

Оглавление

ВВЕДЕНИЕ.....	8
1 Исследование предметной области.....	11
1.1 О предприятии.....	11
1.2 Требования к производимому тестированию	12
2 Проведение тестирования	14
2.1 Подготовительный этап.....	14
2.2 Проведение тестирования	15
2.3 Документирование багов.....	16
2.4 Код автоматизированного тестирования веб-приложения и API	17
ЗАКЛЮЧЕНИЕ	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	22

ВВЕДЕНИЕ

Тестирование и отладка программных продуктов являются ключевыми процессами, обеспечивающими высокую надежность, стабильность и соответствие программного обеспечения корпоративным стандартам и требованиям заказчика. В эпоху цифровой трансформации, когда ИТ-инфраструктуры становятся все более сложными, качество и безопасность программного обеспечения выходят на передний план, особенно в условиях растущих объемов данных и усложняющихся бизнес-процессов. Одним из таких критически важных внутренних приложений в современной компании является «Инструмент мониторинга и оповещений». Это программное обеспечение предназначено для автоматического отслеживания состояния различных систем компании, выявления багов и оперативного уведомления ответственных сотрудников о потенциальных проблемах.

Основной целью прохождения практики в компании «РУСТЕЛ ИННОВАЦИИ» было всестороннее изучение и освоение процессов тестирования и отладки программного обеспечения, а именно внутреннего приложения компании — «Инструмент мониторинга и оповещений». В рамках практики предстояло ознакомиться с архитектурой, принципами функционирования и технологическими решениями, заложенными в основу данного приложения. Помимо этого, необходимо было провести комплексное тестирование всех его функциональных модулей и компонентов, а также предложить меры по повышению качества работы данного программного обеспечения.

Процесс тестирования включал несколько этапов, начиная с детального анализа технической документации и требований, предъявляемых к продукту, разработки детализированных тестовых сценариев, охватывающих ключевые аспекты функционирования системы, и заканчивая проведением функционального и ручного тестирования. Эти этапы сопровождались анализом результатов и оформлением итоговых отчетов с рекомендациями для

дальнейшего совершенствования системы. Особое внимание в ходе практики уделялось процессу отладки программного обеспечения, который заключался в обнаружении и устранении ошибок, выявленных в процессе тестирования, а также в улучшении общего пользовательского интерфейса и опыта взаимодействия с приложением.

Одной из центральных задач практики стало глубокое изучение методов автоматизированного и ручного тестирования, что позволило провести всестороннюю проверку стабильности и производительности работы «Инструмента мониторинга и оповещений» в различных эксплуатационных условиях. Для автоматизированного тестирования использовались такие инструменты, как Selenium для тестирования веб-интерфейсов. Параллельно было выполнено ручное тестирование пользовательских сценариев для выявления скрытых багов и ошибок в логике работы системы. В процессе практики также возникла необходимость тестирования программного обеспечения на соответствие требованиям безопасности, включая защиту от SQL-инъекций, XSS-атак и других уязвимостей, что стало важным аспектом в обеспечении безопасности корпоративных данных.

Таким образом, основными задачами практики стали:

1. ознакомление с архитектурой и функциональными возможностями внутреннего приложения «инструмент мониторинга и оповещений», включая его взаимодействие с внешними системами и сервисами,
2. разработка и выполнение тестовых сценариев для комплексной проверки функциональности и производительности приложения,
3. проведение автоматизированного и ручного тестирования с использованием современных инструментов и методик,
4. анализ и документирование найденных дефектов, их последующее устранение в процессе отладки и доработки программного обеспечения, с целью улучшения как функциональной составляющей,

5. исследование аспектов безопасности приложения и проведение тестирования на соответствие стандартам информационной безопасности,

6. формирование предложений по улучшению качества и производительности приложения на основе проведенного тестирования и анализа.

Содержание и оформление отчета выполнялось в соответствии с требованиями ГОСТ 7.32.2017 [1].

