инвестиционного портфеля, который легко позволяет визуализировать риск, доходность и их связь.

Одно из ограничений СПТ состоит в том, что она подразумевает следующее:

1. Стандартное отклонение доходностей ценных бумаг портфеля является истиной мерой риска;
2. Доходности ценных бумаг портфеля могут быть адекватно представлены нормальным распределением.

Т.е. СПТ использует такие меры риска и доходности, которые не всегда точно отражают объективную реальность. Так, использование нормального распределения подразумевает, что неопределенность получения дохода выше ожидаемого равна неопределенности получения дохода ниже ожидаемого. Также, СПТ оценивает ценные бумаги с доходностями, в которых количество положительных отклонений от среднего превышает количество отрицательных, как более рискованные чем они есть на самом деле. Обратное искажение характерно и для доходностей, где преобладает отрицательные доходности.

Интересно, что сам Г. Марковиц отмечал [6], что в некоторых случаях СПТ не точно отражает поведение и ожидания инвесторов и предполагал, что модель с использованием полудисперсии будет более предпочтительной. Но из-за сложности расчетов полудисперсии он все же взял за основу стандартное отклонение.

Кроме того, другой проблемой, отмеченной в работах [9] и [10] является высокая чувствительность модели к входным параметрам и в первую очередь к параметру ожидаемой доходности. Так, было показано, что при изменении доходности и риска на одинаковую величину, влияние доходности на состав портфеля оказывается в 10 раз более сильным. Другими словами, незначительные колебания в доходности ценных бумаг портфеля приводят к непропорционально большой, кардинальной ребалансировке портфеля.

Данный феномен ведет к росту транзакционных издержек для инвестора, а с учетом того, что данный сервис предоставляется компанией, которая зарабатывает на комиссионных отчислениях с каждой сделки, вопрос конфликта интересов становится особенно актуальным.

## 11.3 Сравнение существующих решений

В Таблица 3 представлен обобщенный сравнительный анализ рассмотренных решений.

Таблица 3. Сравнительный анализ существующих решений

Параметры сравнения	Наше решение	Сайты	Торговые терминалы	Мобильные приложения	Робо-советники	Доверительное управление
<b>Количественная оценка риска</b>	Да (по многим параметрам для неограниченного количества модельных портфелей)	Нет	Нет	Нет	Да <sup>1</sup> (качественная, непрозрачная)	Да
<b>География рынков</b>	Не ограничена	США	РФ	РФ	РФ	РФ
<b>Платформа</b>	Любая с доступом в интернет	Любая с доступом в интернет	ПК	Мобильный телефон	Мобильный телефон	Голосовая связь, электронная почта
<b>API</b>	Да	Нет	Да	Нет	Нет	Нет

<b>Конфликт интересов</b>	Нет	Нет	Нет	Да	Да	Да
<b>Доверие</b>	Низкое	Высокое	Очень высокое	Очень высокое	Очень высокое	Очень высокое
<b>Стоимость</b>	400 руб/мес.	14,99 USD/мес.	Мин сумма на счете, ~50 000 руб + комиссия с каждой торговой операции.	Бесплатно, но комиссия с каждой торговой операции.	Мин сумма на счете, ~50 000 руб + комиссия с каждой торговой операции.	Мин сумма на счете, более миллиона рублей, комиссия за управление портфелем

*Примечания:*

*1 – реальный торговый портфель;*

*2 – доступно только в платной версии;*

*3 – в терминах коэффициента покрытия долга.*

### **11.3.1 Результаты анализа рынка**

Рассмотрев существующие на рынке решения, как зарубежные, так и отечественные, можно сделать вывод, что на рынке отсутствуют простые, доступные широкому кругу инвесторов, в особенности розничных, инструменты для анализа риска инвестиционного портфеля, для оптимизации портфеля по совокупности параметров доходности и риска, для наглядного отображения степени риска портфеля в зависимости от того, какие бумаги и в каких пропорциях он содержит, т.е. для анализа риска инвестиций так, как риск понимается в экономике, финансах и в соответствии с современной портфельной теорией. Даже беглого взгляда на главные страницы сайтов брокеров, инвестиционных компаний и банков достаточно, чтобы понять, что диалог с инвестором ведется только в рамках потенциальной доходности, обещаний прибыльности инвестиций и лишь мелким шрифтом, в самом низу страницы говорится о рисках, но при этом ни сам термин «риск», ни то, какой смысл в него вкладывается той или иной компанией, не объясняется.

Поэтому становится очевидным, что программная реализация данного исследования имеет не только высокую практическую ценность, т.к. будет создан удобный и полезный инструмент для инвесторов, но и обладает значимой общественной ценностью – т.к. поможет значительно повысить финансовую грамотность инвесторов, настоящих и потенциальных, рассказав и показав второй, часто замалчиваемый или вскользь упоминаемый аспект инвестирования – риск.

### **11.4 Анализ целевой аудитории**

Количество индивидуальных инвесторов в 2020 году в России растет темпами, которые можно охарактеризовать как взрывные. Если на конец 2019 г. количество инвесторов в России составило около 2,5 миллиона человек, то в 2020 г. на рынок пришло около 4,7 миллиона человек – т.е. больше, чем за все предыдущие года вместе взятые [20].

Динамика изменения количества индивидуальных инвесторов в России представлен на Рисунок 56 [21].

### **Динамика количества индивидуальных инвесторов в России** тыс. человек на конец года

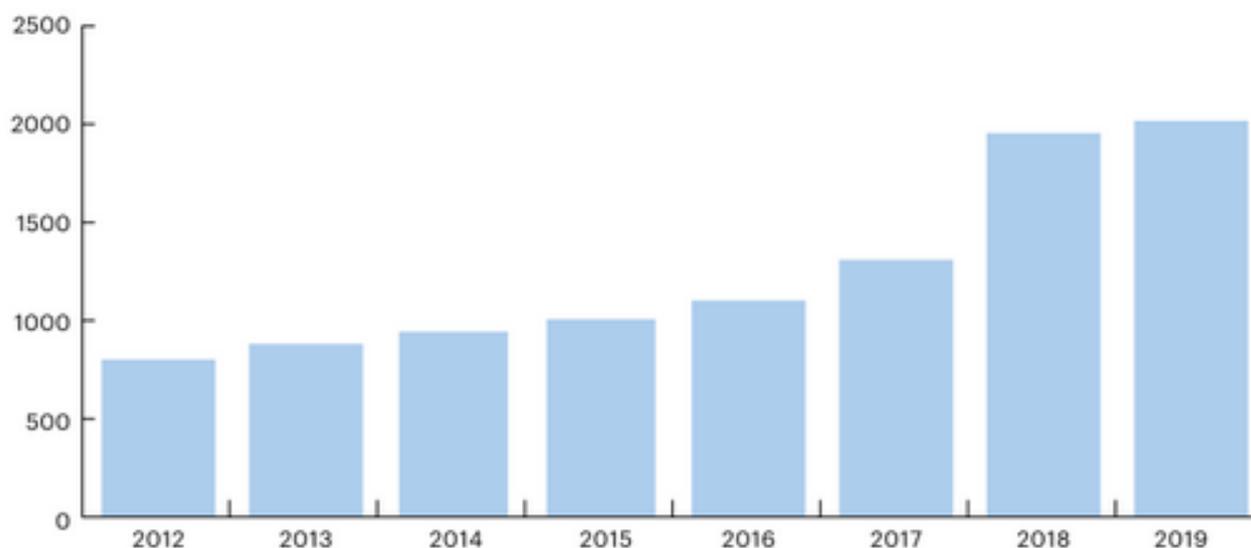


Рисунок 56 – Динамика количества индивидуальных инвесторов в России

Естественно, такие темпы роста не могут сохраняться долго и рано или поздно они должны замедлиться. Так, согласно оценке первого зампреда ЦБ РФ Сергея Швецова, рост количества индивидуальных инвесторов стабилизируется на отметке 10 миллионов человек [22].

Тем не менее, предполагается, что рост количества индивидуальных инвесторов в России продолжится, пусть и не столь высокими темпами. Если взять рынок США как идеальный, с точки зрения вовлечения домохозяйств в процесс инвестирования на фондовом рынке, то согласно последним оценкам, в США так или иначе инвестировали в различные инструменты финансовых рынков около 50 миллионов человек [23]. Это составляет около 15 процентов от населения США. Экстраполируя эти цифры на РФ, можно прийти к заключению, что число индивидуальных инвесторов на фондовом рынке может достигнуть около 20 миллионов человек.

Сводная качественная характеристика целевой аудитории продукта в РФ и США представлена в Таблица 4 [23], [24], [25] и [26].

Таблица 4. Сводная качественная характеристика целевой аудитории продукта в РФ и США

<b>Характеристика</b>	<b>Россия</b>	<b>США</b>
Количество, млн человек	≈ 8 (6% от населения страны)	≈ 50 (15% от населения страны)
Средний возраст, лет	≈ 35	≈ 51
Высшее образование, %	≈ 83	≈ 42
Средневзвешенный доход	Минимум 65 000 РУБ/мес. (≈ 200% от средней ЗП)	Минимум 9 000 USD/мес. (≈ 300% от средней ЗП)
Предпочитаемая стратегия	«buy and hold»	«buy and hold»
Предпочитаемые инструменты	Акции	Акции, ПИФ, ETF

Явно, российский рынок обладает значительным потенциалом роста как с точки зрения объема рынка (рост количества индивидуальных инвесторов), так и с точки зрения величины платежеспособного спроса (рост благосостояния индивидуальных инвесторов).

Примечательна разница в количестве инвесторов с высшим образованием и возрасте инвесторов – практически двукратная в обоих случаях.

Также, для обеих стран примечательно превышение среднего дохода инвестора над средним доходом по стране примерно в два раза, что подтверждает факт того, что независимо от того на какой ступени развития находится общество в плане культуры инвестирования, в основном это делают люди с высоким доходом – высококвалифицированные специалисты, руководители, предприниматели и тд.

### **11.4.1 Портрет потребителя**

Таким образом, выполнив анализ рынка, становится возможным составить портрет потенциального потребителя продукта, который необходимо будет учесть в стратегии продвижения продукта. Причем для разных рынков портрет будет различать. Условно, рынки можно разделить на два:

1. Развивающийся (Россия).
2. Развитый (США).

#### **11.4.1.1 Развивающийся рынок**

На развивающемся рынке индивидуальный инвестор – это мужчина 35 лет с высшим образованием, занимает среднюю руководящую должность или является высококвалифицированным специалистом с заработной выше средней по стране. Предпочитаемые инструменты инвестирования – акции, предпочитаемая инвестиционная стратегия – долгосрочная «купи и держи» с целью сохранения капитала на крупные покупки и образование детей.

#### **11.4.1.2 Развитый рынок**

На развитом рынке индивидуальный инвестор – это мужчина или женщина, возрастом более 50 лет, занимающий высокую руководящую должность, чрезвычайно высококвалифицированный специалист или владелец бизнеса с зарплатой значительно выше средней. Предпочитаемые инструменты инвестирования – акции и фонды акций, предпочитаемая инвестиционная стратегия – долгосрочная «купи и держи» с целью сохранения капитала для обеспечения требуемого уровня жизни в период пенсии.

### **11.5 Бизнес-модель**

Бизнес-модель – это последовательное, непротиворечивое и схематическое описание проекта как отдельного бизнеса, призванное помочь в оценке ключевых факторов успеха компании.

Для построения бизнес-модели стартапа, использована методология А. Остервальдера и И. Пинье [27]. Основным инструментом построения бизнес-модели в рамках данной методологии является шаблон или макет. Это – универсальный язык описания, представления, анализа и преобразования бизнес-моделей. Шаблон состоит из 9 блоков, позволяющих просто и наглядно описать и представить деятельность организации на одном листе.

В Таблица 5 представлена бизнес-модель стартапа, выполненная в рамках данной методологии.

Эффективность бизнес-модели заключается в следующих блоках:

1. Потребительские сегменты, сегментированы по профилям потребителя – продукт при единообразном функционале может быть доступен как индивидуальным инвесторам, которые пользуются продуктом с помощью клиентского приложения, так и юридическим лицам, которые могут интегрировать решения, предоставляемые продуктом, в свои системы с помощью API продукта.

2. Также, для продукта характерна чрезвычайно низкая зависимость от партнеров, в данном случае это поставщик облачных вычислительных ресурсов. Однако, даже данное незначительное ограничение было учтено при разработке продукта для того, чтобы максимально избежать ситуации «vendor lock-in». Для этого для конфигурирования облачных ресурсов применяется подход «Инфраструктура как код», при помощи которого вся инфраструктура описана в коде и таким образом переход на другую облачную платформу, при необходимости, займет несколько минут, а не дней.

