



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных  
технологий

## **Отчет по практическим работам №11–14**

по дисциплине «Системная и программная инженерия»

### **Выполнили:**

Студенты группы ИМБО-02-22

Ким Кирилл Сергеевич  
Макаров Арсений Сергеевич  
Ломакин Дмитрий Владимирович  
Смирнов Дмитрий Михайлович  
Чахнин Михаил Анатольевич

### **Проверила:**

ассистент кафедры МОСИТ  
Золотухина М. А.

2025 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ПРАКТИЧЕСКАЯ РАБОТА №11 .....	4
11.1 Тестовые сценарии (тест-кейсы).....	4
ПРАКТИЧЕСКАЯ РАБОТА №12 .....	32
Аннотация .....	32
ПРАКТИЧЕСКАЯ РАБОТА №13 .....	39
13.1 Введение в теорию управления рисками .....	39
13.2 Описание рисков .....	39
13.3 Матрица рисков .....	46
13.4 Планы реагирования на риски .....	46
ПРАКТИЧЕСКАЯ РАБОТА №14 .....	50
14.1 Обзор Disaster Recovery и DRP .....	50
14.2 Составление вектора CVSS .....	50
14.3 Преобразование угроз в риски с использованием CVSS .....	52
14.4 Методы обработки эксплуатационных рисков .....	53
Риск 1. Неработоспособность серверного оборудования .....	53
Риск 2. Ошибки в программном обеспечении при обновлении системы .....	53
Риск 3. Сбой внешнего API интеграции с платёжными системами .....	54
ВЫВОД.....	55
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	58

## **ВВЕДЕНИЕ**

В настоящей работе представлена реализация основных аспектов тестирования программного продукта «Маркетплейс для жителей Африки». Работа охватывает создание и оформление тестовых сценариев, проведение функционального и нагрузочного тестирования, а также разработку методик анализа эксплуатационных рисков и обеспечения непрерывности работы системы. Особое внимание уделено не только техническим аспектам проверки корректности работы ключевых функций, таких как регистрация, авторизация, поиск товаров, оформление заказа и управление ассортиментом, но и вопросам безопасности, устойчивости и масштабируемости системы. Кроме того, рассмотрены принципы управления рисками и составление Disaster Recovery Plan (DRP) с использованием методики CVSS, что позволяет своевременно выявлять потенциальные угрозы и планировать меры по их минимизации. Общая цель работы — сформировать методологическую базу для создания надежного и качественного программного обеспечения, способного удовлетворить требования современного рынка и обеспечить высокий уровень пользовательского опыта.

# ПРАКТИЧЕСКАЯ РАБОТА №11

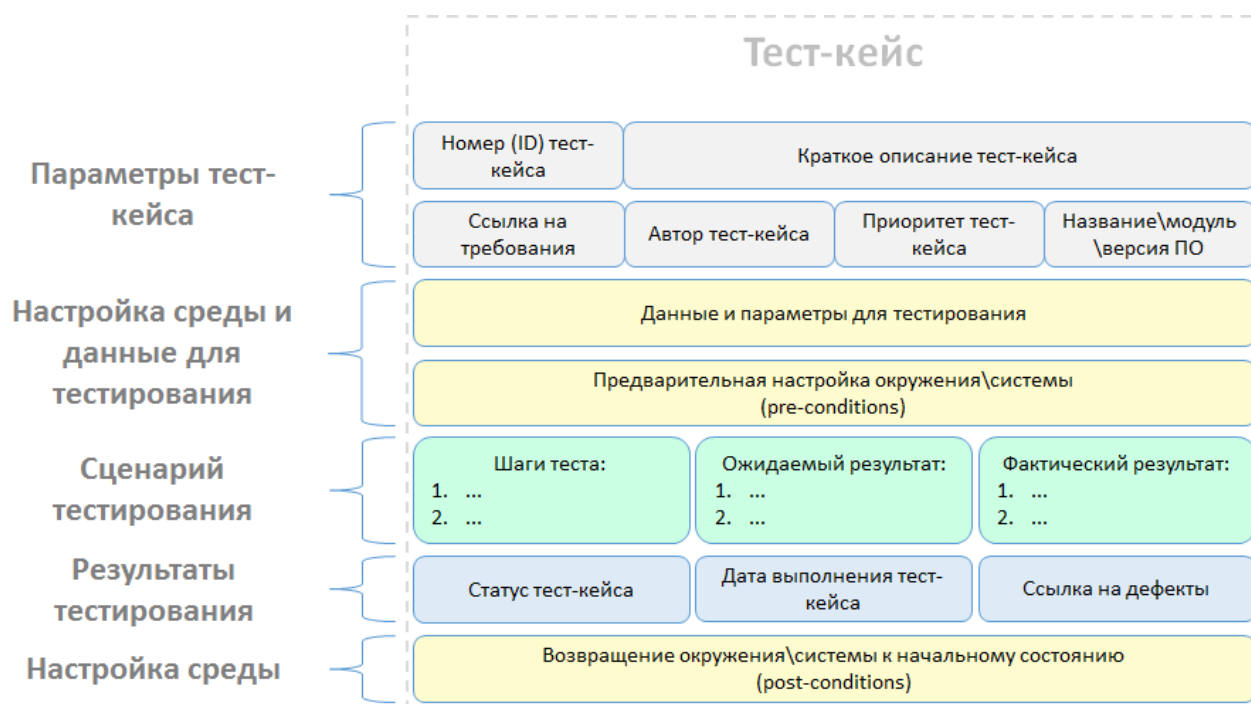
## 11.1 Тестовые сценарии (тест-кейсы)

В рамках разработки проекта «Маркетплейс для жителей Африки» сформированы функциональные и нефункциональные требования. Для подтверждения качества продукта проведено комплексное тестирование, которое включает разработку тестовых сценариев (тест-кейсов) с оформлением по образцу из методички.

Тестовые сценарии (или тест-кейсы) – это детализированные описания шагов, которые необходимо выполнить для проверки работы конкретной функции или компонента системы. Они составляют основу процесса тестирования, позволяя стандартизировать проверку каждого функционального элемента продукта.

У стандартного тестового случая есть 5 частей, то есть 5 атрибутов (рис. 11.1):

1. Порядковый номер тестового сценария
2. Название тестового сценария. Из него должно быть понятно, в чем суть тест кейса.
3. Предусловия тестового сценария. Это условия, которые необходимы для проведения тест кейса. Они должны быть выполнены еще до запуска тест кейса.
4. Порядок действий в тест кейсе и описания действий в тестовом сценарии.
5. Ожидаемый результат тестового сценария



**Рисунок 11.1 – Атрибуты тестового сценария**

Ниже описан каждый тест-кейс с помощью таблиц. Рисунки 11.2-11.11



**Рисунок 11.2 – Тест-кейс 1**



Рисунок 11.3 – Тест-кейс 2

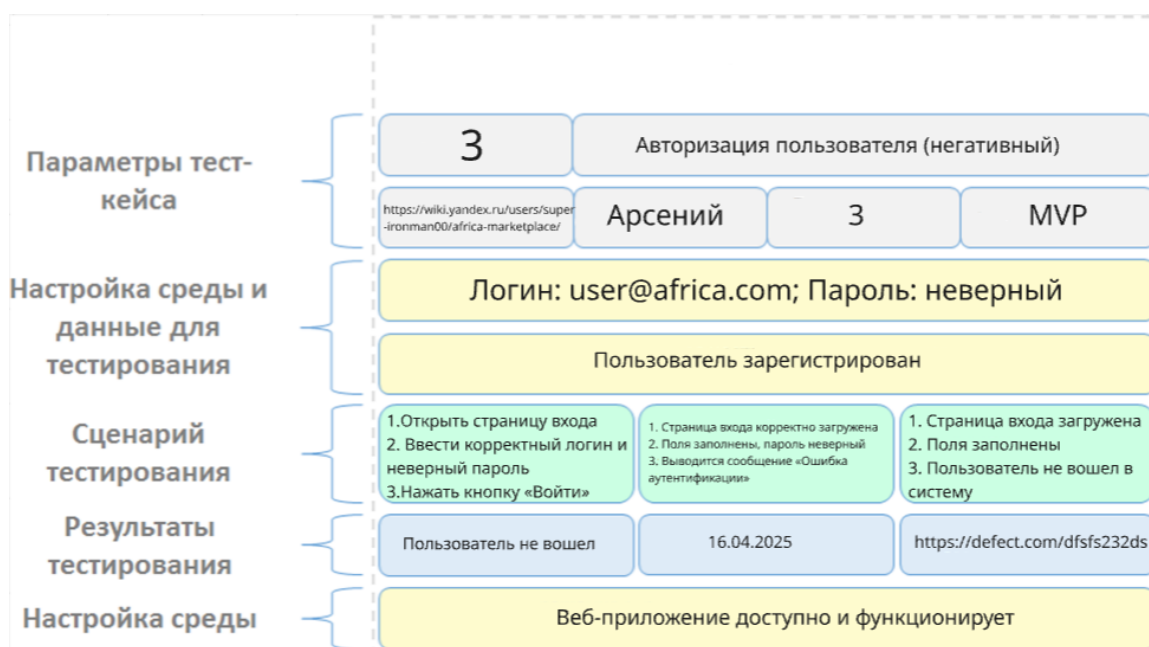


Рисунок 11.4 – Тест-кейс 3

Параметры тест-кейса	4	Поиск товара (позитивный)		
	<a href="https://wiki.yandex.ru/users/super-ironman00/africa-marketplace/">https://wiki.yandex.ru/users/super-ironman00/africa-marketplace/</a>	Арсений	4	MVP
Настройка среды и данные для тестирования	Ввод для поиска: «феромонных ловушки»;			
	Пользователь авторизован			
Сценарий тестирования	1. Ввести в строку поиска слово 2. Нажать кнопку «Поиск»	1. Текст введен в строку поиска 2. Отображается список товаров, соответствующих заданным критериям	1. Текст введен 2. Отображен список товаров	
Результаты тестирования	Отображен список товаров	16.04.2025	<a href="https://defect.com/drnfs622ds">https://defect.com/drnfs622ds</a>	
Настройка среды	Веб-приложение доступно и функционирует			

Рисунок 11.5 – Тест-кейс 4

Параметры тест-кейса	5	Поиск товара (негативный)		
	<a href="https://wiki.yandex.ru/users/super-ironman00/africa-marketplace/">https://wiki.yandex.ru/users/super-ironman00/africa-marketplace/</a>	Арсений	5	MVP
Настройка среды и данные для тестирования	Ввод: «неизвестный товар»			
	Пользователь авторизован веб-приложение доступно и функционирует			
Сценарий тестирования	<b>Шаги теста:</b> 1. Ввести в строку поиска значение «неизвестный товар» 2. Нажать кнопку «Поиск»	<b>Ожидаемый результат</b> 1. Текст введен в строку поиска 2. Товар не найден	<b>Результат теста</b> 1. Текст введен 2. Пустая страница, товар не найден	
Результаты тестирования	Сообщение получено	16.04.2025	<a href="https://defect.com/dfsfs232ds">https://defect.com/dfsfs232ds</a>	
Настройка среды	Веб-приложение доступно и функционирует			