1 Исследование предметной области

1.1 О предприятии

Предприятием, на котором проходила производственная практика, является общество с ограниченной ответственностью «РУСТЕЛ ИННОВАЦИИ» — развивающаяся компания, специализирующаяся на разработке и внедрении инновационных решений в области информационных технологий. Основной акцент в деятельности ООО «РУСТЕЛ ИННОВАЦИИ» делается на создании высокотехнологичных программных продуктов для корпоративного сектора, таких как системы мониторинга, оповещения и комплексных решений для автоматизации бизнес-процессов, которые позволяют компаниям эффективно управлять своими IT-инфраструктурами и оперативно реагировать на любые сбои или критические ситуации.

Компания структурирована таким образом, чтобы каждая команда или подразделение было сфокусировано на конкретных задачах — от начального этапа проектирования и разработки программного обеспечения до его финального тестирования, отладки и обеспечения высокого качества выпускаемого продукта. Такой подход позволяет гибко адаптироваться под потребности клиентов и быстро реагировать на изменения в рыночных тенденциях. Высококвалифицированные специалисты компании занимаются созданием как индивидуальных решений, разработанных под конкретные нужды заказчиков, так и универсальных программных продуктов, ориентированных на широкий круг корпоративных пользователей. Среди ключевых внутренних продуктов компании выделяется «инструмент мониторинга и оповещений», которые обеспечивают автоматическое отслеживание состояния различных информационных систем и оперативное уведомление пользователей о возможных проблемах.

Одной из ключевых задач, стоящих перед ООО «РУСТЕЛ ИННОВАЦИИ», является обеспечение высокого уровня качества и надежности программного обеспечения, разработанного как для внутренних нужд компании,

так и для ее внешних клиентов. В целях достижения этой цели компания активно внедряет современные методики тестирования, отладки и автоматизации процессов разработки программного обеспечения. Это не только ускоряет процесс вывода новых продуктов на рынок, но и гарантирует их стабильную и безопасную эксплуатацию, особенно в условиях постоянно усложняющейся IT-инфраструктуры.

Таким образом, ООО «РУСТЕЛ ИННОВАЦИИ» создает современное программное обеспечение, ориентированное на потребности бизнеса, и предоставляет своим сотрудникам, включая практикантов, возможность работы с передовыми технологиями и решениями. Практиканты получают уникальный опыт, участвуя в реальных проектах, что позволяет им приобрести ценные навыки и знания, которые могут стать основой для дальнейшего профессионального развития в сфере IT.

1.2 Требования к производимому тестированию

Для оптимизации и отладки программного обеспечения руководитель практики поставил перед тестами требования:

- тестирование программного средства в высоконагруженных условиях,
- тестирование программного средства методом черного ящика,
- минимизация числа действий, требуемых от пользователя,
- возможность экспорта результатов в MS Excel,
- поддержка Ubuntu Server, Arch Linux,
- обеспечение авторизованного доступа к информации,
- возможность масштабирования программного средства.

После исследования выдвинутых требований был проведен бриффинг, в результате которого было принято решение разработать собственное ПО по тестированию. Для данного ПО были выдвинуты следующие требования:

- ограничения для вводимых данных,
- функции, которые должно выполнять программное средство.

2 Проведение тестирования

Основной акцент в рамках практики был сделан на проведении всестороннего тестирования внутреннего приложения компании — «Инструмента мониторинга и оповещений». Одной из главных задач, поставленных передо мной, было тестирование новой функциональности, которая касалась обработки данных и генерации уведомлений. Данный этап работы включал в себя множество шагов, начиная с детальной проработки плана тестирования и создания тестовых сценариев, до выполнения непосредственного тестирования, используя как ручные, так и автоматизированные подходы. Важной составляющей данного процесса также было тщательное документирование всех обнаруженных багов и проблем, возникавших в ходе тестирования.

2.1 Подготовительный этап

Первоначальный этап тестирования начался с разработки детализированного плана, который должен был обеспечить систематизированный и последовательный подход к проверке функциональности приложения. План тестирования был создан с целью охвата всех ключевых аспектов работы приложения и обеспечения комплексного анализа его функциональности. Основными элементами плана стали следующие:

- определение ключевых целей тестирования, включая проверку корректности обработки данных, своевременность и точность генерации оповещений, а также оценку устойчивости системы при различных нагрузках;
- составление списка тестовых сценариев, которые охватывали все основные функциональные элементы новой версии "Инструмента мониторинга и оповещений", каждый сценарий был ранжирован по приоритету, исходя из критичности проверяемой функции для работы системы в целом;
- выбор ручных и автоматизированных методов тестирования: ручное тестирование было направлено на проверку пользовательского интерфейса и

общей логики работы приложения; тогда как автоматизированное тестирование обеспечивало проверку стабильности и производительности системы;

- определение ожидаемых результатов и критерий, по которым можно судить о его успешности: корректность данных, отображаемых в интерфейсе, своевременность и точность оповещений, а также отсутствие ошибок при нагрузочном тестировании.