3. Эффективный канал сбыта – прямые продажи через Интернет, без привлечения посредников и минимальными комиссионными отчислениями за эквайринг.

Однако стоит отметить зависимость от ключевых ресурсов – высококвалифицированных специалистов, разработчиков продукта. Удержание и мотивацию планируется осуществлять высокой заработной платой и опционами на акционерный капитал компании для наиболее важных сотрудников.

Таблица 5. Бизнес-модель проекта

<p><b>Ключевые партнеры</b></p> <p>Поставщики: Поставщики облачных вычислительных ресурсов.</p>	<p><b>Ключевые виды деятельности</b></p> <p>Решение проблемы: Оптимизация портфеля ценных бумаг в соответствии с заданными параметрами риска/доходности.</p>	<p><b>Ценностные предложения</b></p> <p>1. Простой и в тоже время мощный аналитический функционал продукта 2. Новизна 3. Социальная функция – повышение финансовой грамотности 4. Демократизация продвинутых алгоритмов оценки риска.</p>	<p><b>Взаимоотношения с клиентами</b></p> <p>Сегментированное: 1. Онлайн сообщество для бесплатных пользователей; 2. Электронная почта для платных пользователей.</p>	<p><b>Потребительские сегменты</b></p> <p>Многопрофильное сегментирование: B2B и B2C Выделены следующие сегменты: 1. Индивидуальные инвесторы; 2. Брокерские компании, банки и другие юридические лица, занимающиеся инвестиционной деятельностью.</p>
	<p><b>Ключевые ресурсы</b></p> <p>Интеллектуальные ресурсы – алгоритмы, реализующие ценностное предложение; Персонал – высококвалифицированные специалисты, работающие на стыке математики, статистики, программирования.</p>		<p><b>Каналы сбыта</b></p> <p>Прямые: 1. Продажа через интернет</p>	
<p><b>Структура издержек</b></p> <p>Фиксированные издержки – заработная плата работникам, налоги. Переменные издержки – облачные вычислительные ресурсы, продвижение, организационные расходы (ведение счетов, бухгалтерия, юридические услуги).</p>		<p><b>Потоки поступления доходов</b></p> <p>1. Регулярный доход от периодических платежей, получаемых от пользователей, оформивших подписку. Увеличение дохода за счет: 1. Масштабирования продукта на другие географические рынки. <i>Механизм ценообразования</i> Фиксированная цена на подписку для: 1. Физических лиц; 2. Юридических лиц.</p>		

Форма бизнеса в соответствии с моделью А. Остервальдера и И. Пинье – «Иновационные продукты»:

Экономика – как было показано в анализе рынка и анализе конкурентов, в настоящее время на рынке отсутствуют решения, обеспечивающие, заявленный в продукте функционал. Необходимо сосредоточиться на скорейшем выходе продукта на рынок с целью занятия большей доли.

Культура – как было показано в бизнес-модели проекта, секция «Ключевые ресурсы», персонал является одним из ключевых ресурсов функционирования продукта. Для его реализации требуются высококвалифицированные специалисты, обладающие знаниями в нескольких дисциплинах и имеющие возможность и способность применять их.

Конкуренция – как следует из раздела «Культура», конкурентные преимущества компании приобретаются за счет увеличения функционала продукта, качества взаимодействия с ним и точности результатов его работы. Это в свою очередь реализуется сотрудниками предприятия, таким образом для компании характерна ориентация на сотрудников и повышенное внимание к одаренным личностям.

## **11.6 Оценка экономической целесообразности бизнеса**

Оценка экономической целесообразности стартапа выполнялась методом дисконтирования денежных потоков предприятия с учетом определенных допущений.

Основное допущение расчета – выход только на рынок РФ, хотя задача перевода интерфейса на другие языки, и в первую очередь английский, представляется тривиальной, а данные со всех мировых бирж доступны в реальном времени как часть оформленной платной подписки.

Таким образом TAM (Total Available Market) в нашем расчете составляет 8 000 000 человек и SOM (Service Obtainable Market) – 0.5% от TAM.

Предполагается, что в течение 10 лет количество индивидуальных инвесторов в РФ достигнет прогнозного значения в 20 миллионов человек, что предполагает темпы роста около 10% в год. Исходя из этого, данное значение использовано как единая ставка темпов роста для всех остальных статей расчета.

Средневзвешенная стоимость капитала взята по ставке 20%, причем используется только ставка доходности акционерного капитала предприятия, заемный капитал не используется.

Результаты расчета представлены в Таблица 6.

Таблица 6. Оценка стоимости предприятия методов DCF

	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
<b>Общий объем рынка</b>	<b>8 000 000</b>	<b>8 800 000</b>	<b>9 680 000</b>	<b>10 648 000</b>	<b>11 712 800</b>	<b>12 884 080</b>	<b>14 172 488</b>	<b>15 589 737</b>	<b>17 148 710</b>	<b>18 863 582</b>	<b>20 749 940</b>
Пользовательская база	0	0	48 400	53 240	58 564	64 420	70 862	77 949	85 744	94 318	103 750
Платные пользователи	0	0	4 840	5 324	5 856	6 442	7 086	7 795	8 574	9 432	10 375
Стоимость подписки, руб./мес	400	400	400	400	400	400	400	400	400	400	400
<b>Выручка, руб./год</b>	<b>0</b>	<b>0</b>	<b>23 232 000</b>	<b>25 555 200</b>	<b>28 110 720</b>	<b>30 921 792</b>	<b>34 013 971</b>	<b>37 415 368</b>	<b>41 156 905</b>	<b>45 272 596</b>	<b>49 799 855</b>
<b>Затраты всего, руб</b>	<b>360 000</b>	<b>794 000</b>	<b>885 720</b>	<b>974 292</b>	<b>1 071 721</b>	<b>1 178 893</b>	<b>1 296 783</b>	<b>1 426 461</b>	<b>1 569 107</b>	<b>1 726 018</b>	<b>1 898 619</b>
<i>Облачная инфраструктура</i>	360 000	396 000	435 600	479 160	527 076	579 784	637 762	701 538	771 692	848 861	933 747
<i>Продвижение</i>	0	198 000	217 800	239 580	263 538	289 892	318 881	350 769	385 846	424 431	466 874
<i>Организационные расходы</i>	0	200 000	232 320	255 552	281 107	309 218	340 140	374 154	411 569	452 726	497 999
<b>ФОТ, руб.</b>	<b>9 000 000</b>	<b>9 900 000</b>	<b>10 890 000</b>	<b>11 979 000</b>	<b>13 176 900</b>	<b>14 494 590</b>	<b>15 944 049</b>	<b>17 538 454</b>	<b>19 292 299</b>	<b>21 221 529</b>	<b>23 343 682</b>
<i>Разработчик</i>	1 800 000	1 980 000	2 178 000	2 395 800	2 635 380	2 898 918	3 188 810	3 507 691	3 858 460	4 244 306	4 668 736
<i>Разработчик</i>	1 800 000	1 980 000	2 178 000	2 395 800	2 635 380	2 898 918	3 188 810	3 507 691	3 858 460	4 244 306	4 668 736
<i>Инвестиционный аналитик</i>	1 800 000	1 980 000	2 178 000	2 395 800	2 635 380	2 898 918	3 188 810	3 507 691	3 858 460	4 244 306	4 668 736
<i>DevOps / DBA</i>	1 800 000	1 980 000	2 178 000	2 395 800	2 635 380	2 898 918	3 188 810	3 507 691	3 858 460	4 244 306	4 668 736
<i>Дизайнер</i>	1 800 000	1 980 000	2 178 000	2 395 800	2 635 380	2 898 918	3 188 810	3 507 691	3 858 460	4 244 306	4 668 736
<b>ЕВТДА</b>	<b>-9 360 000</b>	<b>-10 694 000</b>	<b>11 456 280</b>	<b>12 601 908</b>	<b>13 862 099</b>	<b>15 248 309</b>	<b>16 773 140</b>	<b>18 450 454</b>	<b>20 295 499</b>	<b>22 325 049</b>	<b>24 557 554</b>
Процентные платежи	0	0	0	0	0	0	0	0	0	0	0
Амортизация	0	0	0	0	0	0	0	0	0	0	0
<b>Прибыль</b>	<b>-9 360 000</b>	<b>-10 694 000</b>	<b>11 456 280</b>	<b>12 601 908</b>	<b>13 862 099</b>	<b>15 248 309</b>	<b>16 773 140</b>	<b>18 450 454</b>	<b>20 295 499</b>	<b>22 325 049</b>	<b>24 557 554</b>
<b>Ставка налога на прибыль</b>	<b>0%</b>	<b>24%</b>	<b>24%</b>	<b>24%</b>	<b>24%</b>	<b>24%</b>	<b>24%</b>	<b>24%</b>	<b>24%</b>	<b>24%</b>	<b>24%</b>
<b>Чистая прибыль</b>	<b>-9 360 000</b>	<b>-8 127 440</b>	<b>8 706 773</b>	<b>9 577 450</b>	<b>10 535 195</b>	<b>11 588 715</b>	<b>12 747 586</b>	<b>14 022 345</b>	<b>15 424 579</b>	<b>16 967 037</b>	<b>18 663 741</b>
WACC	20.00%										
Дисконтированный денежный поток	-9 360 000	-6 772 867	6 046 370	5 542 506	5 080 630	4 657 244	4 269 141	3 913 379	3 587 264	3 288 325	3 014 298
Терминальные темпы роста		Долгосрочные темпы роста мировой экономики									4.00%

Терминальная налоговая ставка	Номинальная ставка налога на прибыль для российских компаний	24%
Терминальный WACC		10.00%
Терминальная стоимость		323 504 840
Приведенная стоиомсть	52 247 838	
<hr/>		
<b>Стоимость предприятия, руб.</b>	<b>75 514 129</b>	

### **11.6.1 Расчет себестоимости продукта**

Расчет себестоимости продукта представлен в Таблица 7, исходя из предположений и допущений, сделанных при оценке бизнеса.

Стоит отметить уникальную масштабируемость данного решения – прирост количества пользователей вызовет линейный рост прямых затрат, связанных с приобретением дополнительных вычислительных ресурсов

Темпы роста затрат на ФОТ будут гораздо ниже, возможно вообще нулевыми, т.к. функциональность продукту добавляется лишь единожды и затем становится доступной неограниченному количеству пользователей.

Таблица 7. Расчет себестоимости продукта

	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
<b>Общий объем рынка</b>	8 000 000	8 800 000	9 680 000	10 648 000	11 712 800	12 884 080	14 172 488	15 589 737	17 148 710	18 863 582	20 749 940
Пользовательская база	0	0	48 400	53 240	58 564	64 420	70 862	77 949	85 744	94 318	103 750
Платные пользователи	0	0	4 840	5 324	5 856	6 442	7 086	7 795	8 574	9 432	10 375
Стоимость подписки, руб./год	4800	4800	4800	4800	4800	4800	4800	4800	4800	4800	4800
<b>Выручка, руб./год</b>	0	0	23 232 000	25 555 200	28 110 720	30 921 792	34 013 971	37 415 368	41 156 905	45 272 596	49 799 855
<b>Затраты всего, руб</b>	360 000	794 000	885 720	974 292	1 071 721	1 178 893	1 296 783	1 426 461	1 569 107	1 726 018	1 898 619
<i>Облачная инфраструктура</i>	360 000	396 000	435 600	479 160	527 076	579 784	637 762	701 538	771 692	848 861	933 747
<i>Продвижение</i>	0	198 000	217 800	239 580	263 538	289 892	318 881	350 769	385 846	424 431	466 874
<i>Организационные расходы</i>	0	200 000	232 320	255 552	281 107	309 218	340 140	374 154	411 569	452 726	497 999
<b>ФОТ, руб.</b>	9 000 000	9 900 000	10 890 000	11 979 000	13 176 900	14 494 590	15 944 049	17 538 454	19 292 299	21 221 529	23 343 682
Процентные платежи	0	0	0	0	0	0	0	0	0	0	0
Амортизация	0	0	0	0	0	0	0	0	0	0	0
<b>Прибыль</b>	-9 360 000	-10 694 000	11 456 280	12 601 908	13 862 099	15 248 309	16 773 140	18 450 454	20 295 499	22 325 049	24 557 554
Ставка налога на прибыль	0%	24%	24%	24%	24%	24%	24%	24%	24%	24%	24%
<b>Чистая прибыль</b>	-9 360 000	-8 127 440	8 706 773	9 577 450	10 535 195	11 588 715	12 747 586	14 022 345	15 424 579	16 967 037	18 663 741
<b>Прямые затраты, руб./пользователь/год</b>			183	183	183	183	183	183	183	183	183
<b>ФОТ, руб./пользователь/год</b>			2 250	2 250	2 250	2 250	2 250	2 250	2 250	2 250	2 250
<b>Себестоимость продукта</b>			2 433	2 433	2 433	2 433	2 433	2 433	2 433	2 433	2 433
<b>Маржинальность</b>			97%	97%	97%	97%	97%	97%	97%	97%	97%

## 11.7 Стратегия продвижения

Стратегия продвижения – это позиционирование торговой марки и создание системы интегрированных маркетинговых коммуникаций. Стратегия продвижения продукта предполагает разработку комплекса мер по созданию и укреплению отношения потребителя к продукции компании.