Рисунок 11.6 – Тест-кейс 5

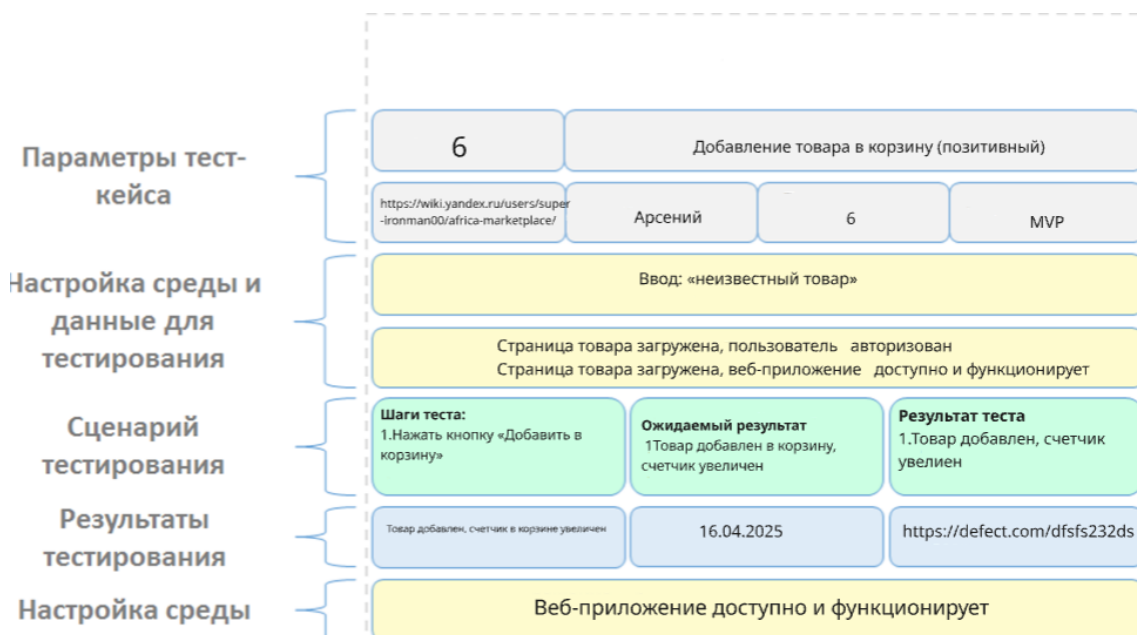


Рисунок 11.7 – Тест-кейс 6



Рисунок 11.8 – Тест-кейс 7



Параметры тест-кейса	8	Оформление заказа (негативный)		
	<a href="https://wiki.yandex.ru/users/super-ironman00/africa-marketplace/">https://wiki.yandex.ru/users/super-ironman00/africa-marketplace/</a>	Макаров А.С.	8	MVP
Настройка среды и данные для тестирования	Результат 7 теста			
	веб-приложение доступно и функционирует			
Сценарий тестирования	1. Перейти в корзину 2. Нажать кнопку «Оформить заказ» 3. Заполнить форму, кроме поля «Адрес» 4. Нажать кнопку «Подтвердить заказ»	1. Отображается содержимое корзины с товарами. 2. Переход на страницу оформления заказа. 3. Валидация формы: сообщение «Поле «Адрес» обязательно для заполнения». 4. Заказ не оформлен.	1. Корзина пуста 2. Заказ не оплачен	
Результаты тестирования	Заказ не оформлен	16.04.2025	<a href="https://defect.com/dfsfs232ds">https://defect.com/dfsfs232ds</a>	
Настройка среды	Веб-приложение доступно и функционирует			

Рисунок 11.9 – Тест-кейс 8

Параметры тест-кейса	9	Просмотр заказа		
	<a href="https://wiki.yandex.ru/users/super-ironman00/africa-marketplace/">https://wiki.yandex.ru/users/super-ironman00/africa-marketplace/</a>	Макаров А.С.	9	MVP
Настройка среды и данные для тестирования	Результат 7 теста			
	Корзина пуста, пользователь авторизован			
Сценарий тестирования	1. Перейти в раздел «Мои заказы» 2. Выбрать доставленный заказ 3. Нажать кнопку «Оставить отзыв»	1. Отображается список завершенных заказов 2. Заказ выбран для просмотра 3. Открывается форма просмотра заказа	1. Страница с заказами пользователя 2. Заказ выбран 3. Открыта форма просмотра заказа	
Результаты тестирования	Открыта форма просмотра заказа	16.04.2025	<a href="https://defect.com/dfsfs232ds">https://defect.com/dfsfs232ds</a>	
Настройка среды	веб-приложение доступно и функционирует			

Рисунок 11.10 – Тест-кейс 9



**Рисунок 11.11 – Тест-кейс 10**

Каждый тест-кейс включает четко определённые атрибуты: название кейса, функцию, которую он тестирует, предусловия (состояние системы до начала теста), последовательность действий, ожидаемые результаты и итоговый статус (пройден/провален). Такая структура помогает обеспечить полноту проверки и позволяет легко воспроизвести тест. Ниже приведены таблицы всех тест-кейсов, оформленные согласно требованиям.

**Таблица 11.1 – Регистрация пользователя**

<b>Название:</b>	Регистрация пользователя	
<b>Функция:</b>	Регистрация нового пользователя через веб-интерфейс	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
<b>Предусловие:</b>		
Пользователь не зарегистрирован	веб-приложение доступно и функционирует	Пользователь зарегистрирован
<b>Шаги теста:</b>		
Открыть страницу регистрации	Страница регистрации корректно загружена	Страница загружена
Ввести корректные	Поля заполнены согласно	Поля заполнены

данные (email, пароль, имя)	требованиям, данные введены корректно	
Нажать кнопку «Зарегистрироваться»	Отображается сообщение об успешной регистрации; создается учетная запись	Учетная запись создана

Таблица 11.2 – Авторизация пользователя

<b>Название:</b>	Авторизация пользователя (позитивный)	
<b>Функция:</b>	Вход в систему	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
<b>Предусловие:</b>		
Пользователь зарегистрирован	веб-приложение доступно и функционирует	Пользователь зарегистрирован
<b>Шаги теста:</b>		
Открыть страницу входа	Страница входа корректно загружена	Страница загружена
Ввести корректные логин и пароль	Данные введены корректно	Поля заполнены
Нажать кнопку «Войти»	Пользователь успешно авторизован, переход в личный кабинет	Авторизация успешна

Таблица 11.3 – Авторизация пользователя

<b>Название:</b>	Авторизация пользователя (негативный)	
<b>Функция:</b>	Вход в систему	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
<b>Предусловие:</b>		
Пользователь зарегистрирован	веб-приложение доступно и функционирует	Пользователь зарегистрирован
<b>Шаги теста:</b>		
Открыть	Страница входа корректно загружена	Страница входа

страницу входа		загружена
Ввести корректный логин и неверный пароль	Поля заполнены, пароль неверный	Поля заполнены
Нажать кнопку «Войти»	Выводится сообщение «Ошибка аутентификации»	Пользователь не вошел в систему

Таблица 11.4 – Поиск товара

<b>Название:</b>	Поиск товара (позитивный)	
<b>Функция:</b>	Поиск товаров	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
<b>Предусловие:</b>		
Пользователь авторизован	веб-приложение доступно и функционирует	Пользователь авторизован
<b>Шаги теста:</b>		
Ввести в строку поиска слово	Текст введен в строку поиска	Текст введен
Нажать кнопку «Поиск»	Отображается список товаров, соответствующих заданным критериям	Отображен список товаров

Таблица 11.5 – Поиск товара

<b>Название:</b>	Поиск товара (негативный)	
<b>Функция:</b>	Поиск товаров	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
<b>Предусловие:</b>		
Пользователь авторизован	веб-приложение доступно и функционирует	Пользователь авторизован

<b>Шаги теста:</b>		
Ввести в строку поиска значение «неизвестный товар»	Текст введен в строку поиска	Текст введен
Нажать кнопку «Поиск»	Товар не найден	Пустая страница, товар не найден

Таблица 11.6 – Добавление товара в корзину

<b>Название:</b>	Добавление товара в корзину (позитивный)	
<b>Функция:</b>	Добавление товара в корзину	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
<b>Предусловие:</b>		
Страница товара загружена, пользователь авторизован	Страница товара загружена, веб-приложение доступно и функционирует	Пользователь авторизован
<b>Шаги теста:</b>		
Нажать кнопку «Добавить в корзину»	Товар добавлен в корзину, счетчик увеличен	Товар добавлен, счетчик увеличен

Таблица 11.7 – Оформление заказа (позитивный)

<b>Название:</b>	Оформление заказа (позитивный)	
<b>Функция:</b>	Оформление заказа	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>

<b>Предусловие:</b>		
В корзине есть товары, пользователь авторизован	веб-приложение доступно и функционирует	Пользователь авторизован
<b>Шаги теста:</b>		
Перейти в корзину	Отображается содержимое корзины	Страница с содержимым корзины
Нажать кнопку «Оформить заказ»	Переход на страницу оформления заказа	Страница оформления заказа
Заполнить форму заказа (адрес, способ оплаты и доставки)	Форма заполнена корректно, все обязательные поля заполнены	Форма заказа заполнена
Нажать кнопку «Подтвердить заказ»	Заказ успешно оформлен, появляется страница подтверждения заказа и уведомление	Заказ успешно оформлен

Таблица 11.8 – Оформление заказа (негативный)

<b>Название:</b>	Оформление заказа (негативный: пустая корзина)	
<b>Функция:</b>	Оформление заказа	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
<b>Предусловие:</b>		
Корзина пуста, пользователь авторизован	веб-приложение доступно и функционирует	Пользователь авторизован
<b>Шаги теста:</b>		

Нажать кнопку «Оформить заказ»	Выводится сообщение «Корзина пуста»	Заказ не оформлен
--------------------------------	-------------------------------------	-------------------

Таблица 11.9 – Просмотр заказа

<b>Название:</b>	Просмотр заказа	
<b>Функция:</b>	Страница заказов открывается	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
<b>Предусловие:</b>		
Заказ оформлен, пользователь авторизован	веб-приложение доступно и функционирует	Пользователь авторизован
<b>Шаги теста:</b>		
Перейти в раздел «Мои заказы»	Отображается список завершенных заказов	Страница с заказами пользователя
Выбрать доставленный заказ	Заказ выбран для просмотра	Заказ выбран
Нажать кнопку «Оставить отзыв»	Открывается форма просмотра заказа	Открыта форма просмотра заказа

Таблица 11.10 – Управление товарами

<b>Название:</b>	Управление товарами	
<b>Функция:</b>	Добавление нового товара в магазин	
<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
<b>Предусловие:</b>		

Продавец авторизован	веб-приложение доступно и функционирует	Пользователь авторизован
<b>Шаги теста:</b>		
Перейти в раздел «Мой магазин»	Страница магазина загружена	Страница загружена
Нажать кнопку «Добавить товар»	Открывается форма добавления нового товара	Появилась форма
Заполнить форму данными (название, описание, цена, изображение)	Форма заполнена корректно, все обязательные поля заполнены	Форма заполнена
Нажать кнопку «Сохранить товар»	Новый товар успешно добавлен, отображается в каталоге магазина	Товар зарегистрирован

Важно тестировать не только корректное поведение системы (позитивные кейсы), но и устойчивость к некорректным данным или действиям пользователя (негативные кейсы). Например, позитивный кейс регистрации проверяет ввод корректных данных, а негативный – обработку ошибки при неверном формате.

Тест-кейсы должны быть достаточно подробными, чтобы любой тестировщик мог повторить шаги без дополнительных разъяснений. Детализация включает описание конкретных действий, используемых данных и ожидаемых результатов на каждом этапе.

Каждый тест-кейс связывается с определёнными требованиями системы, что позволяет при необходимости отследить, какие функциональные или нефункциональные требования были проверены. Это облегчает последующую валидацию изменений и регрессионное тестирование.

## 11.2 Используемые инструменты тестирования



Для проведения комплексного тестирования маркетплейса применяются следующие инструменты:

- 1) Selenium WebDriver – автоматизация тестирования веб-интерфейса (регистрация, авторизация, поиск, оформление заказа, управление товарами).
- 2) Apache JMeter – нагрузочное тестирование для оценки времени отклика (<500 мс).
- 3) OWASP ZAP – тестирование безопасности для выявления уязвимостей (SQL-инъекции и др.) и проверки шифрования данных.

