

Министерство науки и высшего образования и РФ
федеральное государственное автономное образовательное учреждение высшего образования
«Омский государственный технический университет»

Факультет информационных технологий и компьютерных систем

Кафедра «Информатика и вычислительная техника»

Дисциплина «Проектная деятельность»

ЗАДАНИЕ

на выполнение курсового проекта

Студенту (ке) Кулагину Александру Васильевичу Группа ИВТ-213

фамилия, имя, отчество полностью

Направление (специальность) 09.03.01 – Информатика и вычислительная техника

код, наименование

Тема проекта Роль инструментов автоматизированного тестирования в разработке приложений

Срок сдачи проекта (работы) на кафедру « » 20 г.

Исходные данные к проекту (работе)

Содержание пояснительной записки (перечень подлежащих разработке вопросов)

Введение. 1. Описание предметной области. 2. Проектирование алгоритмов для решения задач 2.1 Схема данных 2.2 Схема ресурсов системы 2.3 Схема работы системы 2.4 Схема алгоритма 3. Разработка сценариев использования системы (use case) 4. Общие понятия тестирования и его виды 5. Автоматизированное тестирование 6. Популярные инструменты автоматизированного тестирования

Перечень графического материала с указанием основных чертежей и (или) иллюстративного материала Схема алгоритма, схема ресурсов, схема работы системы, схема данных

Методическая литература и иные информационные источники

1. Петренко А. В., Махненко А. С. Основы тестирования программного обеспечения. — Москва: ДМК Пресс, 2019. — 240 с.
2. Боев Н.В. Проектирование информационных систем. — Москва: Издательский дом «Питер», 2019. — 384 с.
3. Иванов А. П. Автоматизированное тестирование веб-приложений // Программные продукты и системы. — 2020. — № 3. — С. 45-52.
4. Ковалевский А. Введение в автоматизированное тестирование [Электронный ресурс]. — URL: <https://www.testing.ru/articles/automation-testing> (дата обращения: 18.03.2025).

Дата выдачи задания « » 20 г.

Руководитель ст. пр. Блохин А.В.

подпись

ученая степень, звание, ФИО

дата

Зав. кафедрой к.т.н., доцент Грицай А.С.

подпись

ученая степень, звание, ФИО

дата

Задание принял к

исполнению студент (ка) « » 20 г.

подпись

Министерство образования и науки РФ
федеральное государственное автономное образовательное учреждение высшего образования
«Омский государственный технический университет»

Факультет (институт) Информационных технологий и компьютерных систем
Кафедра Информатики и вычислительной техники

КУРСОВОЙ ПРОЕКТ

по дисциплине Проектная деятельность

на тему Роль инструментов автоматизированного тестирования в
разработке приложений

Пояснительная записка

Шифр проекта 020-КП-09.03.01-№29 ПЗ

Студента (ки) Кулагина Александра Васильевича
фамилия, имя, отчество полностью

Курс 4 Группа ИВТ-213

Направление (специальность) 09.03.01 –
Информатика и вычислительная техника
код, наименование

Руководитель Старший преподаватель
ученая степень, звание
Блохин А.В.
фамилия, инициалы

Выполнил (а) _____
дата, подпись студента (ки)

К защите _____
дата, подпись руководителя

| Выполнение и подготовка к защите КП (КР) | Защита КП (КР) | Итоговый рейтинг |
|--|----------------|---------------------|
| | | |

Проект (работа) защищен (а) с оценкой _____

Омск 2025

Министерство науки и высшего образования и РФ
федеральное государственное автономное образовательное учреждение высшего образования
«Омский государственный технический университет»

ОТЗЫВ
на курсовой проект

Факультет _____ *информационных технологий и компьютерных систем* _____
Кафедра _____ *Информатика и вычислительная техника* _____
Дисциплина _____ *Проектная деятельность* _____
Тема _____ *Роль инструментов автоматизированного тестирования* _____
_____ *в разработке приложений* _____
Студент _____ *Кулагин Александр Васильевич* _____
_____ *фамилия, имя, отчество полностью* _____
Курс _____ *4* _____ Группа _____ *ИВТ-213* _____
Руководитель _____ *ст. пр. Блохин А.В.* _____
_____ *ученая степень, звание, ФИО* _____

Содержание отзыва

Были спроектированы: схема данных системы, схема ресурсов системы, схема работы системы и схема алгоритма. Разработаны ключевые сценарии использования системы. Написан реферат на тему «Роль инструментов автоматизированного тестирования в разработке приложений». Оформлена пояснительная записка.

| | |
|---|--|
| Рейтинговые баллы за выполнение и подготовку к защите курсового проекта | |
| Заключение о допуске к защите | |

Руководитель _____ Дата _____ 20 _____ г.
_____ подпись

Реферат

Пояснительная записка 23 с., 4 рис.

**АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ, ВЕБ-ТЕХНОЛОГИИ,
ИНСТРУМЕНТЫ ТЕСТИРОВАНИЯ, КАЧЕСТВО ПО.**

Цель работы – изучить роль инструментов автоматизированного тестирования в разработке приложений; спроектировать схему данных, схему ресурсов, схему работы системы и схему алгоритма веб-платформы для предприятий малого и среднего бизнеса.

В ходе курсового проекта была изучена роль инструментов автоматизированного тестирования в разработке приложений, а также спроектирована схема данных подтверждения заявки пользователя, схема ресурсов системы изменения филиала, схема работы системы (изменение статуса заказа на выполненный), схема алгоритма оформления заказа веб-платформы для предприятий малого и среднего бизнеса.

Содержание

| | |
|---|----|
| Введение | 6 |
| 1. Описание предметной области | 7 |
| 2. Проектирование алгоритмов для решения задач | 8 |
| 2.1 Схема данных | 8 |
| 2.2 Схема ресурсов системы | 8 |
| 2.3 Схема работы системы | 9 |
| 2.4 Схема алгоритма | 10 |
| 3. Разработка сценариев использования системы (use case)..... | 11 |
| 4. Общие понятия тестирования и его виды | 13 |
| 5. Автоматизированное тестирование | 15 |
| 6. Популярные инструменты автоматизированного тестирования..... | 18 |
| 6.1. Инструменты для юнит-тестирования | 18 |
| 6.2. Инструменты для функционального тестирования..... | 19 |
| 6.3. Инструменты для нагрузочного тестирования | 19 |
| 6.4. Инструменты для тестирования мобильных приложений..... | 20 |
| Заключение | 22 |
| Список использованных источников | 23 |

Введение

В современном мире разработка программного обеспечения становится все более сложным и многоэтапным процессом, где ключевым фактором успеха является обеспечение высокого качества продукта. С ростом сложности приложений и увеличением объемов кода традиционные методы тестирования перестают справляться с задачами, требующими быстрого и точного анализа функциональности системы. Именно поэтому автоматизированное тестирование занимает центральное место в процессе разработки приложений, позволяя минимизировать ошибки, сократить время на выявление дефектов и повысить общую надежность программного обеспечения.

Разработка веб-платформы для предприятий малого и среднего бизнеса является актуальной задачей, поскольку разработанная веб-платформа позволит управлять всеми филиалами в одном месте, отслеживая их показатели и планируя расширение ассортимента предоставляемых товаров и услуг потребителям.

В ходе работы необходимо рассмотреть основные понятия и принципы автоматизированного тестирования, проанализировать популярные инструменты для автоматизации тестирования, выявить их преимущества и недостатки, а также разработать проектную документацию для веб-платформы, учитывающую требования к качеству и эффективности разработки.

1. Описание предметной области

Разработка программного обеспечения (ПО) — это сложный и многоэтапный процесс, который требует высокой точности, надежности и эффективности. Одним из ключевых аспектов успешной разработки является тестирование, которое направлено на выявление ошибок, проверку корректности работы приложения и обеспечение его соответствия требованиям пользователей. В современной разработке ПО тестирование может быть как ручным, так и автоматизированным. Автоматизированное тестирование становится особенно важным в условиях жестких сроков, высокой сложности приложений и необходимости частых обновлений.

Инструменты автоматизированного тестирования представляют собой программные средства, которые позволяют автоматизировать выполнение тестовых сценариев, анализ результатов и генерацию отчетов. Эти инструменты играют критически важную роль в повышении качества программного обеспечения, снижении затрат времени и ресурсов, а также минимизации человеческого фактора.

Малый и средний бизнес является важнейшим сегментом экономики, обеспечивающим занятость населения и развитие локальных рынков. Для успешного функционирования предприятия требуются эффективные инструменты управления, автоматизации процессов и взаимодействия с клиентами. Web-платформы представляют собой универсальное решение, которое позволяет владельцам бизнеса и их сотрудникам упростить выполнение повседневных задач, оптимизировать ресурсы и повысить конкурентоспособность.