Стратегия продвижения продукции предполагает создание следующих основных методов продвижения продукции:

- информирование потребителей о характеристиках продукта;
- формирование узнаваемого образа продукта;
- поддержание популярности продукта;
- убеждение потребителей в ценности продукта.

Существуют такие основные стратегии продвижения товара на рынок:

- Толкай (от англ. «Push») – используется в случае реализации продукта третьими лицами, стратегия направлена на «выталкивания» продукта на рынок данными третьим лицами.

- Тяни (от англ. «Pull») – используется для работы с конечным потребителем продукта, стратегия направлена на «притягивание» потребителя к продукту.

Т.к. продукт будет реализовывать напрямую конечному потребителю, то для его продвижения планируется использовать стратегию «Тяни». Данная стратегия означает активную рекламную и промо-кампанию, направленную через средства массовой информации на конечного потребителя.

Данную стратегию планируется реализовать в несколько этапов, с использованием различных тактик на разных этапах жизни продукта.

### 11.7.1 Сарафанное радио

Тактика «из уст в уста» или «сарафанное радио» – это неформальные формы и способы передачи устной информации о товаре или услуге от потребителя к потребителю. По сути, это бесплатная форма устной или

письменной рекламы, с помощью которой удовлетворённые пользователи рассказывают другим людям, почему им нравится какой-либо товар или услуга.

Данную тактику планируется применять на первой стадии запуска продукта – закрытом и открытом бета-тестировании продукта.

Во время закрытого бета-тестирования продукта пользователи будут приглашаться непосредственно создателями продукта и одновременно будет проводиться тестирование основного функционала продукта.

Во время открытого бета-тестирования, у существующих пользователей будет возможность пригласить новых пользователей, но не более 2-3 человек, реализуя тем самым подход «дефицит-маркетинга». Одновременно будет выполняться нагрузочное тестирование системы для определения узких мест при одновременном использовании продукта большим количеством пользователей.

## **11.7.2   Онлайн маркетинг**

Данная тактика будет использоваться, когда продукт перейдет в фазу коммерческой эксплуатации. Данная тактика выбрана на основе изучения целевой аудитории – молодые специалисты с высшим образованием, для которых характерно использование сети Интернет для получения информации. При этом послание рекламных сообщений будет также основано на предпочтениях целевой аудитории – это оценка риска инвестиций, минимизация риска с целью сохранения капитала, защита инвестиций от резких рыночных колебаний.

Данная тактика будет использоваться совместно с тактикой входящего маркетинга – т.е. привлечении только тех потребителей, которым продукт необходим, с целью сокращения затрат.

### **11.7.2.1   Таргетирования реклама**

Планируется использовать таргетированную рекламу в результатах выдачи поисковых запросов по теме, соответствующей теме продукта – «инвестиции», «оценка риска» и т.д.

### **11.7.2.2 Контекстная реклама**

Планируется использовать контекстную рекламу в онлайн-сми профильного направления, освещающие экономические новости в целом, так и инвестиционную деятельность в частности.

Также, планируется размещение контекстной рекламы в профильным онлайн сообществах – как в социальных сетях, так и на самостоятельных интернет-площадках.

### **11.7.2.3 Социальный маркетинг**

Социальный маркетинг – это направление маркетинга, использующее инструменты маркетинга для улучшения жизни, как отдельных людей, так и всего общества в целом. Также эту тактику можно назвать «косвенным маркетингом», т.е. маркетингом, где рекламируемый продукт не является основным сообщением рекламного текста.

Эту тактику планируется реализовать публикацией статей в социальных сетях, направленных на повышение финансовой грамотности населения, освещающих различные аспекты инвестиционной деятельности и в первую очередь риска инвестиций, использования инструментов, позволяющих оценить риск инвестиций, формирования стратегий, позволяющих снизить риск инвестиций и т.д. В этих статьях продукт будет упоминаться как инструмент, такой же как «Excel» или калькулятор, помогающий абстрагировать сложную математику и выдающий уже готовый результат.

## **11.8 Защита интеллектуальной собственности**

В рамках реализации проекта планируется выполнить следующие мероприятия по охране прав на интеллектуальную собственность:

Подать заявку на регистрацию результатов интеллектуальной деятельности – программу для ЭВМ в федеральный орган исполнительной власти по интеллектуальной собственности Российской Федерации – Роспатент.

Права на результаты интеллектуальной деятельности будут оформлены на юридическое лицо, созданное для разработки и коммерческого внедрения предлагаемого продукта.

## **11.9 Выводы по главе**

Таким образом, в рамках данного исследования было выполнено следующее:

1. Анализ существующих решений;
2. конкурентный анализ;
3. анализ рынка и исследование целевой аудитории;
4. построена бизнес-модель стартапа;
5. выполнены экономические расчеты, моделирующие деятельность стартапа при определенных допущениях.

Исследование показало, что на рынке в настоящее время отсутствуют продукты, решающие аналогичные задачи в требуемом объеме с получением аналогичных результатов.

Анализ рынка показывает присутствие потенциально большого рынка, испытывающего спрос на подобные решения.

Экономическое моделирование деятельности предприятия показывает возможность получения значительной экономической выгоды, при этом потенциал ее роста представляется довольно существенным при масштабировании бизнеса на другие географические регионы.

Таким образом, данное предприятие является экономически целесообразным и в настоящее время ведутся работы по его запуску.

## **12. Социальная ответственность**

### **12.1 Введение**

Рабочее место разработчика и пользователя продукта представляет собой офисное помещение с освещением смешанного типа, с кабинетным типом планирования, при этом в одном кабинете может находиться несколько рабочих мест. Каждое рабочее место оборудовано столом, на котором находится монитор, клавиатура, мышь, канцелярские принадлежности и креслом.

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

**Специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства**

В соответствии с ТК РФ от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018) [28] предусмотрена рациональная организация труда в течение смены, которая обеспечивает:

- 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.

Кроме того, предусмотрен предварительный медосмотр сотрудника при приеме на работу и периодические медосмотры.

Также, перед приемом на работу каждый сотрудник должен пройти инструктаж по технике безопасности и охране труда.

При организации рабочей зоны сотрудника основной целью является обеспечение качественного и эффективного выполнения работы при эффективном использовании оборудования.

#### **Организационные мероприятия при компоновке рабочей зоны**

Рабочее место – это часть пространства, в котором инженер осуществляет трудовую деятельность, и проводит большую часть рабочего времени. Рабочее

место, хорошо приспособленное к трудовой деятельности инженера, правильно и целесообразно организованное, в отношении пространства, формы, размера обеспечивает ему удобное положение при работе и высокую производительность труда при наименьшем физическом и психическом напряжении.

Применительно к объекту исследования данной работы действуют следующие нормативные документы в части организации рабочей зоны: ГОСТ 12.2.032-78 ССБТ [29], ГОСТ 21889-76 [30], ГОСТ 22269-76 [31], ГОСТ Р ИСО 9241-4-2009 [32], ГОСТ Р ИСО 9241-5-2009 [33].

В соответствии с этими документами требования к компоновке рабочей зоны заключаются в следующем:

Согласно ГОСТ 12.2.032-78 конструкция рабочего места и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. Большое значение имеет также характер работы. В частности, при организации рабочего места программиста должны быть соблюдены следующие основные условия:

1. Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы [31], как то:

а. Рабочее пространство:

- ширина не менее 700 мм;
- глубина не менее 400 мм;
- высота рабочей поверхности стола над полом 700 – 750 мм.

б. Стол:

- высота 710 мм;
- длина стола 1300 мм;
- ширина стола 650 мм.
- поверхность для письма должна иметь не менее 40 мм в глубину и не менее 600 мм в ширину.

- в. Под рабочей поверхностью должно быть предусмотрено пространство для ног:
- высота не менее 600 мм;
  - ширина не менее 500 мм;
  - глубина не менее 400 мм.
- г. Кресло:
- допускать возможность изменения положения тела, т.е. обеспечивать свободное перемещение корпуса и конечностей тела друг относительно друга;
  - допускать регулирование высоты в зависимости от роста работающего человека (в пределах от 400 до 550 мм );
  - иметь слегка вогнутую поверхность;
  - иметь небольшой наклон назад.

2. Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений для эксплуатации и технического обслуживания оборудования [29].

3. При расположении элементов рабочего места должны быть предусмотрены необходимые средства защиты оператора от воздействия опасных и вредных факторов, а также условия для его экстренного ухода с рабочего места [31].

4. Клавиатура офисного компьютера по своей конструкции должна соответствовать требованиям эргономики. Кроме того, необходимо учитывать возможности использования клавиатуры в условиях ограниченных возможностей пользователей и различий в уровне их квалификации [32].

5. Установлении требований к конфигурации рабочих станций, направленных на поддержание и улучшение функционирования терминала и обеспечение комфортности работы пользователей с минимальным риском для их безопасности и здоровья. Пользователи в офисной работе обычно принимают определенные рабочие позы (сидят наклонив корпус вперед, держат корпус прямым, откидываются назад, стоят, или комбинируют эти позы). Рабочее место,

адаптированное к предпочтениям пользователя, может способствовать его перемещениям, поддерживать комфортность работы и уменьшать физические, умственные и зрительные нагрузки [33].

На Рисунок 57 и Рисунок 58 представлены общие требования к организации рабочей зоны. На Рисунок 57 представлена оптимальная поза оператора ЭВМ, а на Рисунок 58 дана примерная организация рабочей зоны, которая создана исходя из того, что средняя зона охвата рук человека составляет около 40 – 50 см. Ближней зоне соответствует область, охватываемая рукой с прижатым к туловищу локтем, дальней зоне – область вытянутой руки.

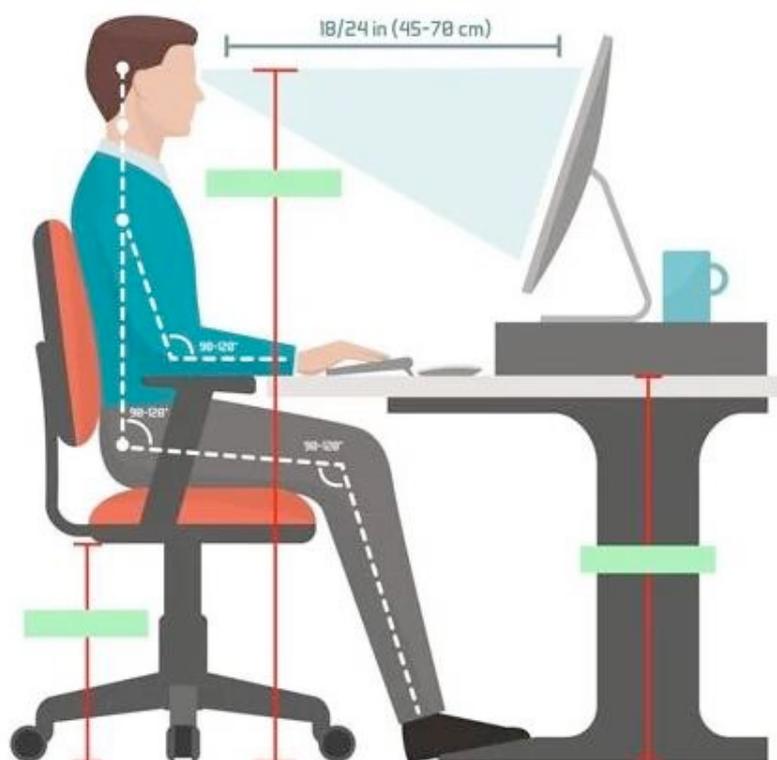


Рисунок 57 – Организация рабочего места - вид сбоку



Рисунок 58 – Организация рабочей зоны - вид сверху

## 12.3 Производственная безопасность

### Анализ выявленных вредных и опасных факторов

При работе оператора ЭВМ могут возникнуть вредные и опасные факторы. По ГОСТ 12.0.003-2015 2015 [34] выделен ряд факторов, которые приведены в Таблица 8, также в ней представлены нормативные документы, регламентирующие их действие.