Выбор инструментов тестирования определяется характером системы, требованиями к её качеству и объемом функционала. Инструменты помогают автоматизировать рутинные проверки, проводить нагрузочное тестирование, а также обнаруживать уязвимости и проблемы совместимости.

### **11.3 Проведение тестирования функциональных и нефункциональных требований**

Функциональное тестирование фокусируется на проверке того, что каждая функция системы работает согласно техническому заданию.

Основные этапы:

- Подготовка тестовой документации: формируются тест-кейсы, чек-листы и матрица требований.
- Исполнение тестов: выполнение сценариев, как в ручном, так и в автоматизированном режиме.
- Анализ результатов: сопоставление фактических результатов с ожидаемыми, фиксация выявленных дефектов.

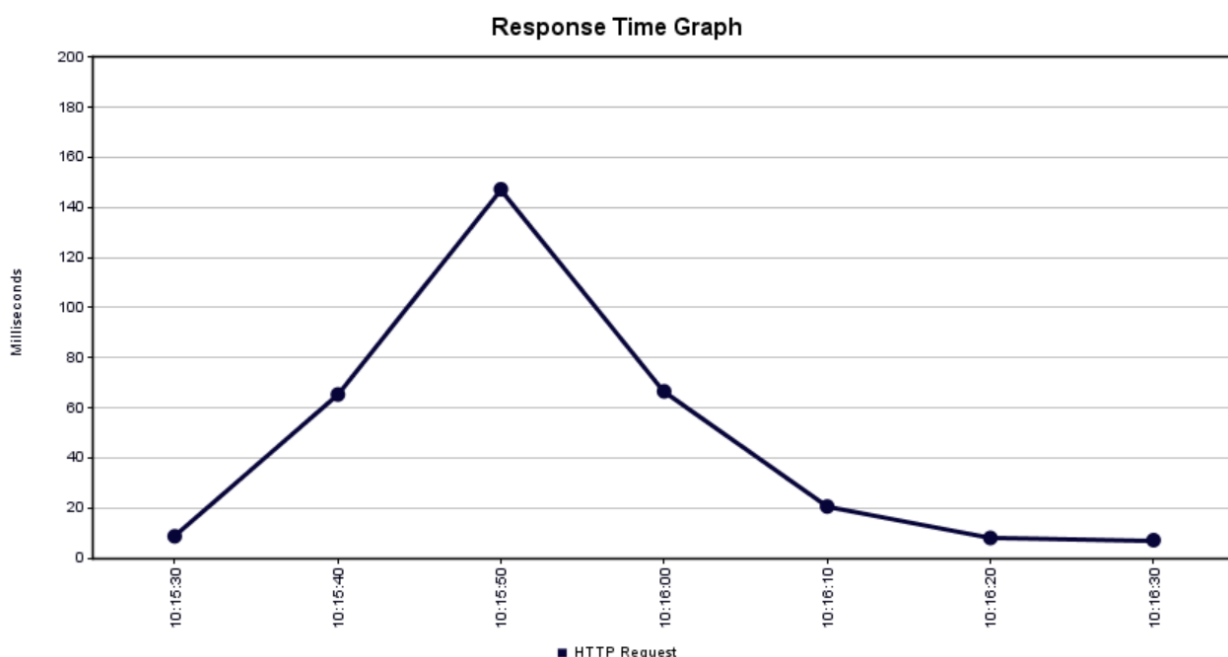
Нефункциональное тестирование ориентировано на проверку характеристик системы, которые не связаны напрямую с отдельными функциями, но влияют на качество продукта в целом.

Включает:

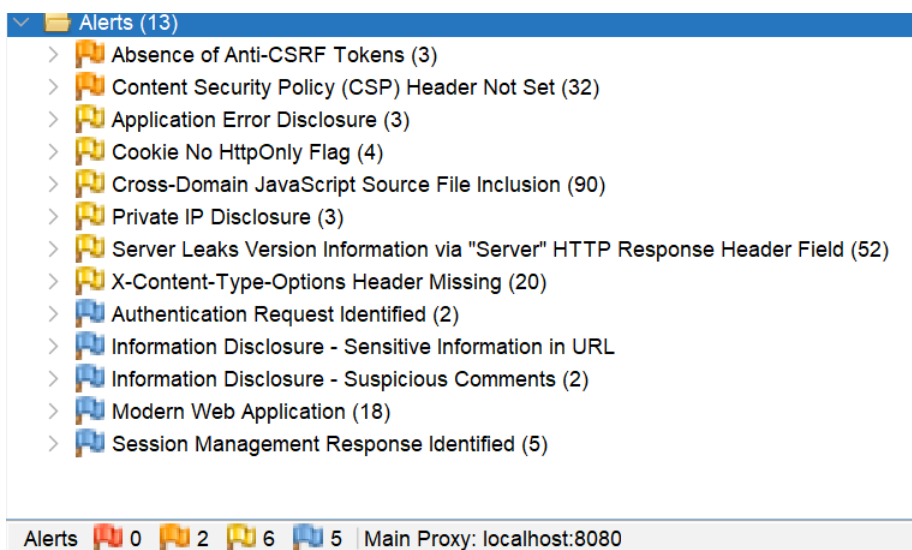
- Нагрузочное тестирование: проверка устойчивости при высоких объемах запросов и пользователей.

- Тестирование безопасности: анализ системы на наличие уязвимостей, оценка механизмов защиты данных.
- Тестирование совместимости, проверка работы приложения в различных браузерах и на разных устройствах.

В процессе тестирования используется циклический подход: планирование, выполнение тестов, анализ результатов и коррекция ошибок. Это позволяет быстро идентифицировать проблемные места и обеспечить непрерывное улучшение качества продукта.



**Рисунок 11.12 – Нагрузочное тестирование**



**Рисунок 11.13 – Тестирование защиты**

Нагрузочное тестирование с использованием Apache JMeter показало

время отклика системы менее 500 мс при высокой нагрузке. Надежность и доступность. Тестирование с помощью OWASP ZAP подтвердило защиту данных и отсутствие критичных уязвимостей.

#### **11.4 Уровни тестирования**

Для тестирования важно различать уровни тестирования, то есть различные ступени или подходы к тестированию ПО, которые обычно выполняются последовательно. Основные уровни тестирования:

1. **Модульное тестирование:** проверяются отдельные модули или компонентное наполнение ПО на корректность их функционирования. Тестовые случаи проводятся над отдельными функциями, процедурами или классами. В контексте маркетплейса это могла бы быть проверка корректности функции валидации данных пользователя или функции расчета налога для одной позиции товара.
2. **Интеграционное тестирование:** проверяется взаимодействие и взаимосвязь между различными модулями или компонентами ПО. Для маркетплейса это включает тестирование связи между модулем регистрации и базой данных пользователей, проверку потока добавления товара из карточки в корзину, или интеграцию модуля оформления заказа с внешними системами.
3. **Системное тестирование:** проводится на интегрированной системе в целом. Проходит проверка функциональности, производительности и поведения системы в соответствии с требованиями и ожиданиями. Применительно к маркетплейсу это может быть полный путь от регистрации и авторизации пользователя до поиска товара, добавления его в корзину и успешного оформления заказа, как иллюстрируют таблицы тест-кейсов
4. **Приемочное тестирование:** Проверка соответствия ПО конечным требованиям заказчика или пользователя. Для подтверждения готовности продукта к использованию от заказчика. Для

маркетплейса это может означать проверку со стороны представителей заказчика или группы пользователей, насколько удобно и интуитивно понятен весь процесс покупки, насколько корректно отображаются цены и опции доставки, и решает ли платформа их задачи.

5. **Регрессионное тестирование:** Тестирование после внесения изменений в готовое ПО для обнаружения новых дефектов или нежелательных побочных эффектов, возникших в результате внесенных изменений. Для маркетплейса это означает повторный прогон существующих тест-кейсов и сценариев с предыдущих уровней для проверки критически важных функций.
6. **Альфа- и бета-тестирование:** Тестирование перед релизом продукта и выполняются в контролируемой среде разработчика (альфа-тестирование) и в реальной среде пользователя (бета-тестирование). Обратная связь с этих этапов помогает выявить неочевидные проблемы и улучшить продукт перед его широким выпуском.

Таким образом, различные уровни тестирования в практической работе формируют всестороннюю стратегию обеспечения качества, начиная от проверки отдельных частей кода и заканчивая валидацией продукта с точки зрения конечного пользователя и бизнеса.

### **11.5 Чек-лист тестирования**

Чек-лист представляет собой контрольный список задач или тестовых случаев, которые необходимо выполнить для полного покрытия требований системы. Это своего рода карта тестировщика, которая помогает не пропустить ни один важный аспект проверки.

Каждый пункт чек-листа соответствует определенному функциональному или нефункциональному требованию. Чек-лист включает короткие описания тестов без подробных инструкций, но с указанием ключевых этапов, которые должны быть проверены. Чек-лист позволяет фиксировать статус выполнения каждого тестового случая (например,

выполнен, не выполнен, выявлены проблемы). Это обеспечивает прозрачность процесса тестирования и позволяет оценивать общий уровень готовности продукта.

Таблица 11.11 – Чек лист тестирования

№	Наименование тестового случая	Действия / Входные данные	Ожидаемый результат	Фактический результат	Примечание
1	Регистрация	Логин: user@africa.com; Пароль: 123456	Успешная регистрация	Успешная регистрация	Прошел корректно
2	Вход в систему (позитивный)	Логин: user@africa.com; Пароль: 123456	Успешный вход, переход в личный кабинет	Успешный вход	Тест выполнен и автоматизирован.
3	Вход в систему (негативный)	Логин: user@africa.com; Пароль: неверный	Сообщение «Ошибка аутентификации»	Сообщение об ошибке	Валидация выполнена корректно
4	Поиск товаров (позитивный)	Ввод для поиска: «феромонов ловушки»;	Список товаров, соответствующих критериям	Результаты корректны	Тестирование проведено вручную и автоматизировано
5	Поиск товаров (негативный)	Ввод: «неизвестный товар»	Сообщение «Товар не найден»	Сообщение получено	Обработка отсутствующих результатов подтверждена
6	Добавление товара в корзину (позитивный)	Нажатие кнопки «Добавить в корзину» на странице товара	Товар добавлен, счетчик в корзине увеличен	Товар добавлен	Тест пройден
7	Оформление заказа (позитивный)	Корзина содержит товары; заполнение формы (адрес, способ	Заказ оформлен, появляется страница подтверждения заказа	Заказ оформлен	Тест выполнен автоматизированно и вручную

		оплаты: кредитная карта)			
8	Оформлен ие заказа (негативны й)	Корзина пуста; нажатие кнопки «Оформить заказ»	Сообщение «Корзина пуста»	Сообщени е получено	Защита от оформления пустого заказа подтверждена
9	Отслежива ние статуса заказа	Авторизация ; переход в раздел «Мои заказы»; выбор заказа для просмотра	Отображение текущего статуса заказа (например, «В пути»)	Статус отображае тся	Тест выполнен успешно
1 0	Управлени е товарами (продавец)	Продавец входит в «Мой магазин», нажимает «Добавить товар», заполняет данные, сохраняет товар	Новый товар добавлен, отображается в каталоге магазина	Товар добавлен	Тест выполнен, данные в базе сохранены корректно

Чек-лист может обновляться в процессе тестирования при появлении новых функций или изменении требований. Это помогает поддерживать актуальность документации и своевременно реагировать на изменения. По результатам выполнения чек-листа формируется сводный отчет о проделанной работе, что позволяет руководителям проекта оценить качество продукта и принять решение о его релизе.

### **11.6 Дополненная матрица требований**

Матрица требований – это таблица, в которой документируются все функциональные и нефункциональные требования системы, а также результаты тестирования для каждого из них. Она служит инструментом для отслеживания соответствия реализованных функций заявленным

требованиям.