На основе вышеупомянутого плана были разработаны детализированные тестовые сценарии. Эти сценарии включали четкие шаги, необходимые для проверки каждой функции приложения, с указанием ожидаемых результатов на каждом этапе.

2.2 Проведение тестирования

После подготовительного этапа было проведено тестирования, которое включало как ручные, так и автоматизированные методы:

- ручное тестирование – в рамках этого этапа я проверила основные аспекты пользовательского интерфейса, такие как навигация, отображение данных и корректность генерации оповещений. Логика работы приложения была тщательно проверена для того, чтобы убедиться в её соответствии предъявленным требованиям и пользовательским сценариям. Важно было оценить, насколько удобно пользователи могут взаимодействовать с системой, а также насколько оперативно и точно приложение генерирует уведомления при возникновении потенциальных проблем. Каждый тестовый сценарий был выполнен в полном объеме, и результаты тестирования были подробно зафиксированы в системе управления задачами JIRA. Это не только помогло наглядно продемонстрировать успешные этапы тестирования, но и предоставило возможность точно отслеживать все выявленные ошибки и проблемы.

- автоматизированное тестирование – для автоматизации тестирования веб-интерфейсов я использовала инструмент Selenium [5]. Это позволило мне провести нагрузочные тесты, оценивая стабильность системы при

работе в различных браузерах и при различных уровнях нагрузки. Я написала несколько автоматизированных тестов, которые проверяли ключевые функциональности приложения, такие как скорость обработки данных и корректность отображения информации.

– тестирование API – в дополнение к тестированию интерфейса я также провела тестирование API с использованием инструмента Postman. Это позволило мне проверить корректность обработки запросов и ответов, убедиться в правильной работе API при взаимодействии с внешними системами. Тестирование включало проверку того, насколько правильно API обрабатывает запросы различных типов, поддерживает ли оно корректные ответы в требуемом формате, а также устойчиво ли к некорректным запросам или ошибочным данным. Особое внимание было уделено интеграции с такими инструментами мониторинга, как Grafana и Prometheus, что позволило оценить, как API взаимодействует с этими системами и обеспечивает сбор данных для их последующего анализа. Проверка включала анализ производительности API, а также его устойчивости к некорректным запросам и высоким нагрузкам.

2.3 Документирование багов

В ходе тестирования было выявлено несколько критических ошибок, которые могли существенно повлиять на работу системы. Например, были обнаружены проблемы с некорректной обработкой данных при высоких нагрузках, что могло привести к сбоям в генерации оповещений. Помимо этого, я выявила несколько мелких багов, связанных с отображением информации в пользовательском интерфейсе.

Все обнаруженные баги были тщательно задокументированы в JIRA. Каждая запись содержала подробное описание проблемы, шаги для ее воспроизведения, а также прикрепленные скриншоты и логи, подтверждающие наличие ошибки. Это позволило команде разработчиков быстро приступить к исправлению багов, а также обеспечить прозрачность процесса тестирования для всех участников проекта.

В итоге, проведенное тестирование помогло выявить и устранить критические ошибки, повысив надежность и стабильность работы «Инструмента мониторинга и оповещений».

2.4 Код автоматизированного тестирования веб-приложения и API

Листинг 1 – Тест на добавление нового сервера в мониторинг

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

# WebDriver setup
driver = webdriver.Chrome()

# Открытие веб-приложения
driver.get("http://localhost:8080")

# Логин
driver.find_element(By.ID, "username").send_keys("admin")
driver.find_element(By.ID, "password").send_keys("admin_pass")
driver.find_element(By.ID, "login-btn").click()