Таблица 8. Опасные и вредные факторы

Факторы	Этапы работ		Нормативные документы
	Проектирование	Разработка	
- аномальные микроклиматические параметры воздушной среды			СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи". СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений» ГОСТ 12.1.005-88 «Общие санитарно-гигиенические требования к воздуху рабочей зоны»
- отсутствие или недостатки необходимого искусственного освещения	+	+	СП 52.13330.2016 «Естественное и искусственное Освещение»
- физические статические перегрузки, связанные с рабочей позой	+	+	Эргономические требования к проведению офисных работ с использованием

			видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора - ГОСТ Р ИСО 9241-5-2009
- умственное перенапряжение, в том числе вызванное информационной нагрузкой	+	+	Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4 - ГОСТ Р ИСО 9241-4-2009
- повышенный уровень и другие неблагоприятные характеристики шума	+	+	Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки - СН 2.2.4/2.1.8.562-96
- разница электрических потенциалов, под действие которого попадает работающий	+	+	Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов - ГОСТ 12.1.038-82 ССБТ

### **Обоснование мероприятий по снижению воздействия вредных и опасных факторов**

#### **Аномальные микроклиматические параметры воздушной среды**

Сотрудник постоянно находится в процессе взаимодействия с окружающей его рабочее место средой. Температура, относительная влажность и скорость движения воздуха характеризуют условия данной окружающей среды.

В соответствии с СП 2.4.3648-20 [35] в помещениях, в которых работа с использованием ЭВМ является основной и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные параметры микроклимата согласно действующим санитарно-эпидемиологическим нормам.

В соответствии с СанПиН 2.2.4.548-96 [36], показателями, характеризующими микроклимат в производственных помещениях, являются:

- температура воздуха;
- температура поверхностей;
- относительная влажность воздуха;
- скорость движения воздуха;
- интенсивность теплового облучения.

В Таблица 9 приведены оптимальные величины показателей микроклимата в рабочих зонах производственных помещений согласно ГОСТ 12.1.005-88 [37]

для оператора ЭВМ, чья категория работ относится к категории Ia - легкие физические работы согласно ГОСТ 12.1.005-88 [37]

Таблица 9. Оптимальные величины показателей микроклимата в рабочих зонах производственных помещений

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

Исходя из требований, которые предъявляют указанные нормативные документы к используемым производственным помещениям, в них должна поддерживаться температура воздуха 22 – 25 С°, при относительной влажности 40–60%. Для этого в помещении необходимо проводить ежедневную влажную уборку и систематическое проветривание после каждого часа работы с ЭВМ. Кроме этого, в теплое время года в помещении должна функционировать система принудительной приточно-вытяжной вентиляции. В зимнее время года в помещении должна функционировать система отопления со встроенными нагревательными элементами и терморегуляторами.

## **12.4 Отсутствие или недостаток необходимого естественного и искусственного освещения**

Естественное и искусственное освещение рабочего места оказывает влияние на физическое состояние и на работу сотрудника. Недостаточный уровень освещенности в помещении приводит к снижению остроты зрения, головным болям, снижению концентрации внимания и, как следствие, к ухудшению производительности труда, совершению ошибок и в итоге – к производственным травмам и несчастным случаям [38].

Причиной недостаточной освещенности рабочего места являются недостаточность естественного освещения, недостаточность искусственного освещения и пониженная контрастность.

Рабочее помещение имеет как естественное, так и искусственное освещение. Коэффициент естественного освещения должен составлять не менее

1,2%, а освещенность на поверхности рабочего стола в зоне размещения документов либо других источников информации для данного типа помещения – 400 лк, т.к. более яркое освещение будет напрягать глаза и создавать блики на экране монитора согласно СП 52.13330.2016 [38], что оказывает значительную нагрузку на глаза сотрудника.

Искусственное освещение в помещениях должно осуществляться системой равномерного освещения. Помимо этого, рабочие места следует размещать так, чтобы естественный свет падал преимущественно слева, а дисплей монитора были ориентированы боковой стороной к световым проемам.

### **Расчет искусственного освещения.**

Задачей расчета искусственного освещения помещения является выполнение общего освещения данного производственного помещения.

Для выполнения данного расчета необходимо воспользоваться методом коэффициента использования светового потока согласно СП 52.13330.2016 «Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95\*» [38].

1. В первую очередь необходимо задать исходные данные для расчёта:
  - нормированная освещенность в помещении  $E_H = 400$  лк, согласно СП 52.13330.2016 для объекта «Помещения для работы с дисплеями и видеотерминалами, залы ЭВМ» [38];
  - длина помещения  $L = 6$  м;
  - ширина помещения  $W = 3$  м;
  - высота помещения  $H = 4$  м;
  - высота рабочей поверхности  $H_{rp} = 0,8$  м.

Световой поток лампы определяется по формуле:

$$\Phi = \frac{E_H \cdot S \cdot K \cdot Z}{N \cdot \eta}$$

- коэффициент отражения стен  $R_c = 50$  %;
- коэффициент отражения потолка  $R_n = 70$  %;
- коэффициент запаса  $k = 1,8$ ;

- коэффициент неравномерности  $Z = 1,1$ .

2. Затем необходимо рассчитать систему общего люминесцентного освещения. Для этого выберем светильники типа ОД с  $\lambda = 1,4$ .

Приняв  $h_c = 0,5$  м, получаем:

$$h = 4 - 0,5 - 0,8 = 2,7 \text{ м};$$

В кабинете светильники типа ОД установлены в один ряд – 2 светильника в ряду. Учитывая, что в каждом светильнике установлено две лампы, общее число люминесцентных ламп ЛТБ в помещении  $N = 4$ .

3. Теперь необходимо рассчитать показатель помещения.

$$i = \frac{A \cdot B}{h \cdot (A + B)} = \frac{6 \cdot 3}{2,7 \cdot (6 + 3)} = 0,7$$

В соответствии с СП 52.13330.2016 «Естественное и искусственное освещение» определен коэффициент использования светового потока  $\eta = 0,42$ .

Необходимый световой поток лампы определяется по формуле:

$$\Phi = \frac{E_n \cdot S \cdot K \cdot Z}{N \cdot \eta} = \frac{400 \cdot 18 \cdot 1,8 \cdot 1,1}{4 \cdot 0,42} = 8485$$

4. Теперь определим потребный световой поток лампы в каждом из рядов, в данном случае ближайшая стандартная лампа – это лампа ЛТБ, 125 Вт с потоком 8150 лм. Делаем проверку выполнения условия:

$$-10 \% \leq (\Phi_{ст} - \Phi_{расч}) / \Phi_{ст} * 100 \% \leq 20 \% \quad (6.3)$$

$$-10 \% \leq (8150 - 8485) / 8150 * 100 \% \leq 20 \%$$

$$-10 \% \leq -4,1 \% \leq 20 \%$$

Определяем электрическую мощность осветительной установки

$$P = 4 * 125 = 500 \text{ Вт}.$$

В результате анализа освещенности рабочего места отклонений от норм выявлено не было. Уровень освещенности соответствует нормам. Использованное кабинета банка полностью соответствует требованиям СП 52.13330.2016 [38] к организации освещения рабочего места.

## 12.5 Акустические колебания в производственной среде

При работе ЭВМ источниками таких шумов выступают акустические колебания, создаваемые работающими частыми ЭВМ, как-то:

- Блок питания;
- Охлаждающий вентилятор системного процессора.

Воздействие шума может привести к ухудшению слуха и как следствие – невозможность сотрудника полноценно выполнять свои обязанности. Кроме этого, шумовое загрязнение в рабочей зоне приводит к снижению внимания оператора, замедлению скорости психических реакций. Шумовой фон в помещении возникает из-за работы компьютеров, а также из-за работы принтеров, телефонов и систем вентиляции.

В Таблица 10 приведены предельно допустимые уровни звукового давления для рассматриваемого вида трудовой деятельности в соответствии с СанПиН 2.2.4.3359-16 « Санитарно-эпидемиологические требования к физическим факторам на рабочих местах » [36].

Таблица 10. Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для основных наиболее типичных видов трудовой деятельности и рабочих мест.

Вид трудовой деятельности, рабочее место	Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентные уровни звука (дБА)
	31,5	63	125	250	500	1000	2000	4000	8000	
Высококвалифицированная работа, требующая сосредоточенности, административно-управленческая деятельность, измерительные и аналитические работы в лаборатории; рабочие места в помещениях цехового управленческого аппарата, в рабочих комнатах конторских помещений, в лабораториях	86	71	61	54	49	45	42	40	38	50

Уровень звука и звукового давления измеряется на расстоянии 50 см от поверхности оборудования и на высоте источника звука.

На рабочем месте уровень шума находится в пределах допустимых значений и не вызывает серьезного беспокойства и не влияет на психологическое состояние сотрудника.

## **12.6 Электромагнитные полями, неионизирующие ткани тела человека**

Источником повышенной напряженности электромагнитного поля на рабочем месте является персональный компьютер или ЭВМ. в связи с наличием в нем различных электронных компонентов, генерирующих электромагнитное излучение – как то сетевые фильтры, источники бесперебойного питания и т.д.

Согласно СанПиН 2.2.4.1191–03 [39], 8-ми часовой рабочий день для сотрудника на рабочем месте, с предельно допустимым уровнем напряженности должен составлять не более 8 кА/м, а уровень магнитной индукции – 10 мТл. Соблюдение данных норм дает возможность избежать негативного воздействия электромагнитных излучений.

Для уменьшения силы воздействия электромагнитного поля на сотрудника, в одну розетку необходимо включать не более двух ЭВМ, обеспечить заземление вычислительной техники, подключать ее к сети через нейтрализатор электрического поля.

Длительное воздействие электромагнитного поля на организм человека вызывает нарушение дыхательной, нервной и сердечно-сосудистой систем, в дальнейшем - головные боли и усталость.

Отдельно стоит обратить внимание на необходимость замены старых ЭЛТ мониторов, которые являются мощным источником электромагнитных полей (ЭМП) из-за использования в них силовых трансформаторов, систем горизонтального отклонения луча электроннолучевой трубки дисплея, блоков модуляции и высоковольтных кинескопов.

Наиболее сильно действие ЭМП проявляется на расстоянии до 30 см от монитора. Но вредное излучение не меньшей интенсивности, чем от экрана, имеют боковые и задняя поверхность монитора (источник - строчный трансформатор). Это обстоятельство необходимо учитывать при организации рабочих зон в помещениях, где размещается не один компьютер.

Для снижения уровня воздействия электромагнитного излучения на рабочем месте сотрудника используется жидкокристаллический монитор, выполнено заземление всего электронного оборудования, а также используется нейтрализатор электрического поля Siemens XLT-20141.

## **12.7 Разница электрических потенциалов, под действие которого попадает работающий**

Поражение током оператора ВМ может произойти в следующих случаях:

- прикосновение к открытым элементам ЭВМ, находящимся под напряжением;
- прикосновение к токоведущим частям в случае повреждения изоляции;
- при контакте с элементами производственного помещения, которые оказались под напряжением;
- возникновение короткого замыкания в блоке питания ЭВМ или монитора.

Защита сотрудника от поражения электрическим током достигается за счет конструктивных решений, техническими способами и средствами защиты, а также организационными мероприятиями.

Для обеспечения защиты от случайного прикосновения к токоведущим частям оборудования применяют: изоляцию токопроводящих частей; защитное заземление, зануление, аппаратуру защитного отключения, предупредительную сигнализацию. Так же, с сотрудниками проводятся инструктажи по технике безопасности при работе с оборудованием, находящемся под напряжением.

На рабочем месте сотруднику запрещено прикасаться к тыльной стороне монитора, вытирать пыль с компьютера в то время когда он находится в рабочем состоянии или включенным в сеть электрического питания, работать на компьютере во влажной одежде и влажными руками.

Помещение кабинета по электробезопасности сухое, с токонепроводящими полами, с температурой 18 – 21 °С и влажностью 40 – 50%, поэтому оно относится к помещению без повышенной опасности, согласно СНиП 2.01.02-85 [40].