Таблица 11.12 – Матрица требований

№	Требование	Суть	Автор	Критерий	Компоненты архитектуры	Результат тестирования
1	Функционал					
1.1	Регистрация пользователя	Пользователь должен иметь возможность зарегистрироваться на сайте через email, телефон или соцсети	Чахнин М.	Регистрация успешна, учетная запись создана	Django, PostgreSQL	Требование выполнено (рисунок 11.4)
1.2	Авторизация пользователя	Пользователь должен иметь возможность авторизовываться на сайте с помощью email, телефон или через соцсети	Чахнин М.	Вход успешный в существующую учетную запись	Django, PostgreSQL	Требование выполнено (рисунок 11.4)
1.3	Поиск товаров	Должен быть реализован поиск товаров по сайту с возможностью фильтрации и	Чахнин М.	Поиск возвращает релевантные товары	Django, PostgreSQL	Требование выполнено (рисунок 11.5)

		сортировки				
1.4	Оформление заказа	Пользователь должен иметь право оформить заказ, выбрать доставку и оплатить товар	Чахнин М.	Заказ оформлен, оплата проведена	Django , PostgreSQL	Требование выполнено (рисунок 11.7)
1.5	Личный кабинет	В личном кабинете должны отображаться заказы, избранные товары и настройки профиля	Чахнин М.	Данные отображаются корректно, профиль можно настроить	Django , PostgreSQL	Требование выполнено (рисунок 11.8)
2	Юзабилити					
2.1	Адаптивный дизайн	Сайт должен корректно отображаться на любых устройствах (мобильных телефонах, планшетах, компьютерах)	Смирнов Д.	Визуальный осмотр, тестирование на всех устройствах	Django	Требование выполнено (рисунок 11.10)
2.2	Простой процесс покупки	Понятный интерфейс, минимум шагов	Смирнов Д.	Время на оформление заказа не	Django	Мануальное тестиров



		при оформлении заказа		превышает 3 минуты		ание подтверждено пригодность
2.3	Отзывы и рейтинг	Возможность оставлять отзывы и ставить рейтинг товарам	Смирнов Д.	Пользователь может оставить оценку и отзыв	Django	Требование выполнено (рисунок 11.10)
3	Производительность					
3.1	Быстрая загрузка страниц	Время загрузки страницы не должно превышать время в 4 секунды	Макаров А.	Тестирование производительности	Django , PostgreSQL	Требование выполнено (рисунок 11.2)
3.2	Оптимизированные изображения	Все изображения должны быть загружены в сжатом формате, без потери качества	Макаров А.	Проверка формата изображения и скорости загрузки	Django , PostgreSQL	Требование выполнено (рисунок 11.3)
4	Безопасность					
4.1	Защита данных пользователя	Должно быть реализовано шифрование	Ким К.	Проверка уязвимости системы	Django , PostgreSQL	Требование выполнено

	телей	данных, защита от SQL-инъекций и XSS-атак			SQL	но (рисунок 11.3)
4.2	HTTPS	Сайт должен работать только по https	Ким К.	Проверка сертификато в безопасност и	Django , PostgreSQL, Nginx	Необходимо внедрить при развертывании
5	Интеграции					
5.1	Интеграция платёжных систем	Поддержка популярных платежных систем (Visa, Mastercard, PayPal, Мир и т. д.)	Смирнов Д.	Успешное проведение тестовых транзакций	Django , PostgreSQL SQL	В данной версии продукта не реализовано
5.2	Интегрированная логистика	Автоматический расчет стоимости всех товаров в корзине и сроков доставки	Смирнов Д.	Проверка корректности и расчета стоимости заказа (товаров и доставки)	Django , PostgreSQL SQL	Требование выполнено (рисунок 11.9)
6	SEO и маркетинг					
6.1	SEO-оптимизация	Использование мета-тегов	Ломакин Д.	Анализ с помощью Google		Планируется на этапе

				Search Console		развертывания
6.2	Email-рассылки	Автоматически email-уведомления о заказах, акциях, предложениях	Ломакин Д.	Рассылка отправляется в нужный момент	Django , PostgreSQL	В данной версии продукта не реализовано

Матрица требований помогает выявить несоответствия между документированными требованиями и фактической реализацией. При изменении функционала или выявлении дефектов, матрица обновляется, что позволяет поддерживать актуальность информации на протяжении всего жизненного цикла проекта.

Связь между тест-кейсами и требованиями позволяет четко проследить, какие именно аспекты системы были протестированы. Это помогает в регрессионном тестировании и упрощает анализ качества продукта при внесении изменений.

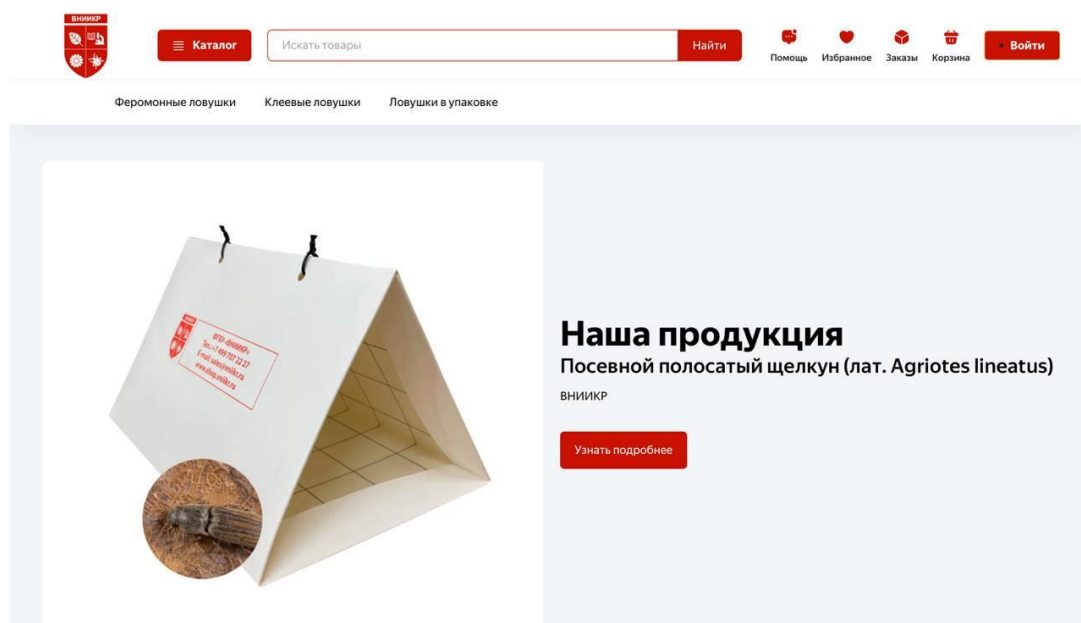


Рисунок 11.14 – Отображение сайта

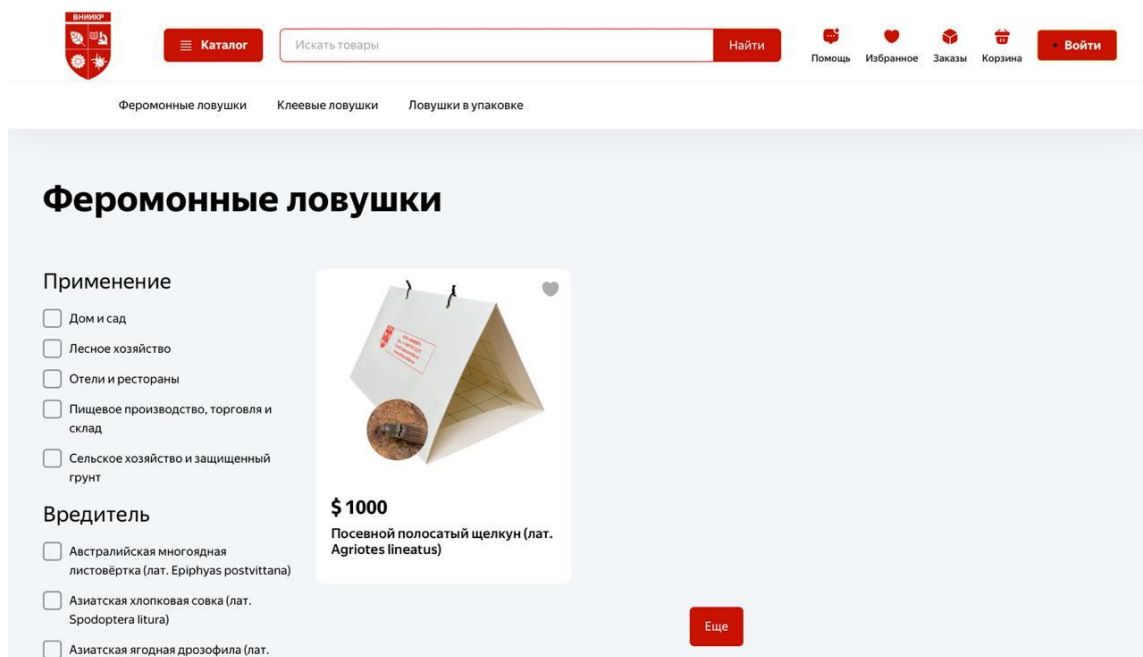


Рисунок 11.15 – Поиск по категории

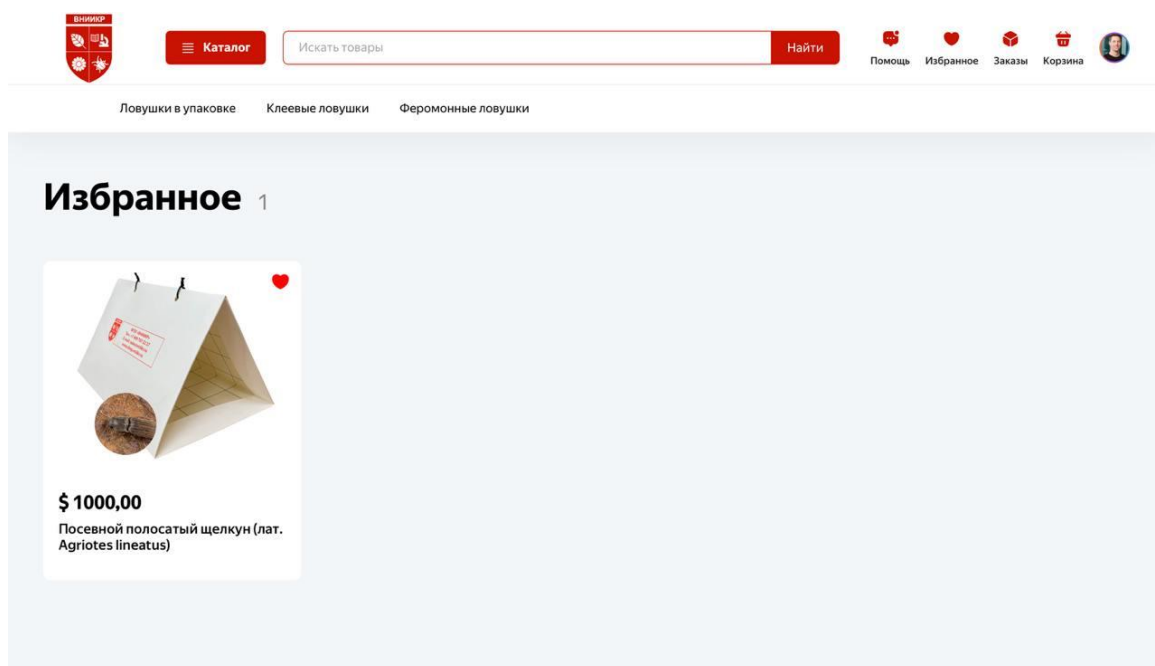
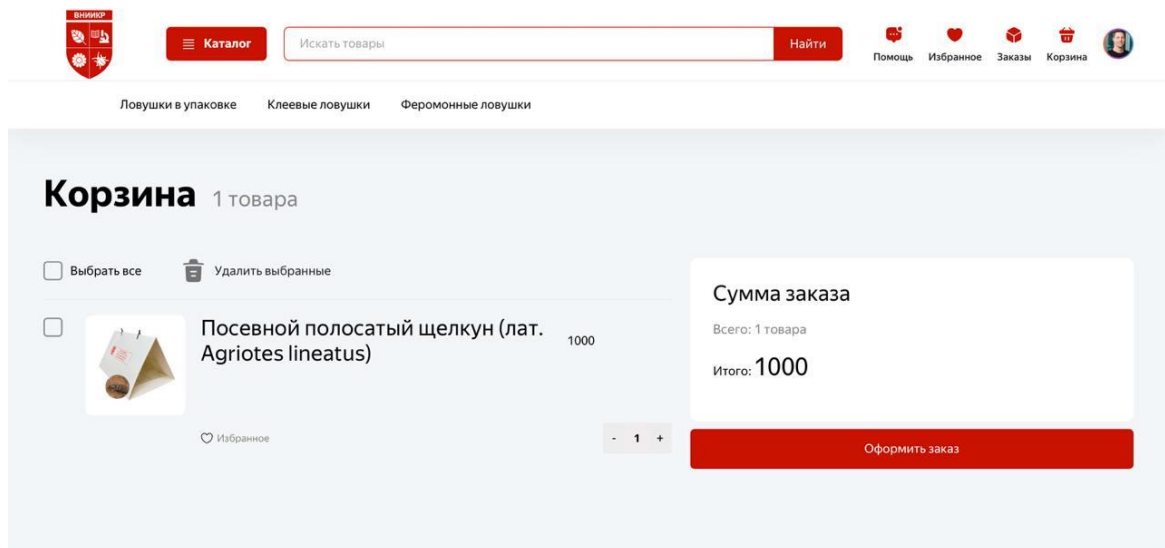