Web-платформа — это комплексное программное обеспечение, доступное через браузер, которое предоставляет инструменты для управления различными аспектами деятельности компании: от учета товаров и услуг до анализа данных и взаимодействия с клиентами. Такие платформы адаптируются под потребности конкретного бизнеса и могут быть масштабируемыми в зависимости от его роста.

2. Проектирование алгоритмов для решения задач

2.1 Схема данных

Была разработана схема данных. Схемы данных отображают путь данных при решении задач и определяют этапы обработки, а также различные применяемые носители данных. Схема данных для подтверждения заявки от пользователя, представлена на рисунке 1.

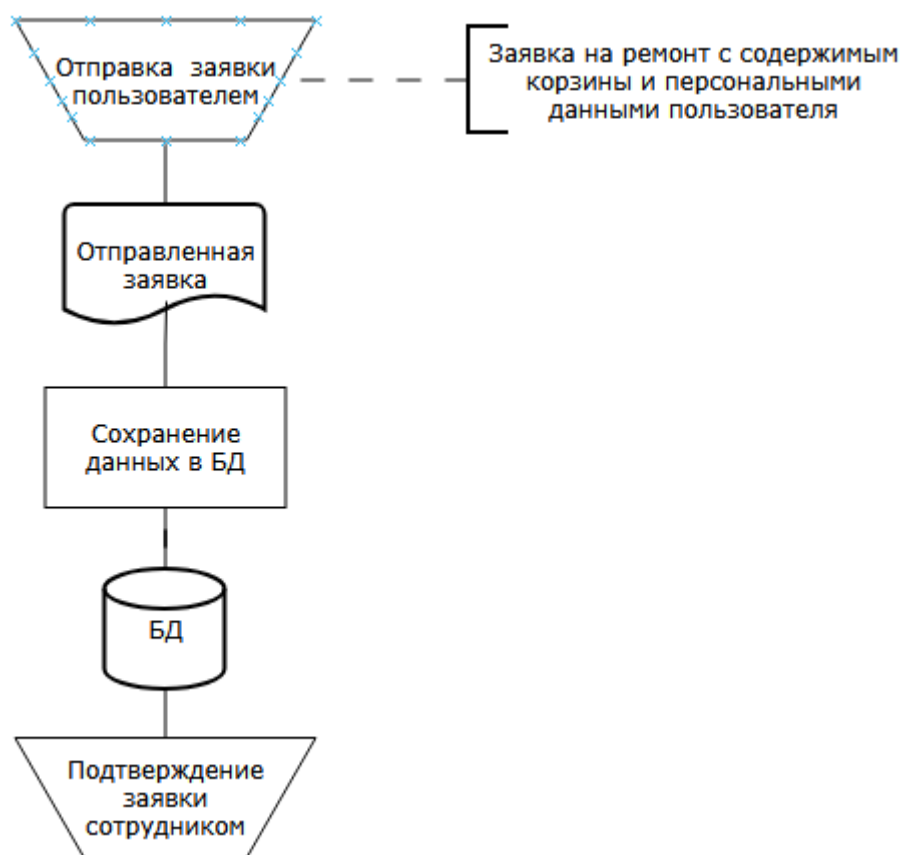


Рисунок 1 – Схема данных (подтверждение заявки пользователя)

2.2 Схема ресурсов системы

Схемы ресурсов системы отображают конфигурацию блоков данных и обрабатывающих блоков, которая требуется для решения задачи или набора задач. Схема ресурсов приложения изображена на рисунке 2.