## **12.8 Физические статические перегрузки, связанные с рабочей позой.**

В период работы сотрудника за ЭВМ, количество мелких стереотипных движений кистей и пальцев рук за смену может превысить 60 тыс., вследствие чего могут развиваться болезненные воспалительные процессы тканей сухожилий, получившие название «травма повторяющихся нагрузок». Также для сотрудника, проводящего большую часть рабочего времени за ЭВМ, характерно появление «синдрома длительных статических нагрузок», выражающегося в болях в шее и спине.

Причиной возникновения этих синдромов является длительное пребывание в положении «сидя», которое приводит к сильному перенапряжению мышц спины и ног, в результате чего возникают боли и неприятные ощущения в нижней части спины. Основной причиной перенапряжения мышц спины, шеи и ног являются нерациональная высота рабочей поверхности стола и сидения, отсутствие опорной спинки и подлокотников, неудобное размещение монитора, клавиатуры и документов, отсутствие подставки для ног.

Работа на ПК сопровождается постоянным и значительным напряжением функций зрительного анализатора. Одной из основных особенностей является иной принцип чтения информации, чем при обычном чтении. При работе на ПК оператор считывает текст, почти не наклоняя голову, глаза смотрят прямо вперед, текст формируется по другую сторону экрана, поэтому пользователь не

считывает отраженный текст, а смотрит непосредственно на источник света, что вынуждает глаза и орган зрения в целом работать в несвойственном ему стрессовом режиме длительное время.

Работа на ПЭВМ более четырех часов в день значительно увеличивает риск расстройства органов зрения. Существует понятие «компьютерный зрительный синдром», симптомами которого являются жжение в глазах, покраснение век и конъюнктивиты и т.д.

Для снижения негативного воздействия данного фактора на предприятии реализован ряд мер согласно ГОСТ 12.2.032-78 ССБТ, ГОСТ Р ИСО 9241-5-2009 и ТК РФ от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018), таких как:

- Каждое рабочее место оборудовано креслом Herman Miller Aeron с упором для головы и подставкой для ног;
- два раза в день проводится 10-минутная «зарядка», направленная на снятие статического напряжения мышц тела и растяжку суставов и связок
- регулярно проводится инструктаж с сотрудниками о необходимости 5 минутных перерывов в работе каждый час работы за ЭВМ.

## **12.9 Умственное перенапряжение**

Причинами умственного перенапряжения при работе за ЭВМ являются: дефицит времени, большой объем и плотность информации, диалоговый режим общения человека и ПК, ответственность за безошибочность информации.

Продолжительная работа на ПК, особенно в диалоговом режиме, может привести к нервно-эмоциональному перенапряжению, нарушению сна и т.д.

Для снижения негативного воздействия данного фактора на предприятии реализован ряд мер в соответствии с ТК РФ от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018) [28], таких как:

- Приобретение за счет компании комплекса поливитаминов для сотрудников;
- оплата посещения спортзала во вне рабочее время.

## **12.10 Экологическая безопасность**

В рамках ВКР источниками экологической опасности являются ЭВМ, на которой выполняется работа по разработке ПО, а также последующее использование данного ПО, и люминесцентные лампы.

В конце срока службы ПК и монитор относятся к отходам электронной промышленности согласно ГОСТ 30775-2001[41]. Переработка таких отходов осуществляется путем разделения на однородные компоненты (пригодные для дальнейшего использования) и передачи для дальнейшего использования согласно ГОСТ 30775-2001 [41].

В соответствии с данным стандартом, все отходы, образованные в соответствии с их классами опасности, передаются на специализированные предприятия для переработки, утилизации или захоронения.

Люминесцентные лампы согласно ГОСТ Р 52105-2003 «Ресурсосбережение. Обращение с отходами. Классификация и методы переработки ртутьсодержащих отходов. Основные положения» [42] относятся к 1-й группе РСО и передаются на специализированные предприятия для переработки и утилизации.

## **12.11 Безопасность в чрезвычайных ситуациях**

### **Анализ вероятных чрезвычайных ситуаций, которые могут возникнуть на производстве при эксплуатации объекта исследований**

Наиболее характерной чрезвычайной ситуацией для помещений, оборудованных ЭВМ, является пожар.

Причинами возникновения данного вида ЧС являются:

- возникновение короткого замыкания в электропроводке оборудования или производственного помещения;
- возгорание компонентов ПК или монитора из-за неисправности;
- возгорание устройств искусственного освещения.

**Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС.**

Согласно СП 12.13130.2009 [43] рабочее помещение офисного типа относится к категории пожароопасности В и считается пожароопасным. Такая классификация связана с наличием в офисных помещениях горючих и трудногорючих материалов и веществ, которые при контакте с воздухом горят без образования взрывоопасных смесей.

Согласно ФЗ от 22.07.2008 N 123-ФЗ «Технический регламент о требованиях пожарной безопасности» [44] помещение оборудовано следующими средствами пожаротушения:

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

Принудительная эвакуация при пожаре производится при усиливающемся действии опасных факторов пожара. Сокращение времени эвакуации достигается наличием аварийных маршрутов и выходов. Количество, размеры и планировка аварийных выходов регламентируются строительными нормами СНиП 2.01.02-85 [40].

Для предотвращения возникновения пожара необходимо проводить следующие мероприятия:

- периодическая проверка проводки;
- отключение оборудования при покидании рабочего места;
- проведение с работниками инструктажа по пожарной безопасности.
- установка противопожарной сигнализации;
- оборудование помещений огнетушителями и планами эвакуации;

Всякий работник при обнаружении пожара должен:

- незамедлительно сообщить об этом в пожарную охрану;
- принять меры по эвакуации людей, каких-либо материальных ценностей согласно плану эвакуации;
- отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения.

Производственные помещения банка оснащены ручными углекислотными огнетушителями ОУ-2 по одному на офис, а также аптечками первой помощи согласно требованиям ГОСТ Р 51057-01 [45].

План эвакуации предоставлен на Рисунок 59.

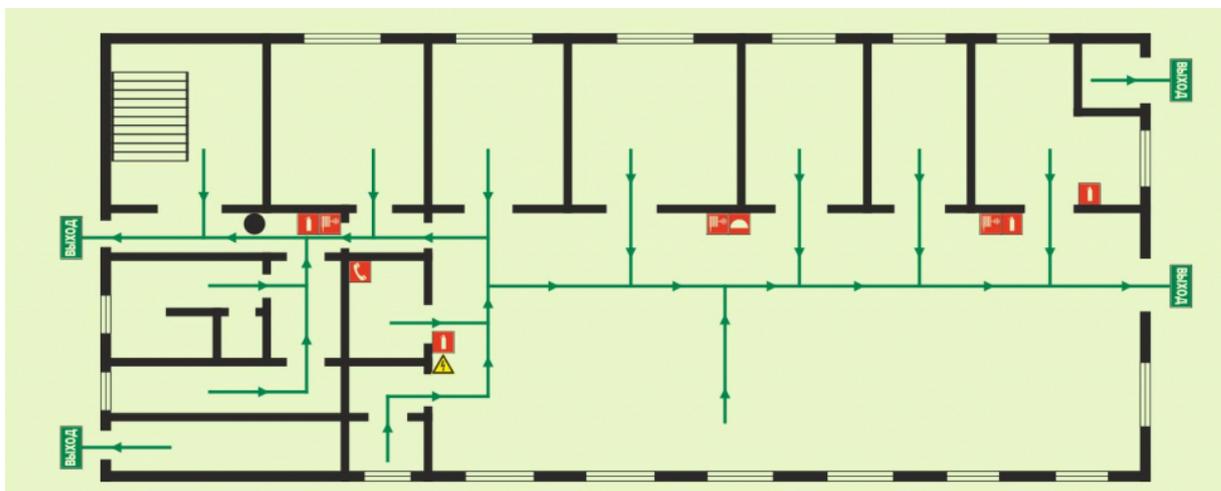


Рисунок 59 – План эвакуации при пожаре и других ЧС из офисного помещения банка, 2-й этаж

## 12.12 Выводы по главе

В результате проделанной работы можно сделать вывод, что социальная ответственность – это принятие организацией и ее руководством определённых обязательств по обеспечению комфортных и безопасных условий труда сотрудников. Это взаимовыгодное сотрудничество между государством, организацией и ее сотрудниками, обеспечивающее эффективную работу сотрудника на благо страны, организации и самого себя.

Будущий специалист должен проводить профессиональную деятельность с учетом социальных, правовых, экологических и культурных аспектов, вопросов здоровья и безопасности, нести социальную ответственность за принимаемые решения, осознавать необходимость устойчивого развития.

В данном разделе ВКР была продемонстрирована способность анализировать характер негативных воздействий различных факторов здоровье сотрудника в соответствии с полученными знаниями, методами оценки и расчетов уровня воздействий. Социальная ответственность при разработке новых решений обеспечивает: устранение несчастных случаев, защиту здоровья

сотрудников, снижение вредного воздействия на окружающую среду, бережливое использование не возобновляемых природных ресурсов.

## 13. Заключение

В ходе выпускной квалификационной работы были решены следующие задачи

1. Исследования рынка и экономическое обоснование создания стартапа.
2. Создание высокотехнологичного продукта, являющегося основой данного стартапа.

В технической части работы, был выполнен детальный анализ математического аппарата, лежащего в основе современной портфельной теории. Показана функциональная зависимость доходности портфеля ценных бумаг от его риска.

Был выполнен выбор технологий для разработки веб-приложения, а также для оптимизации выполнения расчетов, как-то – для организации параллельных вычислений на распределенных ядрах и узлах, а также для организации асинхронного выполнения вычислительных задач.

Были выбраны технологии, основанные на подходе «инфраструктура как код», для описания и развертывания облачной инфраструктуры для хостинга веб-приложения.

Была выполнена настройка тестов по всем трем уровням детализации.

Бы настроен конвейер непрерывной интеграции и непрерывного развертывания для непрерывного запуска нового кода в работу.

В экономической части работы был выполнен детальный анализ рынка и проведено сравнение разрабатываемого веб-приложения и конкурентных решений, выполнен Customer Development и определена целевая аудитория продукта, которая учтена в стратегии продвижения продукта на рынке. Рассчитана экономическая целесообразность создания стартапа и себестоимость продукта, которые показали экономическую целесообразность коммерциализации предлагаемого решения.

В результате создан продукт – веб-приложение, отвечающий поставленным целям исследования и позволяющий пользователю принимать обоснованные инвестиционные решения.

## 14. Публикации и достижения

Иштуганов Р. А. Разработка веб-приложения для оптимизации портфеля ценных бумаг // Сборник избранных статей научной сессии ТУСУР: В 2 частях. Часть 2, Томск, 25-27 Мая 2020. - Томск: В-Спектр, 2020 - С. 196-200

Иштуганов Р. А. Классификация современных портфельных теорий // Математическое и программное обеспечение информационных, технических и экономических систем: материалы Международной научной конференции, Томск, 28-30 Мая 2020. - Томск: ТГУ, 2020 - С. 216-224

Диплом III степени за доклад на подсекции 5.2 «Информационные системы в экономике» Международной научно-технической конференции студентов, аспирантов и молодых ученых «НАУЧНАЯ СЕССИЯ ТУСУР», Томск, ТУСУР, 2020.

Диплом за доклад на Международной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем», Томск, ТГУ, 2020.

Сертификат участника и докладчика на XIX Международной конференции имени А.Ф. Терпугова «Информационные технологии и математическое моделирование», Томск, ТГУ, 2020.

Сертификат финалиста «VIII Всероссийского инженерного конкурса» ВИК 2021, МИФИ, Москва.

Повышенная академическая стипендия за достижения в учебной деятельности, осенний семестр 2020/2021 учебного года.

Повышенная академическая стипендия за достижения в научной деятельности, весенний семестр 2020/2021 учебного года.

Стипендия благотворительного фонда имени В. Потанина, весенний семестр 2020/2021 учебного года.