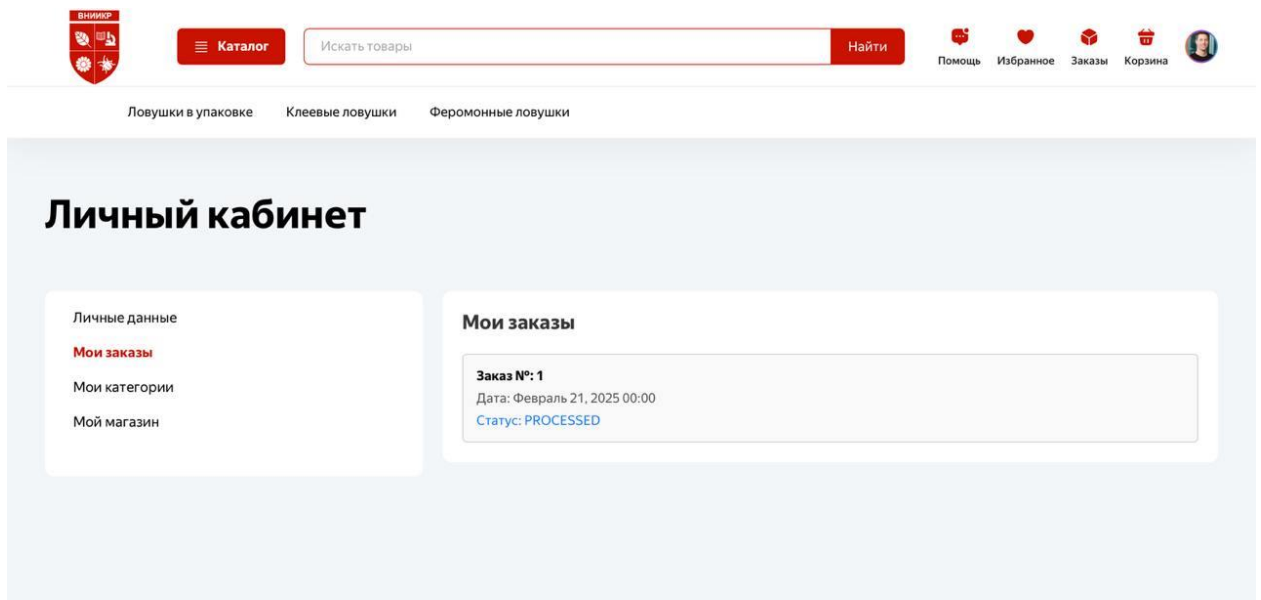
Рисунок 11.16 – Добавление в избранное



Поддержка

Сотрудничество

Рисунок 11.17 – Проверка корзины



Поддержка

Сотрудничество

Рисунок 11.18 – Отслеживание статусов заказов

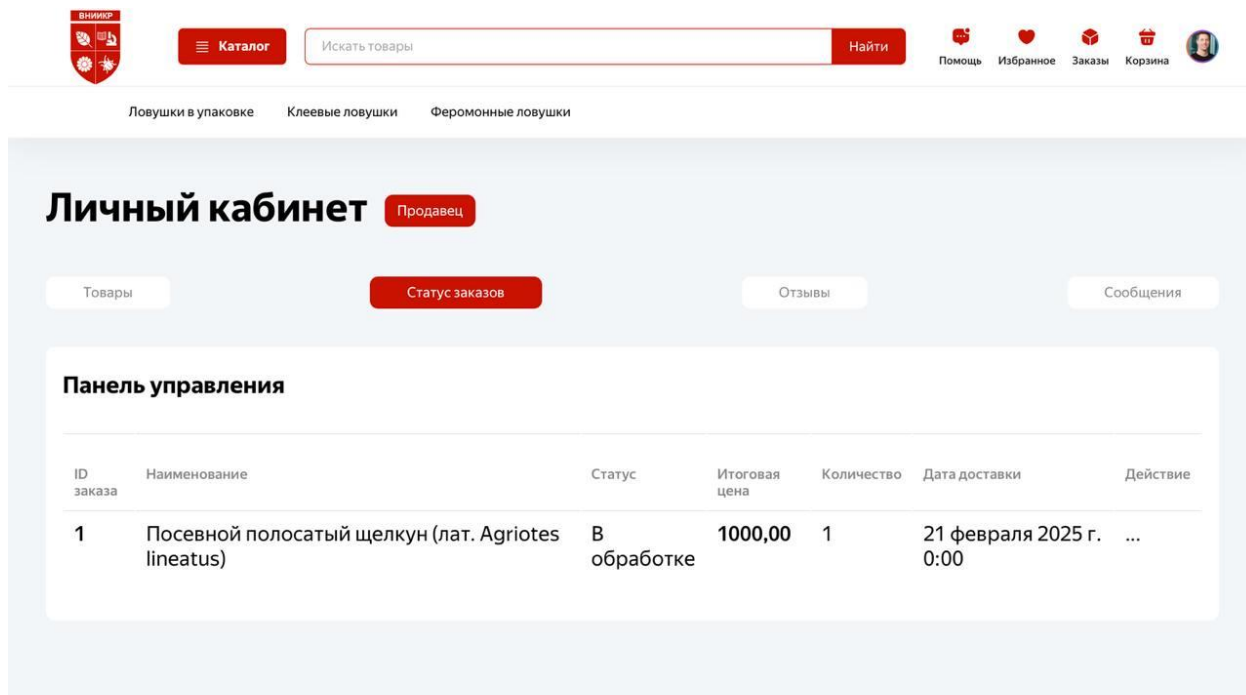


Рисунок 11.19 – Отслеживание заказов

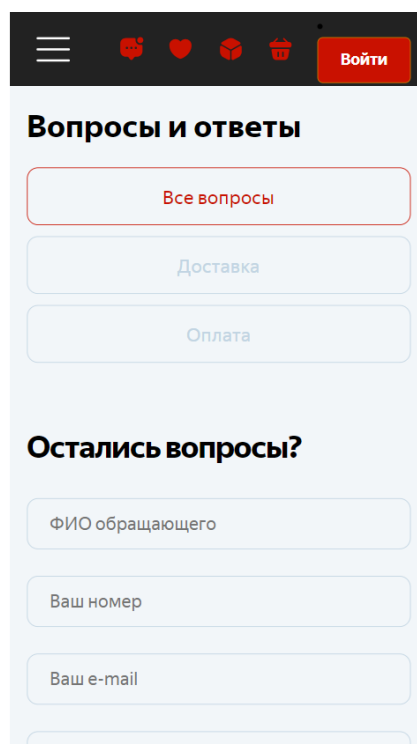


Рисунок 11.20 – Адаптивный дизайн для мобильных устройств

```
DevTools listening on ws://127.0.0.1:59953/devtools/browser/4188268a-79ce-4672-b193-c538329f4f8a
test_T11_10_seller_add_product_positive (__main__.SeleniumPureMockTests) ... ok
test_T11_1_user_registration_positive (__main__.SeleniumPureMockTests) ... ok
test_T11_2_user_login_positive (__main__.SeleniumPureMockTests) ... ok
test_T11_3_user_login_negative (__main__.SeleniumPureMockTests) ... ok
test_T11_4_product_search_positive (__main__.SeleniumPureMockTests) ... ok
test_T11_5_product_search_negative (__main__.SeleniumPureMockTests) ... ok
test_T11_6_add_to_cart_positive (__main__.SeleniumPureMockTests) ... ok
test_T11_7_checkout_positive (__main__.SeleniumPureMockTests) ... ok
test_T11_8_checkout_negative_empty_cart (__main__.SeleniumPureMockTests) ... ok
test_T11_9_leave_review_positive (__main__.SeleniumPureMockTests) ... ok

-----
Ran 10 tests in 10.465s

OK
```

**Рисунок 11.21 – Успешное прохождение тестов**

# ПРАКТИЧЕСКАЯ РАБОТА №12

## Аннотация

Программный продукт «Маркетплейс для жителей Африки» (далее – MAF) разработан организацией «Юные Орлы» базирующейся в РТУ МИРЭА и предназначен для предоставления удобной онлайн-платформы, позволяющей жителям африканских городов искать, сравнивать и приобретать товары. Документ подготовлен в 2025 году, версия 1.0, и утвержден ответственным лицом. Его назначение – зафиксировать порядок проведения испытаний системы, определить перечень проверяемых требований, описать методику испытаний и установить критерии приемки системы с целью подтверждения её готовности к опытной эксплуатации.

## 1. ОБЪЕКТ ИСПЫТАНИЙ

### 1.1. Наименование системы

«Маркетплейс для жителей Африки» (MAF).

### 1.2. Комплектность

Комплектность (ГОСТ 19). Система включает следующие модули:

- Модуль поиска и фильтрации товаров;
- Модуль сравнения товаров;
- Модуль оформления заказов (корзина, оплата, доставка);
- Модуль отслеживания заказов;
- Модуль отзывов и рейтингов;
- Административный модуль (управление ассортиментом, модерация, аналитика).

### 1.2. Область применения

Область применения (ГОСТ 34). Онлайн-платформа для поиска, сравнения и покупки товаров жителями африканских городов.

## 2. ЦЕЛЬ ИСПЫТАНИЙ

Цель испытаний – подтверждение соответствия системы требованиям ТЗ и нормативным документам, а также оценка её работоспособности,



надежности и безопасности. В ходе испытаний будут:

- Демонстрироваться функциональные возможности (от 70% до 100% требований, доступных для демонстрации).
- Проводиться нагрузочное тестирование для оценки производительности.
- Проверяться механизмы безопасности и корректность работы интерфейса.
- Оформляться итоговый акт приемочных испытаний для передачи системы в опытную эксплуатацию.