# Проверка успешного входа
assert "Dashboard" in driver.title

# Переход на страницу управления серверами
driver.find_element(By.LINK_TEXT, "Servers").click()

# Добавление нового сервера
driver.find_element(By.ID, "add-server-btn").click()
driver.find_element(By.ID, "server-name").send_keys("New Server")
driver.find_element(By.ID, "server-ip").send_keys("192.168.1.10")
driver.find_element(By.ID, "server-type").send_keys("Linux")
driver.find_element(By.ID, "save-server-btn").click()

# Ожидание и проверка успешного добавления сервера
time.sleep(2)
assert "Server added successfully" in driver.page_source

# Logout
driver.find_element(By.ID, "logout-btn").click()
time.sleep(2)
assert "Login" in driver.title

# Закрытие браузера
driver.quit()
```

Листинг 2 – Автоматизированный тест на проверку скорости обработки данных

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

# Установка WebDriver
driver = webdriver.Chrome()

# Открытие веб-приложения
driver.get("http://localhost:8080")

# Начало отсчета времени
start_time = time.time()

# отправка формы
driver.find_element(By.ID, "cpu-load").send_keys("85") # Процент
загрузки CPU
driver.find_element(By.ID, "memory-usage").send_keys("16 GB") #
Использование оперативной памяти
driver.find_element(By.ID, "disk-space").send_keys("500 MB") #
Свободное место на диске
driver.find_element(By.ID, "submit-
btn").click()

# Ожидание загрузки данных
time.sleep(2)

# Конец отсчета времени
end_time = time.time()

# Проверка, что данные отобразились корректно
assert "Результат успешно обработан" in driver.page_source

# Вывод времени обработки данных
processing_time = end_time - start_time
print(f"Время обработки данных: {processing_time} секунд")

# Закрытие браузера
driver.quit()
```

Листинг 3 – Тестирование API для получения списка серверов

```
import requests

base_url = "http://localhost:8080/api"
headers = {'Content-Type': 'application/json'}

# Тест GET-запроса на получение списка серверов
response = requests.get(f"{base_url}/servers", headers=headers)
assert response.status_code == 200
servers = response.json()
assert isinstance(servers, list)
```

```
# Проверка, что список серверов не пуст
assert len(servers) > 0
print("Список серверов получен успешно.")
```

ЗАКЛЮЧЕНИЕ

Практика в компании предоставила уникальную возможность применить знания и навыки, полученные в университете, в реальных условиях. Работая над тестированием и отладкой внутреннего приложения "Инструмент мониторинга и оповещений", я значительно расширила свои профессиональные горизонты и приобрела ценный опыт, который окажется полезным в моей дальнейшей карьере.

В ходе практики я освоила современные инструменты тестирования, такие как Selenium и Postman. Этот опыт позволил понять важность грамотной организации тестирования и планирования работы над проектом.

Работа с функциональными и нефункциональными требованиями помогла глубже понять ключевые аспекты тестирования, включая важность выявления ошибок на ранних этапах разработки и своевременного их устранения. Особенно ценной была возможность освоить автоматизированное тестирование, которое ускоряет процессы проверки и обеспечивает стабильность системы при различных сценариях использования.

Основные выводы по итогам практики:

- Тестирование является неотъемлемой частью процесса разработки программного обеспечения, определяющей его качество, стабильность и надёжность.
- Автоматизация тестирования играет ключевую роль в сокращении времени на проверку и позволяет более эффективно выявлять ошибки, что особенно важно при работе с крупными и сложными проектами.
- Эффективная коммуникация между членами команды — тестировщиками, разработчиками, аналитиками — является важнейшим фактором для успешного завершения проектов и оперативного устранения обнаруженных дефектов.

– Практика позволила не только укрепить профессиональные знания и навыки, но и получить более глубокое представление о процессе разработки программного обеспечения в ИТ-индустрии.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32.2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.
2. Python Documentation. [Электронный ресурс]. URL: <https://docs.python.org/3/index.html> (дата обращения: 20.07.2024).
3. Ubuntu Server Documentation. [Электронный ресурс]. URL: <https://ubuntu.com/server/docs> (дата обращения: 17.07.2024).
4. Clarke J. SQL injection attacks and defense – 2012 г.
5. David Bruns. Selenium 2 Testing Tools: Beginner's Guide: учеб. пособие – Birmingham B3 2PB, UK. – 232 стр. – ISBN 978-1-84951-830-7