## 15. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Age Structure [Электронный ресурс] / Н. Ritchie, М. Roser. – Электрон. дан. - Our World in Data: Global Change Data Lab, 2019. URL: <https://ourworldindata.org/age-structure>, свободный. – Загл. с экрана – Яз. англ. Дата обращения 06.01.2020 г.
2. Portfolio Selection / Н. Markowitz // Journal of Finance – 1952 – Vol.7, №1 – P.77-91.
3. Mutual Fund Performance / W. Sharpe // The Journal of Business. – 1966 – Vol.39, №1 – P. 119-138.
4. Post-Modern Portfolio Theory Comes of Age / В. Rom, К. Ferguson // The Journal of Investing – 1993 – Vol.2, №4 – P.27-33.
5. Sortino F. Managing Downside Risk in Financial Markets / F. Sortino, S. Satchell – Oxford: Oxford Press, 2001. – P. 351.
6. Capital Asset Prices: A Theory of Market Equilibrium under Consideration of Risk / W. Sharpe // Journal of Finance – 1964 – Vol. 19 №3 – P.425-442.
7. Mean-Risk Analysis with Risk Associated with Below-Target Returns / P. Fishburn // The American Economic Review. – 1997. – Vol.67, №2 – P. 116-126.
8. Performance Measurement in a Downside Risk Framework / F. Sortino, L. Price // The Journal of Investing. – 1994. – Vol.3, №3 – P. 59-64.
9. On estimating the expected return on the market: An exploratory investigation / R. Merton // Journal of Financial Economics. – 1980. – Vol.8, №4 – P. 323-361.
10. The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice / V. Chopra, W. Ziemba // The Journal of Portfolio Management. – 1993. – Vol.19, №2 – P. 6-11.
11. Global Portfolio Optimization / F. Black, R. Litterman // Financial Analysts Journal. – 1992 – Vol.48, №5 – P.28-44.
12. 2018 Kaggle ML & DS Survey [Электронный ресурс] / Kaggle. – Электрон. дан. – Kaggle, 2018. URL <https://www.kaggle.com/kaggle/survey-2018/>, свободный. – Загл. с экрана. – Яз. англ. Дата обращения 27.12.2019 г.
13. Software Defect Reduction Top 10 List / В. Boehm, V. Vasili // IEEE Computer. – 2001. – Vol.34, №1 – P. 135-137.
14. Khorikov V. Unit Testing: Principles, Practices and Patterns / V. Khorikov – Shelter Island: Manning, 2020. – P. 561.
15. Развёртывание программного обеспечения [Электронный ресурс] / Wikipedia. – Электрон. дан. – Wikimedia Foundation, Inc., 2020. URL:

[https://ru.wikipedia.org/wiki/Развёртывание\\_программного\\_обеспечения](https://ru.wikipedia.org/wiki/Развёртывание_программного_обеспечения), свободный. – Загл. с экрана – Яз. рус. Дата обращения 01.06.2020 г.

16. yahoo.com Competitive Analysis, Marketing Mix and Traffic [Электронный ресурс] / Alexa Internet, Inc. – Электрон. дан. – Alexa Internet, Inc., 2020. URL: <https://www.alexa.com/siteinfo/yahoo.com>, свободный. – Загл. с экрана. – Яз. англ. Дата обращения 08.01.2020 г.

17. Индивидуальный инвестиционный счет [Электронный ресурс] / Wikipedia. – Электрон. дан. – Wikimedia Foundation, Inc., 2020. URL: [https://ru.wikipedia.org/wiki/Индивидуальный\\_инвестиционный\\_счёт](https://ru.wikipedia.org/wiki/Индивидуальный_инвестиционный_счёт), свободный. – Загл. с экрана. – Яз. рус. Дата обращения 9.01.2020 г.

18. Инвестиционная методология автоматического финансового советника «Простые инвестиции» [Электронный ресурс] / ООО УК «ФинЭкс Плюс». – Электрон. дан. - ООО «УК «ФинЭкс Плюс» при поддержке ПАО Сбербанк, 2020. URL: <https://prosto-invest.ru/static/content/Methodology-prosto-invest.pdf>, свободный. – Загл. с экрана. – Яз. рус. Дата обращения 09.01.2020 г.

19. Робот-советник [Электронный ресурс] / ВТБ Инвестиции. – Электрон. дан. – ВТБ Инвестиции, 2020. URL: [https://broker.vtb.ru/invest/adviser/?code=media\\_rbcquote\\_roboadviser\\_media&utm\\_source=rbcquote\\_roboadviser&utm\\_medium=media&utm\\_content=&utm\\_campaign=media\\_quote\\_sp\\_rbcquote\\_roboadviser\\_roboadviser\\_cpc\\_region](https://broker.vtb.ru/invest/adviser/?code=media_rbcquote_roboadviser_media&utm_source=rbcquote_roboadviser&utm_medium=media&utm_content=&utm_campaign=media_quote_sp_rbcquote_roboadviser_roboadviser_cpc_region), свободный. – Загл. с экрана. – Яз. рус. Дата обращения 16.11.2020 г.

20. 2020 – год частного инвестора [Электронный ресурс] / БКС-Экспресс. – Электрон. дан. - БКС-Экспресс, 2020. URL: <https://bcs-express.ru/novosti-i-analitika/2020-god-chastnogo-investora>, свободный. – Загл. с экрана. – Яз. рус. Дата обращения 01.05.2021 г.

21. Инвестиции выходят в массы [Электронный ресурс] / ИА "РосБизнесКонсалтинг". – Электрон. дан. – ИА "РосБизнесКонсалтинг", 2019. URL: <https://plus.rbc.ru/news/5cf72b497a8aa96ad286ece9>, свободный. – Загл. с экрана. – Яз. рус. Дата обращения 01.05.2021 г.

22. Число частных инвесторов на фондовом рынке РФ превысило 8 млн [Электронный ресурс] / ИА "Финмаркет". – Электрон. дан. – Финмаркет, 2020. URL: <http://www.finmarket.ru/news/5370817>, свободный. – Загл. с экрана. – Яз. рус. Дата обращения 01.05.2021 г.

23. More than half of U.S. households have some investment in the stock market [Электронный ресурс] / Pew Research Center; K. Parker, R. Fry. – Электрон.дан. – Pew Research Center, 2020. URL: <https://www.pewresearch.org/fact-tank/2020/03/25/more-than-half-of-u-s-households-have-some-investment-in-the-stock-m>, свободный. – Загл.с экрана. – Яз. англ. Дата обращения 01.05.2021 г.

24. A Snapshot of Investor Households in America / G. Mottola // FINRA Investor Education Foundation. – 2015. – P.53.

25. Портрет частного российского инвестора [Электронный ресурс] / ФИНАМ. – Электрон. дан. – ФИНАМ, 2008. URL: <https://www.finam.ru/about/newsitem2CE05>, свободный. – Загл. с экрана. – Яз. рус. Дата обращения 01.05.2020 г.
26. Исследование Тинькофф Инвестиций: российский розничный инвестор – кто он такой? [Электронный ресурс] / vc.ru. – Электрон. дан. – vc.ru, 2020. URL: <https://vc.ru/finance/107104-issledovanie-tinkoff-investiciy-rossiyskiy-rozничnyy-investor-kto-on-takoy>, свободный. – Загл. с экрана. – Яз. рус. Дата обращения 01.05.2021 г.
27. Остервальдер А. Построение бизнес-моделей: Настольная книга стратега и новатора: пер. с англ. / А. Остервальдер, И. Пинье. – М.: АЛЬПИНА ПАБЛИШЕР, 2013. – 288 стр.
28. Трудовой кодекс Российской Федерации от 30.12.2001 №197-ФЗ (ред. от 27.12.2018). – Москва, Кремль, 2001.
29. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования. – М.: ИПК Издательство стандартов, 2001.
30. ГОСТ 21889-76. Система «человек-машина». Кресло человека-оператора. Общие эргономические требования – М.: Издательство стандартов, 1993.
31. ГОСТ 22269-76. Система «человек-машина». Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования. – М.: Издательство стандартов, 1990
32. ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре. – М.: Стандартиформ, 2019.
33. ГОСТ Р ИСО 9241-5-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора. – М.: Стандартиформ, 2019.
34. ГОСТ 12.0.003-2015 ССБТ. Опасные и вредные производственные факторы. Классификация. – М.: Стандартиформ, 2017.
35. СП 2.4.3648-20. Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи. [Электронный ресурс] / Официальный интернет-портал правовой информации. – Электрон. дан. – Государственная система правовой информации, 2020. URL: [www.pravo.gov.ru](http://www.pravo.gov.ru), свободный. – Загл. с экрана. – Яз. рус. Дата обращения 01.06.2021.
36. СанПиН 2.2.4.3359-16. Санитарно-эпидемиологические требования к физическим факторам на рабочих местах. – М.: Стандартиформ, 2017.

37. ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны. – М.: Стандартинформ, 2008.
38. СП 52.13330.2016. Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95\*. – М.: Стандартинформ, 2018.
39. СанПиН 2.2.4.1191–03. Электромагнитные поля в производственных условиях.
40. СНиП 2.01.02-85. Противопожарные нормы. – М.: Госстрой СССР, 1991.
41. ГОСТ 30775-2001. Классификация, идентификация и кодирование отходов. – М.: ИПК Издательство стандартов, 2002.
42. ГОСТ Р 52105-2003. Ресурсосбережение. Обращение с отходами. Классификация и методы переработки ртутьсодержащих отходов. Основные положения. – М.: ИПК Издательство стандартов, 2003.
43. СП 12.13130.2009. Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности. – М.: ФГУ ВНИИПО МЧС России, 2009.
44. Федеральный Закон от 22.07.2008 №123-ФЗ. Технический регламент о требованиях пожарной безопасности. – М.: Кремль, 2008.
45. ГОСТ Р 51057-01. Техника пожарная. Огнетушители переносные. Общие технические требования. Методы испытаний. – М.: ИПК Издательство стандартов, 2002.

# 16. Приложение А

(справочное)

## Раздел 8 Software testing

Студент:

Группа	ФИО	Подпись	Дата
8ВМ92	Иштуганов Руслан Александрович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Соколова В.В.	к.т.н., доцент		

Консультант – лингвист отделения иностранных языков ШБИП:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИЯ ШБИП	Аксёнова Н. В.	к.ф.н., доцент		

## 16.1 Software Testing

Testing a software product is the process of studying the operation of a given software product in order to obtain information about its quality. The quality of a software product, in turn, characterizes how fully it meets the requirements of the concerned parties - primarily the customer, but also the contractor, both as a company and as its departments and employees - designers, programmers, marketers, etc. Each of the participants may have a different idea of the product, its qualities and the ideal condition of the product. Thus, setting the task of ensuring the quality of the product involves determining the parties concerned, their quality criteria and then finding the optimal solution that meets these criteria. Testing is one of the most established ways to ensure the quality of software development and is part of the set of effective tools of a modern software product quality assurance system.

From a technical point of view, testing consists in executing an application on a set of initial data and comparing the results obtained with previously known reference data in order to establish the correspondence of various properties and characteristics of the application to the ordered properties. After all, if the program does not contain syntax errors, has been translated and can be executed on a computer, it necessarily calculates some function that maps the input data to the output. This means that the computer using its resources redefines the function partially defined by the program to total certainty. Consequently, it is possible to judge the correctness or incorrectness of the program execution results only by comparing the specification of the desired function with the results of its calculation, which is carried out in the testing process.

It should be noted that testing is the most expensive stage in the development of a software product for a customer in terms of the cost of eliminating errors in the code. For example, according to [13], the average cost of fixing one bug in a code during the testing phase in the United States is more than \$7,000, compared to \$1,000 during the implementation phase (see Figure 1). However, the correction costs double and exceed \$14,000 when the product is put into production. At the same time, if the contractor's costs are easy to calculate, delay damages can be given for delivery time and an increase in labor costs for finding and eliminating errors in the code, the

customer's costs at the production stage and, particularly, revenue loss are much more difficult to calculate, and impossible to estimate accurately.

That is why, in the 2000s, there was a significant increase in interest both from the scientific community and from the business side in the problem of software testing.