### **3. ОБЩИЕ ПОЛОЖЕНИЯ**

#### **3.1. Перечень руководящих документов**

Испытания проводятся на основании:

- Технического задания (ТЗ) на разработку МАФ;
- Технического проекта (архитектура, диаграммы бизнес-процессов, описание модулей);
- Руководств пользователя и администратора;
- Протоколов предыдущих испытаний (при наличии).

#### **3.2. Место и продолжительность испытаний**

Место проведения: тестовая площадка заказчика, имитирующая реальные условия эксплуатации.

Продолжительность: согласно утверждённому графику (сроки определяются в приказе заказчика).

#### **3.3. Организации, участвующие в испытаниях**

Исполнитель: ООО ЮНЫЕ ОРЛЫ (разработчик и тестировщик).

Заказчик: РТУ МИРЭА.

Дополнительные организации: лаборатории тестирования, специалисты по безопасности.

#### **3.4. Перечень предъявляемых на испытания документов**

На испытания предъявляются:

- Техническое задание;

- Технический проект;
- Руководства пользователя и администратора;
- Документация по установке и сопровождению;
- Протоколы предварительных испытаний.

## **4. ОБЪЁМ ИСПЫТАНИЙ**

### **4.1. Перечень этапов испытаний и проверок**

Испытания включают:

- Проверку документации;
- Испытания программных и технических средств.

### **4.2. Испытания функциональных блоков**

Испытания подсистемы 1 (модуль поиска и фильтрации товаров): проверка корректности поиска, фильтрации, пагинации и сортировки.

Испытания подсистемы 2 (модуль оформления заказов): проверка добавления товаров в корзину, оформления заказа, выбора способа оплаты и доставки.

### **4.3. Испытания структурированных данных (ГОСТ 34)**

Испытания подсистемы хранилища структурированных данных и метаданных: проверка корректности хранения, выборки и отображения данных.

### **4.4. Испытания электронных документов (ГОСТ 34)**

Проверка ввода, обработки, поиска и аннотирования электронных документов.

### **4.5. Испытания информационного обмена и интеграции (ГОСТ 34)**

Оценка корректности интеграции с внешними системами (платёжные системы, уведомления).

### **4.6. Испытания информационных киосков (ГОСТ 34)**

Проверка полноты и корректности предоставляемой информации для пользователей киосков.

## **5. МЕТОДИКА ПРОВЕДЕНИЯ ИСПЫТАНИЙ**

Для проведения испытаний разработаны детальные тест-кейсы и

сценарии. Ниже представлена таблица, включающая примеры методов проверки функциональности системы.

Таблица 12.1 – Испытания

№	Тест-кейс	Шаги выполнения	Ожидаемый результат
1	Поиск товаров	1. Ввести ключевое слово в поисковую строку 2. Нажать "Поиск"	Отображается список релевантных товаров с корректной пагинацией и сортировкой
2	Применение фильтров	1. Выбрать категорию, установить диапазон цены 2. Нажать "Применить"	Товары отображаются с учетом заданных фильтров
3	Добавление товара в корзину и оформление заказа	1. Выбрать товар 2. Нажать "Добавить в корзину" 3. Перейти в корзину, оформить заказ, выбрать способ оплаты	Товар успешно добавлен в корзину, заказ оформлен, выбран способ оплаты и доставки
4	Отслеживание статуса заказа	1. Перейти в личный кабинет 2. Выбрать заказ 3. Просмотреть статус	Статус заказа отображается корректно: оформлен, оплачен, в пути, доставлен
5	Оставление отзыва и рейтинга после покупки	1. Перейти на страницу товара 2. Нажать "Оставить отзыв" 3. Ввести текст и оценку, сохранить	Отзыв опубликован, оценка сохранена, информация отображается корректно

Кроме таблицы функционального тестирования, методика включает

дополнительные этапы.

Нагрузочное тестирование:

- Моделирование до 10 000 одновременных запросов с использованием JMeter.
- Контроль времени отклика (не более 500 мс) и нагрузка на сервер.

Тестирование безопасности:

- Проведение тестов на SQL-инъекции, XSS, имитация DDoS-атак.
- Проверка работы шифрования (AES-256) и двухфакторной аутентификации.

Тестирование пользовательского интерфейса:

- Проверка адаптивности и корректного отображения интерфейса на ПК, планшетах и смартфонах.

## **6. ТРЕБОВАНИЯ ПО ИСПЫТАНИЯМ ПРОГРАММНЫХ СРЕДСТВ**

Испытания программных средств направлены на:

- Проверку соответствия реализованных функций требованиям ТЗ.
- Замер времени отклика (не более 500 мс при 10 000 одновременных запросах).
- Проверку устойчивости системы, корректности обработки ошибок и отказоустойчивости.
- Тестирование средств логирования и мониторинга.

## **7. ПЕРЕЧЕНЬ РАБОТ, ПРОВОДИМЫХ ПОСЛЕ ЗАВЕРШЕНИЯ ИСПЫТАНИЙ**

После завершения испытаний выполняются следующие работы:

- Анализ результатов испытаний и составление сводных протоколов.
- Формирование заключения о соответствии системы требованиям ТЗ.
- Оформление Акта предварительных и приемочных испытаний.
- При необходимости внесение доработок и повторное тестирование.
- Подготовка итогового отчёта с рекомендациями по улучшению.

## **8. УСЛОВИЯ И ПОРЯДОК ИСПЫТАНИЙ**

Испытания проводятся в тестовой среде, максимально приближенной к рабочим условиям эксплуатации.

Система тестируется на оборудовании, указанном в ТЗ, с использованием специализированных измерительных средств.

Порядок испытаний включает:

- Предварительное тестирование отдельных модулей.
- Комплексное тестирование интеграции модулей.
- Нагрузочное тестирование с имитацией до 10 000 пользователей.
- Тестирование безопасности (включая шифрование, двухфакторную аутентификацию, имитацию атак).
- Проведение опытной эксплуатации и приемочных испытаний с участием заказчика.

## **9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ИСПЫТАНИЙ**

Испытания проводятся на серверном оборудовании заказчика, рабочих ПК и мобильных устройствах.

Используются специализированные программные средства (Selenium, JMeter, ELK-стек, Prometheus) для автоматизированного тестирования, логирования и мониторинга.

## **10. МЕТРОЛОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ИСПЫТАНИЙ**

Для точного замера времени отклика, нагрузки и других параметров применяются измерительные приборы и специализированное ПО.

Метрологическая проверка проводится согласно утверждённым методикам.

## **11. ОТЧЁТНОСТЬ**

### **11.1. Протоколирование результатов**

Для каждого этапа испытаний составляются детальные протоколы, включающие:

- Перечень выполненных тест-кейсов и сценариев.

- Описание используемых технических и программных средств.
- Сравнение ожидаемых и фактических результатов.
- Замечания, выявленные отклонения и рекомендации по доработке.

### **11.2. Заключительный акт**

По завершении испытаний оформляется «Акт предварительных и приемочных испытаний», подтверждающий соответствие системы требованиям ТЗ и готовность к эксплуатации.

### **11.3. Отчёт перед заказчиком**

Итоговый отчёт включает сводную таблицу результатов испытаний, аналитический раздел с выводами по работоспособности, безопасности и производительности системы, а также рекомендации по улучшению.

## **ЗАКЛЮЧЕНИЕ**

Настоящая Программа и методика испытаний разработана для всесторонней проверки программного продукта «Маркетплейс для жителей Африки» в полном соответствии с нормативными требованиями ГОСТ 19.301-79, ГОСТ 19.105-78 и ГОСТ 34.603-92.

## **ПРАКТИЧЕСКАЯ РАБОТА №13**

### **13.1 Введение в теорию управления рисками**

Управление рисками является неотъемлемой частью современного проектного менеджмента и системной инженерии. Цель анализа рисков – заблаговременно выявить потенциальные угрозы, оценить их вероятность и последствия, а также разработать эффективные меры реагирования для минимизации негативного влияния на проект. Такой подход позволяет:

1. Снизить неопределённость: путём систематического анализа рисков можно подготовиться к возможным проблемам и заранее определить способы их решения.
2. Оптимизировать затраты: принятие корректирующих мер на ранних этапах разработки помогает избежать больших финансовых и временных потерь в будущем.
3. Повысить качество продукта: чётко структурированная система управления рисками способствует стабильности и надёжности конечного продукта, что особенно важно для проектов с высокой конкуренцией.
4. Обеспечить прозрачность процессов: документирование рисков и мер реагирования улучшает коммуникацию внутри команды и с заказчиком, создавая благоприятную атмосферу для совместного решения проблем.

Эти принципы широко применяются в IT-проектах, где изменения требований, технические сбои, проблемы коммуникации и другие факторы могут существенно повлиять на успех разработки.

### **13.2 Описание рисков**

Во-первых, необходимо описать риски в формате «причина-риск-эффект» (пример представлен в табл. 13.1).

Таблица 13.1 – Причина-риск-эффект

№	Риск (причина–риск–эффект)
1	<ul style="list-style-type: none"> <li>– Требования к проекту могут меняться в процессе со стороны преподавателя/куратора ("заказчика")</li> <li>– Нам приходится переделывать или добавлять функционал</li> <li>– Тратим дополнительное время, можем не уложиться в изначально запланированные сроки.</li> </ul>
2	<ul style="list-style-type: none"> <li>– Мы недостаточно эффективно обмениваемся информацией и статусами задач внутри команды.</li> <li>– Может возникнуть недопонимание, дублирование работы или упущение важных деталей.</li> <li>– Снижение качества финального продукта, задержки в выполнении задач.</li> </ul>
3	<ul style="list-style-type: none"> <li>– Мы не ведем подробную и актуальную документацию по проекту (принятые решения, инструкции).</li> <li>– Будет сложно разобраться в проекте позже, новым участникам будет трудно включиться, могут возникнуть ошибки при передаче знаний</li> <li>– Потеря времени на восстановление информации, потенциальные ошибки при доработке или демонстрации.</li> </ul>
4	<ul style="list-style-type: none"> <li>– Среда разработки (наши компьютеры, виртуальные машины, используемые онлайн-сервисы) может быть нестабильной.</li> <li>– Могут происходить сбои в работе инструментов, проблемы с развертыванием или запуском проекта.</li> <li>– Потеря времени на решение проблем с окружением, задержки в разработке и тестировании.</li> </ul>
5	<ul style="list-style-type: none"> <li>– Кто-то из членов команды может временно или полностью выбыть из работы над проектом по личным причинам или из-за учебной нагрузки.</li> <li>– У нас может не хватить ресурсов или специфических знаний для выполнения определенных задач.</li> </ul>



	<ul style="list-style-type: none"> <li>– Замедление темпа разработки, неравномерное распределение нагрузки, риск невыполнения запланированного функционала.</li> </ul>
6	<ul style="list-style-type: none"> <li>– Наш проект может требовать интеграции с внешними API или сторонними сервисами.</li> <li>– Эти внешние сервисы могут быть нестабильны, менять свои API или иметь ограничения.</li> <li>– Задержки в интеграции, необходимость доработки нашего кода из-за изменений у провайдера сервиса, потенциальное невыполнение функционала, зависящего от внешнего сервиса.</li> </ul>
7	<ul style="list-style-type: none"> <li>– Мы не уделяем достаточного внимания вопросам безопасности при разработке, особенно при работе с данными.</li> <li>– В проекте могут появиться уязвимости, позволяющие получить несанкционированный доступ или нарушить работу.</li> <li>– Потенциальные проблемы с данными.</li> </ul>
8	<ul style="list-style-type: none"> <li>– Мы используем сторонние библиотеки, фреймворки или компоненты в проекте.</li> <li>– В этих компонентах могут быть баги, уязвимости или они могут перестать поддерживаться.</li> <li>– Необходимость поиска обходных решений или замены компонентов, потенциальные проблемы со стабильностью или безопасностью нашего проекта.</li> </ul>
9	<ul style="list-style-type: none"> <li>– Появляется новая информация, инструменты или подходы, которые делают наш текущий подход менее оптимальным или актуальным.</li> <li>– Наш проект может перестать быть уникальным или использовать устаревшие решения.</li> <li>– Необходимость пересмотра части решений или стратегии, что может привести к дополнительной работе и задержкам.</li> </ul>
10	<ul style="list-style-type: none"> <li>– Архитектура проекта не рассчитана на потенциальный рост объема данных или количества.</li> </ul>

	<ul style="list-style-type: none"> <li>– Проект будет работать медленно или некорректно.</li> <li>– Низкая производительность.</li> </ul>
--	---

Ниже представлена таблица, в которой описаны 10 рисков для проекта, сформулированных по схеме «причина–риск–эффект». Каждому риску присвоены оценки вероятности и последствий по 10-балльной шкале, а также вычислена важность (произведение вероятности и последствий).