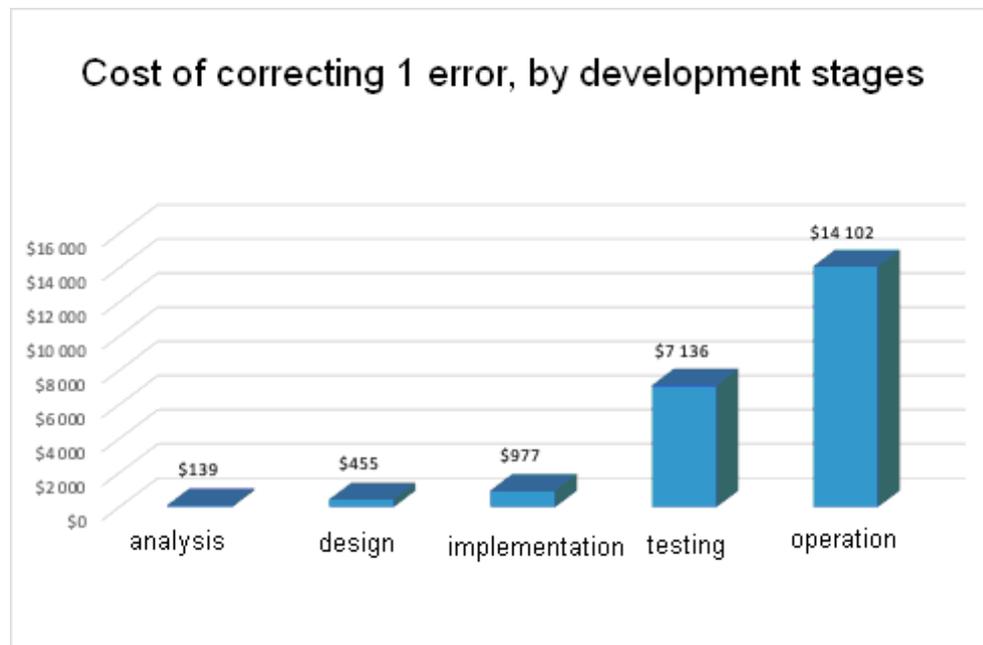


Figure 1 – Cost of correcting 1 error in code at development stages

By now, various methods and approaches to test software products have formed into a separate branch of scientific knowledge - QA (or Quality Assurance). The given industry has developed its own methodology, methods and tools, as well as a theoretical framework, with the goal of improving the quality of testing, reducing the cost of testing and meeting the ever-changing business requirements and contexts as closely as possible.

However, when developing software in small teams, it is impossible to cover all existing software testing tools and, therefore, often we have to choose which methods and tools to use based on certain business constraints. For a more reasoned choice, we first need to consider the classification of test methods in order to understand what exactly is right for developers at the moment and in the present context.

## **Classification of testing methods**

### **Functional / non-functional testing**

Functional testing is testing that validates the performance of specific interactions with software or a specific segment of the program code. Non-functional testing is used to verify such aspects of software as scalability or security, that is, not directly related to the performance of the final product.

### **Dynamic/static testing**

Dynamic testing is an approach in which the program code is run for execution and the results of the work are checked against the reference data. In static testing, it is not the work of the code that is studied, but the work of the entire software as a whole in the form of reviews, end-to-end testing, step-by-step execution of user tasks, etc.

### **Black box / White box**

When approaching testing according to the "black box" principle, the details of the software implementation, as the code itself, the algorithms and data structures used, remain unknown. When approaching testing according to the "white box" principle, this information is disclosed.

### **Testing by levels of refinement**

One of the most common ways to classify software testing is by level of refinement. These levels are:

3. Unit testing is testing the minimum segment of code (module) that performs one atomic function.
4. Integration testing is testing a set of modules that perform a certain functionality of business logic.
5. System testing is testing the entire system, the complete software product.
6. System integration testing is testing the entire system in integration with other systems, if there are any.

Thus, having considered this classification of testing methods, we will apply functional dynamic testing according to the white box principle. Testing will be performed at the following levels of refinement: unit, integral, system.

## **Unit Testing**

### **Definition**

The historical definition of unit testing is the following: unit testing is the testing of minimal blocks of code that perform atomic functions. In terms of Object Oriented Programming, it is a test of software functions, methods and attributes of a class, etc. The module test shall meet the following requirements:

7. It does a test of a small piece of code
8. It runs quickly
9. It runs isolated

Interpretation of these requirements gave rise to two testing schools:

10. Classical school
11. London school

The differences between these schools are presented in Table 1.

Table 1. Differences between Classical and London testing schools

School	Isolation	Module	Mocks
London	Of modules to be tested	Class	For all dependencies except unchangeable
Classical	Of unit tests	Class or a set of classes	For shared dependencies only

We will take a classic approach to unit testing, as it has a number of advantages, such as:

- A unit of program behavior within the problem domain is tested rather than a unit of code, making this test useful and understandable to the bigger number of the concerned parties.
- If testing a unit of program behavior reveals that too many classes and/or software modules are involved, then it means that the code has high cohesion and requires re-factoring.

### Objectives

The main objective of unit testing is to ensure the sustained growth of the scale of the software product. Although, writing tests requires some labor costs, and, at the initial stage of the project this may not be so obvious, but with its increasing complexity, the project modification without the presence of high-quality unit tests will

become more and more difficult (see Figure 2), and ultimately, it will become impossible due to the presence of a code entropy effect (its disordering) that increases with each new change.

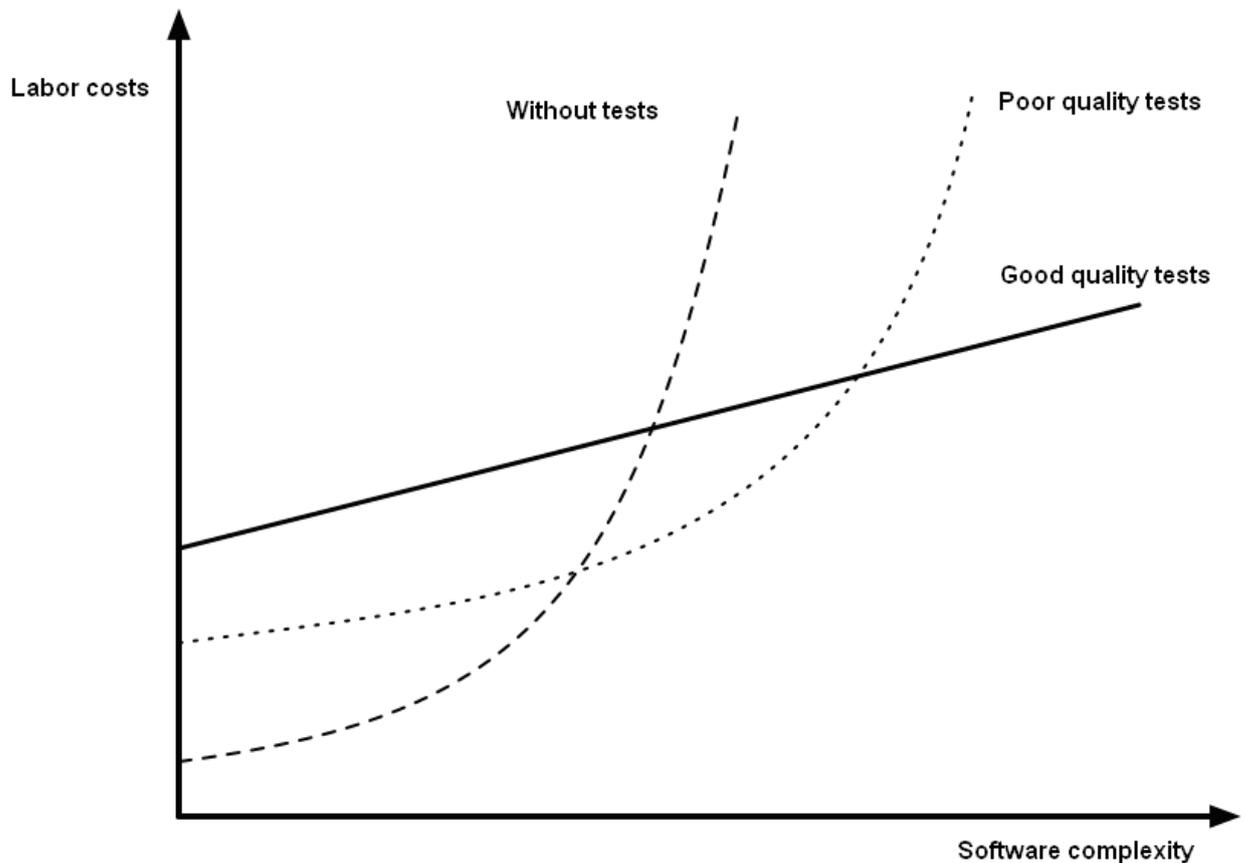


Figure 2 – Dependence of labor costs on tests availability within the increasing complexity

### Evaluation of tests

Unit tests are usually evaluated by coverage ratio.

The coverage factor is the ratio of the number of lines of code executed by the test to the total number of lines of code in a given software product. Usually, better to try and achieve 100% of the coverage, however, one should remember that, although, this factor is a strong negative indicator - that is, a low coverage factor clearly indicates that the quality of the code is poor, but at the same time the coverage factor is a weak positive indicator - that is, even a 100% coverage factor does not say anything about the high quality of the code.

Therefore, subjective criteria for evaluation of the quality of tests are often used. For instance, only tests that meet the following conditions and requirements are considered qualitative:

- A quality test is integrated into the development cycle and runs every time the code changes.
- The most critical code segments (related to the problem domain) should be subjected to the most thorough testing, the rest code segments – to less thorough testing.
- A quality test ensures maximum usefulness with minimal support costs.

## Testing

Django framework provides full support for testing the application at all three levels of refinement. To do this, a standard unittests library of the Python language is used together with its SimpleTestCase and TestCase subclasses.

To calculate the percentage of code coverage with unit tests, we can use the third-party “Coverage” library.

An example of a unit test using the unittest.TestCase library is shown in Figure 3.

```
class HomePageViewTest(TestCase):
def test_view_url_exists_at_proper_location(self):
    resp = self.client.get('/')
    self.assertEqual(resp.status_code, 200)
def test_view_url_by_name(self):
    resp = self.client.get(reverse('home'))
    self.assertEqual(resp.status_code, 200)
def test_view_uses_correct_template(self):
    resp = self.client.get(reverse('home'))
    self.assertEqual(resp.status_code, 200)
    self.assertTemplateUsed(resp, 'home.html')
```

Figure 3 – Sample of a unit test

This test verifies all of the following requirements:

12. Presence of a home page at the required address.
13. The home page alias used in the code leads to the actual home page.
14. The home page uses the required presentation template.

The test results are presented in Figure 4.

```
coverage run --source='.' manage.py test pages
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...
```

```
-----
Ran 3 tests in 0.094s
```

```
OK
```

```
Destroying test database for alias 'default'...
```

Figure 4 – An example of how a unit test works

The “Coverage” library displays a report on the ratio of code coverage by tests.

An example of a report is presented in Figure 5.

```
coverage report
```

Name	Stmts	Miss	Cover
manage.py	12	2	83%
med/__init__.py	0	0	100%
med/asgi.py	4	4	0%
med/settings.py	24	0	100%
med/urls.py	3	0	100%
med/wsgi.py	4	4	0%
pages/__init__.py	0	0	100%
pages/admin.py	1	0	100%
pages/apps.py	3	3	0%
pages/migrations/__init__.py	0	0	100%
pages/models.py	1	0	100%
pages/tests.py	14	0	100%
pages/urls.py	3	0	100%
pages/views.py	6	0	100%
users/__init__.py	0	0	100%
users/admin.py	10	0	100%
users/apps.py	3	3	0%
users/forms.py	11	0	100%
users/migrations/0001_initial.py	8	0	100%
users/migrations/__init__.py	0	0	100%
users/models.py	7	0	100%
users/tests.py	1	1	0%
users/views.py	1	1	0%
TOTAL	116	18	84%

Figure 5 – Sample of the report on code coverage by tests

## Integration Testing

The result of testing and verification of the individual modules that make up the software system should be the conclusion that these modules are internally consistent and meet the requirements. However, some modules rarely function on their own, so the next task after testing individual modules is to test the correctness of the interaction of several modules combined into a single whole. Such testing is called an integration testing. Integration tests are designed to verify the interaction of the system modules with each other, as well as the system itself with out-of-process dependencies. First of all, these are databases, file systems, mail servers, and all other similar external dependencies.

If we try to classify the code by the degree of complexity, relevance to the problem domain, and the number of components taking part in interactions, we can arrive at the following matrix (see Figure 6).