Таблица 13.2 – Описание рисков

№	Риск (причина–риск–эффект)	Вероятность (1–10)	Последствия (1–10)	Важность (В*П)
1	<p><b>Изменение требований клиента.</b></p> <ul style="list-style-type: none"> <li>- Клиент часто вносит корректировки (причина)</li> <li>- изменение функционала в процессе разработки (риск)</li> <li>- задержки и увеличение стоимости проекта (эффект).</li> </ul>	4	6	24
2	<p><b>Недостаточная коммуникация внутри команды.</b></p> <ul style="list-style-type: none"> <li>- Отсутствие регулярных встреч и обмена информацией (причина)</li> <li>- недопонимание задач и ошибок в реализации (риск)</li> <li>- снижение качества продукта и задержки (эффект).</li> </ul>	2	7	14
3	<p><b>Неполнота или плохая документация.</b></p>	7	5	35

	<ul style="list-style-type: none"> <li>- Недостаточное ведение и обновление проектной документации (причина)</li> <li>- отсутствие четких требований и историй проекта (риск)</li> <li>- ошибки в реализации и трудности при передаче проекта новым участникам (эффект).</li> </ul>			
4	<p><b>Технические проблемы с сервером.</b></p> <ul style="list-style-type: none"> <li>- Использование нестабильных серверов или оборудования (причина)</li> <li>- частые сбои и downtime (риск)</li> <li>- нарушение доступности маркетплейса для пользователей (эффект).</li> </ul>	9	9	81
5	<p><b>Незапланированное отсутствие ключевых специалистов.</b></p> <ul style="list-style-type: none"> <li>- Болезни или личные обстоятельства членов команды (причина)</li> <li>- потеря экспертного ресурса (риск)</li> <li>- задержки в выполнении критичных задач (эффект).</li> </ul>	4	8	32

6	<p><b>Проблемы с платежными системами.</b></p> <ul style="list-style-type: none"> <li>- Технические сбои у партнёров-провайдеров платежей (причина)</li> <li>- нестабильность или недоступность системы оплаты (риск)</li> <li>- финансовые потери и недовольство клиентов (эффект).</li> </ul>	3	9	27
7	<p><b>Уязвимости в безопасности.</b></p> <p>Недостаточное внимание к вопросам защиты данных (причина)</p> <ul style="list-style-type: none"> <li>- появление уязвимостей в системе (риск)</li> <li>- утечка пользовательских данных и ущерб репутации (эффект).</li> </ul>	8	9	72
8	<p><b>Проблемы с интеграцией сторонних сервисов.</b></p> <ul style="list-style-type: none"> <li>- Сложности при подключении логистических и аналитических модулей (причина)</li> <li>- задержки в интеграции и возникновение ошибок (риск)</li> <li>- снижение эффективности работы платформы (эффект).</li> </ul>	7	7	49

9	<b>Конкурентное давление и изменение рынка.</b> - Появление новых конкурентов и быстрое - изменение тенденций (причина) необходимость пересмотра стратегии (риск) - возможное снижение доли рынка и прибыли (эффект).	6	8	48
10	<b>Проблемы масштабируемостью.</b> -Недостаточное проектирование архитектуры с учётом роста нагрузки (причина) -Снижение производительности при резком росте числа пользователей (риск) - ухудшение пользовательского опыта и потеря клиентов (эффект).	7	9	63

В анализе каждая оценка была установлена с учётом экспертных мнений на опыте предыдущих проектов. Команда проекта, включая менеджера, аналитика и технических специалистов, обсуждала каждый риск и совместно приходила к оценкам, что позволяет отразить как техническую сложность, так и внешние факторы (например, изменение требований со стороны клиента или рыночные изменения).

### 13.3 Матрица рисков

Для наглядного представления распределения рисков составлена матрица, в которой строки соответствуют уровню вероятности, а столбцы – уровню ущерба.

Таблица 13.3 – Матрица рисков

<b>ВЕРОЯТНОСТЬ</b>	<b>УРОВЕНЬ УЩЕРБА</b>				
	<b>Несущественные (1–2)</b>	<b>Низкие (3–4)</b>	<b>Средние (5–6)</b>	<b>Существенные (7–8)</b>	<b>Катастрофические (9–10)</b>
<b>Весьма вероятно (9–10)</b>	–	–	–	–	4
<b>Вероятно (7–8)</b>	–	–	–	2, 5, 8	7, 10
<b>Возможно (5–6)</b>	–	–	–	9	–
<b>Маловероятно (3–4)</b>	–	–	1	5	6
<b>Крайне маловероятно (1–2)</b>	–	–	–	2	–

### 13.4 Планы реагирования на риски

Для каждого риска разработаны стратегии реагирования, включающие основной план (меры по управлению риском) и отходной план (резервный вариант на случай неэффективности основного).

Таблица 13.4 – План реагирования

<b>№</b>	<b>Риск (причина–риск–эффект)</b>	<b>Стратегия реагирования</b>	<b>Основной план</b>	<b>Отходной план</b>
1	Изменение	Минимизация	Проводить	Заложить

	требований клиента		регулярные встречи с клиентом, уточнять и документировать изменения требований, согласовывать изменения с командой.	резерв времени для переработки и корректировок и проекта.
2	Недостаточная коммуникация внутри команды	Минимизация	Организовать ежедневные/еженедельные планерки, использовать современные системы обмена сообщениями (Slack, Microsoft Teams) и вести протоколы встреч.	Провести внеплановую сессию обмена информацией и назначить ответственного за коммуникацию.
3	Неполнота или плохая документация	Минимизация	Внедрить систему ведения документации (Confluence, Wiki), регулярно обновлять и проводить ревизию документации.	Назначить ответственного за документацию и провести внеплановое обновление ключевых разделов.
4	Технические проблемы с сервером	Страхование (передача)	Использовать облачные сервисы с гарантированным uptime (например, AWS, Azure) и заключить SLA с провайдером.	Настроить резервный сервер и план аварийного восстановления (DRP).
5	Незапланированное отсутствие ключевых специалистов	Минимизация/Принятие	Организовать перекрёстное обучение и документирование процессов, иметь резервного кандидата для каждой ключевой позиции.	Пересмотреть сроки проекта и перераспределить задачи между оставшимися членами команды.

6	Проблемы с платежными системами	Страхование (передача)	Интегрировать несколько платёжных провайдеров для обеспечения отказоустойчивости и системы оплаты.	Реализовать альтернативный метод оплаты (например, банковский перевод) в случае сбоя основных систем.
7	Уязвимости в безопасности	Минимизация	Провести аудит безопасности, внедрить актуальные стандарты защиты (шифрование, двухфакторная аутентификация) и регулярно проводить пентесты.	Организовать оперативное реагирование и исправление уязвимостей, привлечь сторонних экспертов по кибербезопасности.
8	Проблемы с интеграцией сторонних сервисов	Минимизация	Провести предварительное тестирование API интеграций, установить регулярный мониторинг и коммуникацию с поставщиками сервисов.	При возникновении и проблем временно использовать альтернативные сервисы или ограничить функциональность до устранения неполадок.
9	Конкурентное давление и изменение рынка	Принятие	Постоянно анализировать рынок, собирать обратную связь от пользователей и корректировать стратегию развития проекта.	Разработать маркетинговую кампанию и усилить продвижение продукта для удержания доли рынка.
10	Проблемы с масштабируем	Минимизация	Спроектировать архитектуру с	Заранее подготовить



	остью		возможностью горизонтального масштабирования, регулярно проводить нагрузочное тестирование.	план перехода на более мощные серверы и оптимизацию кода при резком росте нагрузки.
--	-------	--	---	---

## ПРАКТИЧЕСКАЯ РАБОТА №14

В современных IT-проектах обеспечение непрерывности работы системы является критически важным. Для этого разрабатываются Disaster Recovery (DR) и Disaster Recovery Plan (DRP), позволяющие минимизировать время простоя и финансовые потери при возникновении сбоев. Одновременно с этим производится анализ эксплуатационных рисков с использованием стандартизированной методологии CVSS (Common Vulnerability Scoring System).

### 14.1 Обзор Disaster Recovery и DRP

Disaster Recovery (DR) – совокупность мер для быстрого восстановления IT-инфраструктуры при возникновении критических сбоев (например, отказ серверного оборудования или сбой работы ключевых компонентов).

Disaster Recovery Plan (DRP) включает:

1. Формирование DR-команды с распределёнными уровнями доступа.
2. Проведение предварительной оценки внешних и внутренних рисков.
3. Определение критически важных бизнес-процессов и требуемых уровней доступности.
4. Заключение SLA (Service-Level Agreement) с поставщиками услуг.
5. Установление целевых показателей восстановления (RPO – Recovery Point Objective и RTO – Recovery Time Objective).
6. Планирование резервного копирования, дублирования инфраструктуры и регулярное тестирование.

Такая подготовка позволяет снизить влияние отказов, обеспечить оперативное восстановление системы и защитить бизнес-процессы.

### 14.2 Составление вектора CVSS

Методика CVSS позволяет количественно оценить угрозу, исходя из ряда параметров. При адаптации для оценки эксплуатационных рисков используется следующий набор параметров:

1. Attack Vector (AV)

Описывает, через какую среду может быть реализована угроза:

- N (Network): Угроза осуществляется через сеть.
- A (Adjacent): Используются смежные сети.
- L (Local): Требуется локальный доступ.
- P (Physical): Необходим физический доступ.

## 2. Attack Complexity (AC)

Отражает сложность проведения атаки/эксплуатации уязвимости:

- L (Low): Не требуются специфические условия.
- H (High): Требуется дополнительные или сложные условия.

## 3. Privileges Required (PR)

Определяет, какие привилегии необходимы для осуществления атаки:

- N (None): Привилегии не требуются.
- L (Low): Требуется ограниченные привилегии.
- H (High): Требуется высокие привилегии.

## 4. User Interaction (UI)

Указывает, требуется ли участие пользователя для успешной эксплуатации:

- N (None): Не требуется взаимодействие.
- R (Required): Требуется взаимодействие (например, подтверждение).

## 5. Scope (S)

Определяет, влияет ли воздействие уязвимости только на затронутый компонент или распространяется за его пределы.

- U (Unchanged): Влияние ограничено исходным компонентом.
- C (Changed): Воздействие распространяется на другие компоненты системы.

## 6. Impact метрики

Оценивают, насколько уязвимость влияет на:

- C: Конфиденциальность данных.
- I: Целостность информации.
- A: Доступность системы

После выбора значений для всех параметров они объединяются в строку, называемую вектором. На основе этого вектора рассчитывается базовый балл (от 0 до 10), который характеризует критичность риска.

### 14.3 Преобразование угроз в риски с использованием CVSS

Ниже приведена таблица с тремя выбранными технологическими угрозами, преобразованными в эксплуатационные риски. В таблице указаны описание (Угроза + Уязвимость + Актив), адаптированный CVSS вектор и рассчитанный базовый балл.

Таблица 14.1 – CVSS

№	Технологическая угроза (Угроза + Уязвимость + Актив)	Адаптированный CVSS вектор	Балл CVSS
1	<p><b>Неработоспособность серверного оборудования</b></p> <p><i>Угроза:</i> Отказ серверов из-за нарушения температурного режима.</p> <p><i>Уязвимость:</i> Отсутствие системы мониторинга и аварийного охлаждения.</p> <p><i>Актив:</i> Серверное оборудование ЦОДа.</p>	CVSS:4.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	9.8
2	<p><b>Ошибки в программном обеспечении при обновлении системы</b></p> <p><i>Угроза:</i> Внедрение новой версии с критическими ошибками.</p> <p><i>Уязвимость:</i> Недостаточно отлаженный процесс обновлений (отсутствие автоматизированного тестирования).</p> <p><i>Актив:</i> Программное обеспечение проекта.</p>	CVSS:4.0/AV:N/AC:L/PR:N/UI:N/S:U/C:M/I:M/A:M	7.5
3	<p><b>Сбой внешнего API интеграции с</b></p>	CVSS:4.0/AV:N/AC:	7.0

	<p><b>платёжными системами</b></p> <p><i>Угроза:</i> Нестабильность или отказ работы внешнего API платёжного провайдера.</p> <p><i>Уязвимость:</i> Отсутствие резервных каналов и автоматического переключения.</p> <p><i>Актив:</i> Модуль платежей в маркетплейсе.</p>	<p>M/PR:N/UI:N/S:U/C: M/I:M/A:M</p>	
--	--	---	--

#### 14.4 Методы обработки эксплуатационных рисков

Для каждого риска разработаны стратегии реагирования, которые включают основной (предполагаемый план) и резервный (отходной) варианты.

##### **Риск 1. Неработоспособность серверного оборудования**

Стратегия обработки: Снижение риска.

Основной план:

1. Внедрение системы мониторинга температуры и состояния серверного оборудования.
2. Организация аварийного охлаждения и георезервирования (размещение резервного оборудования в другой геолокации).
3. Регулярное проведение DR-учений для проверки готовности системы.

Отходной план:

Переключение на резервные серверы или использование облачных сервисов с заранее согласованным SLA.

##### **Риск 2. Ошибки в программном обеспечении при обновлении системы**

Стратегия обработки: Снижение риска.

Основной план:

1. Внедрение процессов CI/CD с автоматизированным регрессионным тестированием и поэтапным обновлением.

2. Проведение обязательных код-ревью и аудита обновлений.

Отходной план:

Организация механизма отката до предыдущей стабильной версии и оперативное исправление обнаруженных ошибок.

**Риск 3. Сбой внешнего API интеграции с платёжными системами**

Стратегия обработки: Передача риска.

Основной план:

1. Заключение SLA с платёжными провайдерами, гарантирующего необходимый уровень доступности.

Интеграция альтернативных платёжных решений (дублирование API), позволяющая автоматически переключаться при сбое основного сервиса.

Отходной план:

Разработка временного режима оплаты (например, через банковский перевод или электронные кошельки) до восстановления основного API.

Для каждого риска предложены конкретные меры обработки (снижение или передача риска) с детальным обоснованием. Такой подход обеспечивает прозрачную, стандартизированную оценку эксплуатационных рисков и позволяет своевременно принимать управленческие решения для повышения надежности IT-инфраструктуры проекта.

## ВЫВОД

Таким образом, в обязанности менеджера проекта входило выстраивать этапы работы, благодаря чему команда всегда видела текущие задачи, их статус (в очереди, в работе, выполнено) и дальнейшие шаги.

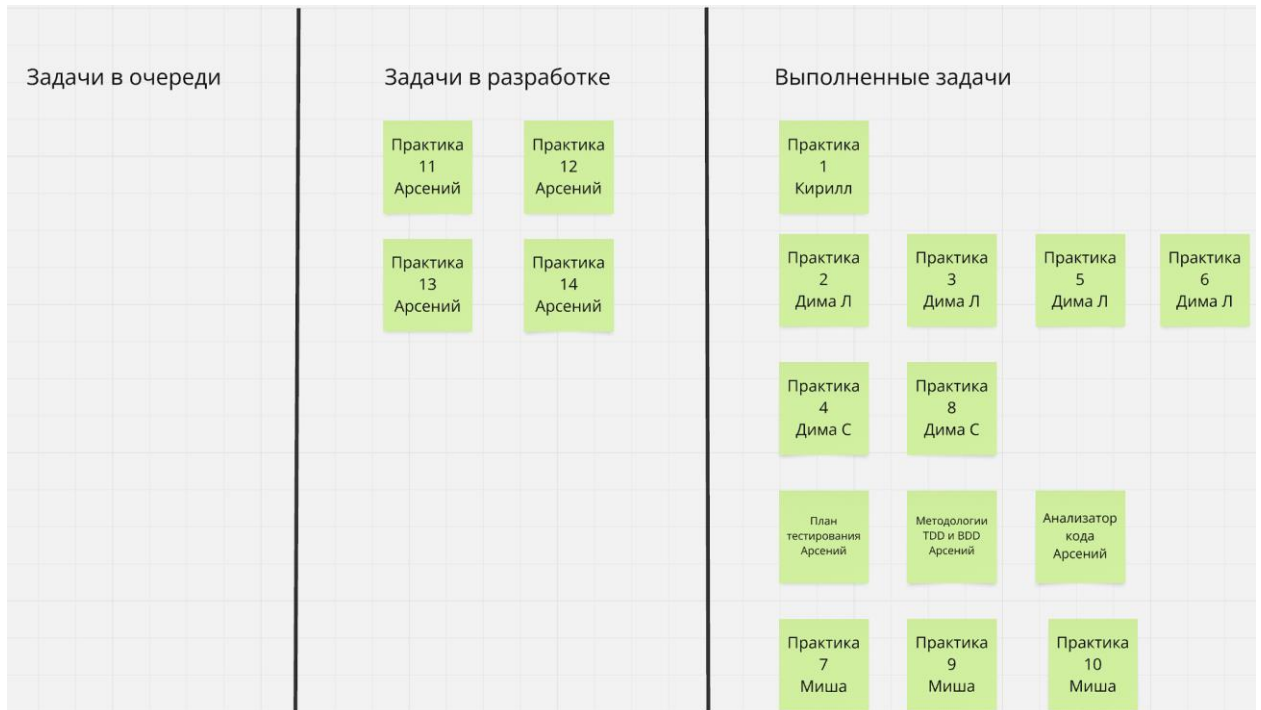


Рисунок 1.1 – Список задач

Аналитик смоделировал диаграммы информационных процессов и определил ключевые требования для пользователя.

Разработчик сделал полноценный сайт.

Тестировщик проверил работоспособность сайта.

Технический писатель составил отчёты о проделанных работах.

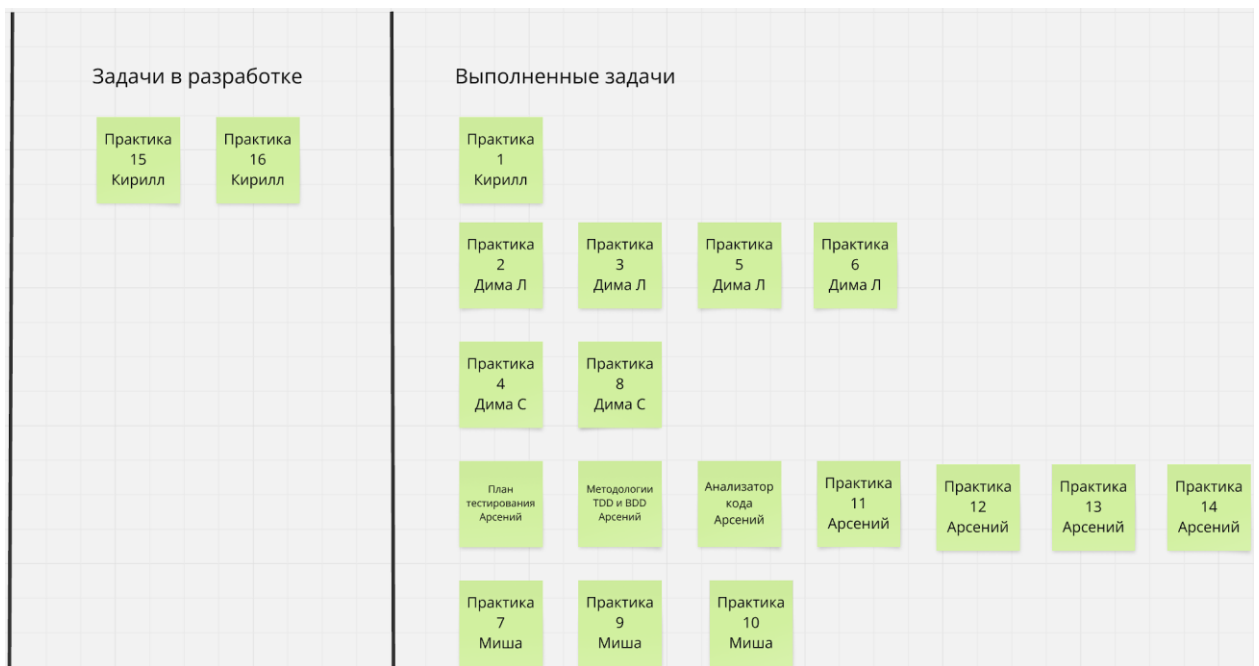
Также были проведены несколько онлайн-встреч.

Таблица 15.1 – Даты конференции

Встречи	Кто присутствовал	Дата	Вывод
Встреча №1	Руководитель, Тестировщик, Технический писатель	17.04.2025	<p>Обсуждены результаты тестирования функциональных и нефункциональных требований (раздел 11.3 отчета).</p> <p>Уточнены замечания по тест-кейсам и чек-листам (таблицы 11.1–11.11), требующие доработки.</p> <p>Определены задачи для анализа рисков (Практическая работа №13).</p> <p>Технический писатель скорректировал структуру отчета (раздел 14.2).</p>
Встреча №2	Руководитель, Разработчик, Тестировщик, Аналитик, Технический писатель	18.04.2025	<p>Проведен анализ эксплуатационных рисков (раздел 14.3) и доработан Disaster Recovery Plan (DRP) (раздел 14.1).</p> <p>Разработчик представил обновления по модулям поиска товаров и оформления заказов.</p> <p>Согласованы финальные версии матрицы требований (таблица 11.12) и плана реагирования на риски (таблица 13.2).</p>

После проведенных конференции список задач уменьшился и стал таким.





**Рисунок 1.2 – Список задач**

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гусев К. В., Воронцов Ю. А., Михайлова Е. К. Системная и программная инженерия: методические указания по выполнению практических работ. — Москва: РТУ МИРЭА, 2021. — 120 с.
2. Баранюк В. В. Системная и программная инженерия: методические указания по выполнению практических работ. Часть 1. — Москва: РТУ МИРЭА, 2020. — 110 с.
3. Лаврищева Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов. — Москва: Юрайт, 2021. — 350 с.
4. Лаврищева Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов. — Москва: Юрайт, 2021. — 330 с.
5. Черткова Е. А. Программная инженерия. Визуальное моделирование программных систем: учебник для вузов. — Москва: Юрайт, 2021. — 280 с.
6. Баранюк В. В., Миронов А. Н., Крылова О. С. Системная и программная инженерия: методические указания по выполнению практических работ. Ч. 1. — Москва: РТУ МИРЭА, 2020. — 130 с.
7. Дешко И. П., Кряженков К. Г., Цветков В. Я. Системная и программная инженерия: учебное пособие. — Москва: МАКС Пресс, 2018. — 250 с.