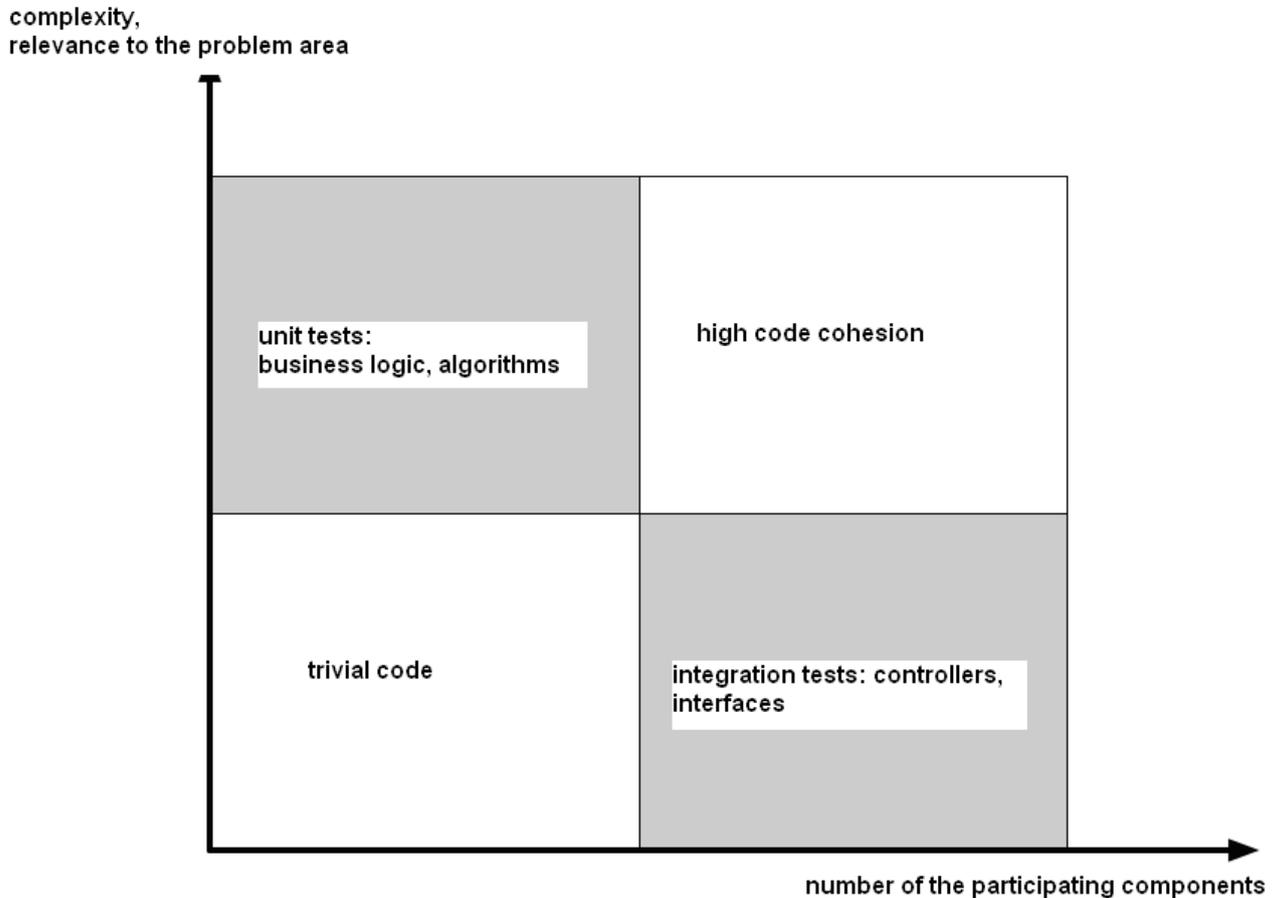


Figure 6 – Classification of code by complexity and number of "participants"

If unit tests verify the main business logic of the application, implemented in the form of certain data structures and algorithms, then integration tests verify the code that connects business logic to each other and to external systems.

It must be remembered, that integration tests are much more expensive both in terms of their writing and support, and in terms of their use - they take much more time and use much more resources compared to unit tests. Therefore, it is recommended to cover as many critical scenarios as possible using unit tests and cover the main business scenario using integration tests. Thus, one can minimize the labor costs and, at the same time, make sure that the system is functional. This rule forms the so-called "Pyramid testing," which graphically reflects the ratio of the number of tests in the system (see Figure 7).

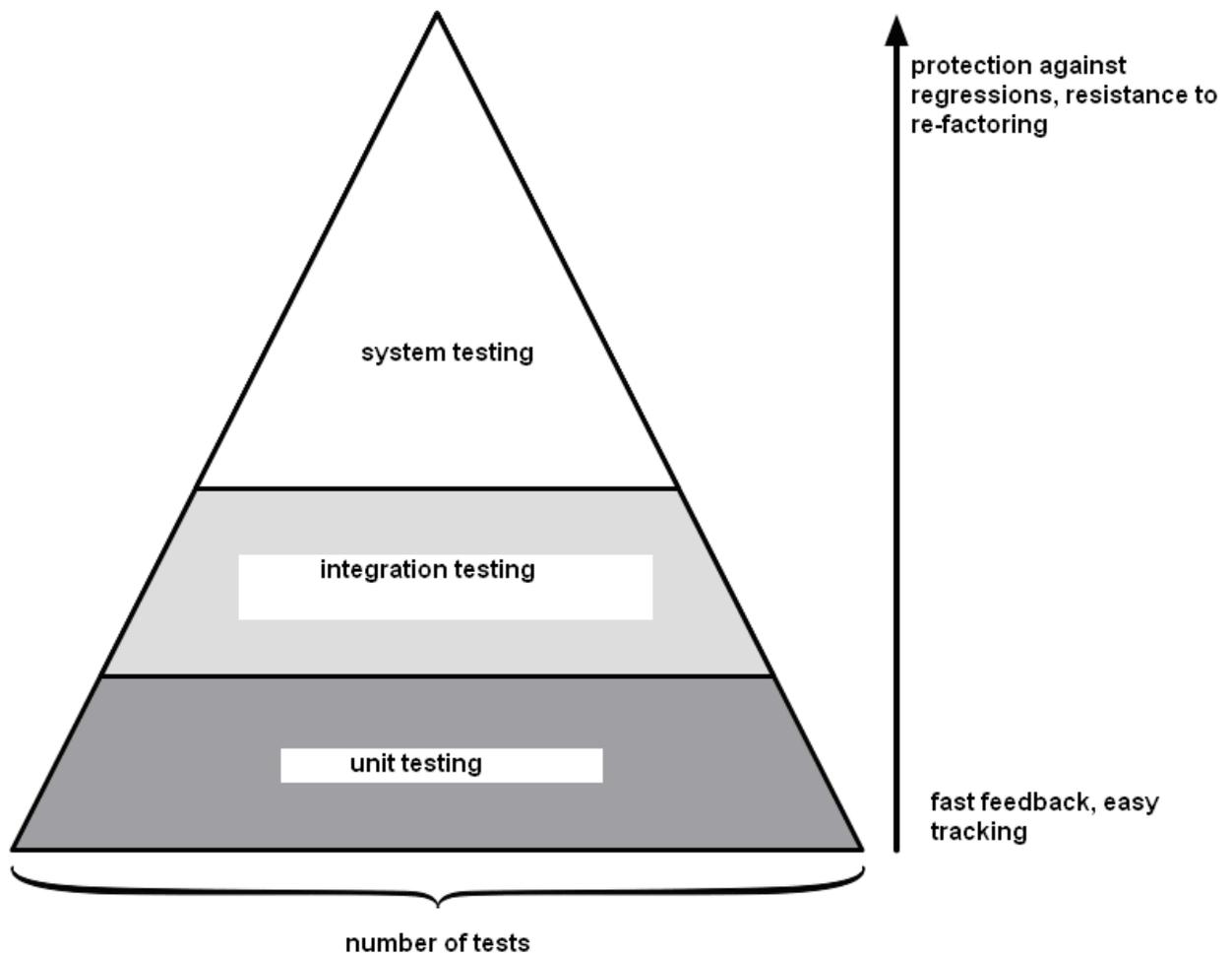


Figure 7 – Testing Pyramid

As a result of integration testing and elimination of all detected defects, a consistent and integral architecture of the software system is obtained, that is, it can be considered that integration testing is a test of the architecture and low-level functional requirements.

### **Classification of integration testing methods**

As a rule, integration testing is carried out after the completion of unit testing for all integrated modules. However, there are other methods of integration testing, which in general can be separated into the following types:

- bottom-up testing;
- monolithic testing;
- top-down testing.

### **Bottom-up testing**

This method assumes that all program modules included in the system are tested first, and only then they are combined for integration testing. With this approach, the localization of errors is significantly simplified: if the modules are tested separately, then the error that appears when they work together is a problem of their interface.

However, the bottom-up testing method has a significant disadvantage, that is the need to develop drivers and stubs for unit testing before conducting integration testing and the need to develop drivers and stubs when using integration testing on part of the system modules.

### **Monolithic testing**

This method assumes that some components of the system did not undergo a serious testing. The main advantage of this method is the lack of the need to develop a test environment, drivers and stubs. After all modules have been developed, they are integrated and the entire system is checked.

Monolithic testing has a number of serious disadvantages. Firstly, it is a difficult process to identify the location of an error.

### **Top-down testing**

This method assumes that the integration testing process follows the development cycle. First, only the uppermost control layer of the system is tested, without lower level modules. Then, more low-level modules are gradually integrated with higher-level modules. As a result of this method, there is no need for drivers (the driver role is fulfilled by a higher-level module of the system), but there is still a need for stubs.

### **Time classification of the integration testing methods**

In practice, most often all the methods considered are used together. Each module is tested separately as it is ready, and then included in the finished system. For some parts, the testing is top-down, and for others – bottom-up. In this regard, another type of classification of integration testing types can be considered - classification by integration time.

- testing with late integration

- continuous integration testing
- testing with regular or layered integration.

### Testing with late integration

This method is almost completely similar to monolithic testing. Integration testing with such a scheme is postponed to the latest possible project dates.

### Continuous integration testing

This method implies that as soon as the development of a new module is completed, it is immediately integrated with the rest of the system. At the same time, tests for this module check both its internal functionality and its interaction with other modules of the system. Thus, this approach combines unit testing and integration testing.

### Testing with regular or layered integration

This method implies that strongly interconnected groups of modules (layers) are subject to integration testing, and those layers are also then integrated with each other. This type of integration testing is also called a hierarchical integration testing, since the consolidation of the integrated parts of the system, as a rule, occurs according to a hierarchical principle.

Table 2 provides a brief comparison of the described integration testing types.

Table 2. Main characteristics of integration testing types

Property	Type of Integration					
	Bottom-up	Top-down	Monolithic	Late integration	Continuous integration	Regular integration
Time of Integration	late (after modules testing)	early (in parallel with development)	late (after all modules have been developed)	late (after all modules have been developed)	early (in parallel with development)	early (in parallel with development)
Integration frequency	seldom	often	seldom	seldom	often	often
The necessity of drivers	Yes	No	No	No	Yes	Yes
The necessity of stubs	Yes	Yes	No	No	No	Yes

## Testing

The Django Framework provides full support for testing the application at all three levels of refinement. To do this, a standard library of the Python language - unittest, and its TestCase subclass which supports integration tests, are used. In addition, the "FactoryBoy" library is used, which allows generating dynamic data that meet various requirements - uniqueness, compliance with regional data formats (for instance, telephone numbers) and so on, as well as automates the process of creating and destroying a test database.

An example of an integration test using unittest.TestCase and FactoryBoy libraries are shown in Figure 8 and Figure 9.

```
class UserFactory(factory.Factory):
    class Meta:
        model = CustomUser

class UserTestCase(TestCase):
    def setUp(self):
        self.test_user = UserFactory.create(
            passport='12345t',
            username='test_user',
            email='test_user@gmail.com',
            first_name='test_first',
            last_name='test_last',
            type='staff')

    def test_user_created(self):
        self.assertEqual(self.test_user.passport, '12345t')
        self.assertEqual(self.test_user.username, 'test_user')
        self.assertEqual(self.test_user.email, 'test_user@gmail.com')
        self.assertEqual(self.test_user.first_name, 'test_first')
        self.assertEqual(self.test_user.last_name, 'test_last')
        self.assertEqual(self.test_user.type, 'staff')
        self.assertNotEqual(self.test_user.type, 'patient')
```

Figure 8 – Example of integration test

```
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
-----
Ran 1 test in 0.015s

OK
Destroying test database for alias 'default'...
```

Figure 9 – Integration test results

## **16.2 Conclusion**

During the work on the thesis two major tasks were accomplished – market research and economic justification for creating a startup company and creation of high technology IT product that is the basis of such startup.

In the economic part of the thesis a detailed market analysis was conducted, rival solutions were reviews and their advantages and disadvantages compared. Customer Development process was carried out, target audience identified and the promotion program was based on the results of this research. Proposed business was evaluated using discounted cash flow method and business prime cost was calculated.

In the technical part of the thesis, a thorough review of modern portfolio theory methods and models was conducted, development tools based on Python programming language were selected, these tools provide powerful means as for creating web applications and for data manipulation, processing and visualization. CI/CD pipeline was setup.

As the result of this, a new product was created, that meets all established goals that allows an investor to make informed investment decisions.