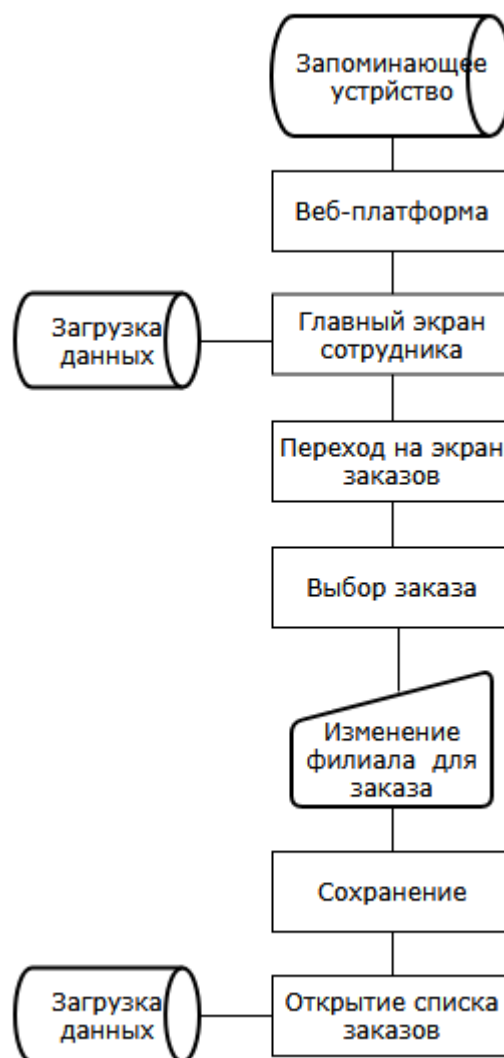


Рисунок 2 – Схема ресурсов системы (изменение филиала)

2.3 Схема работы системы

Схемы работы системы отображают управление операциями и поток данных в системе.

Схема работы системы состоит из:

- 1) символов данных, указывающих на наличие данных;
- 2) символов процесса, указывающих операции, которые следует выполнить над данными, а также определяющих логический путь, которого следует придерживаться;
- 3) линейных символов, указывающих потоки данных между процессами и (или) носителями данных, а также поток управления между процессами;
- 4) специальных символов, используемых для облегчения написания и чтения

блок-схемы.

Схема работы системы представлена на рисунке 3.

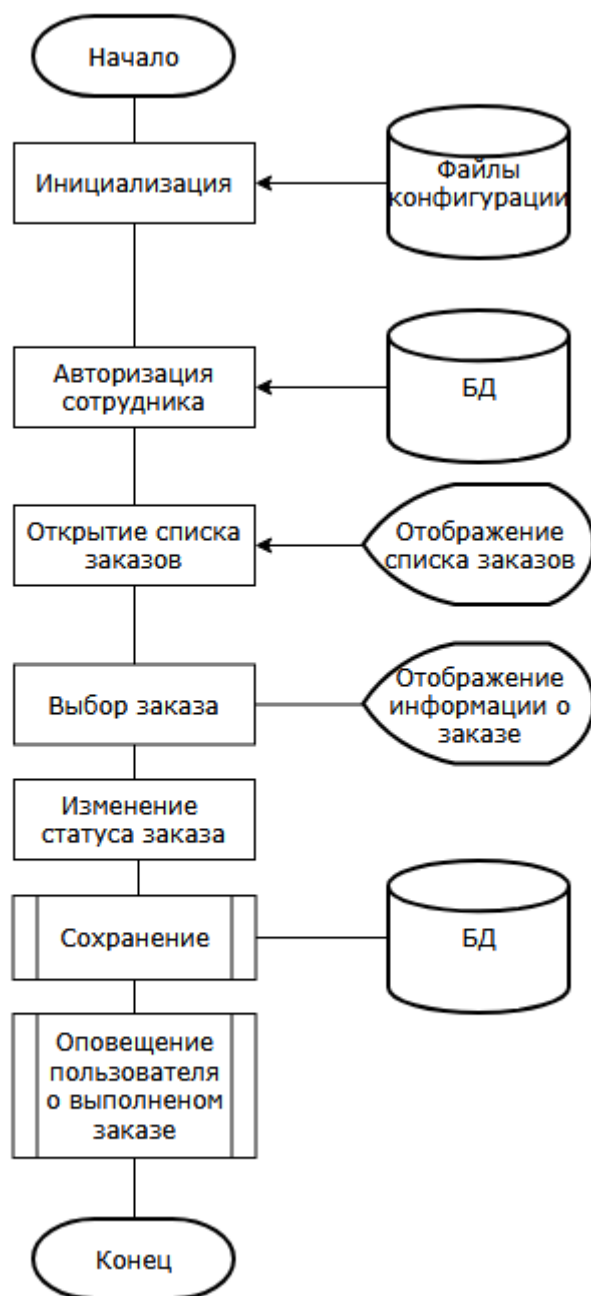


Рисунок 3 – Схема работы системы (изменение статуса заказа на выполненный)

2.4 Схема алгоритма

Схема алгоритма оформления заказа клиентом представлена на рисунке 4.

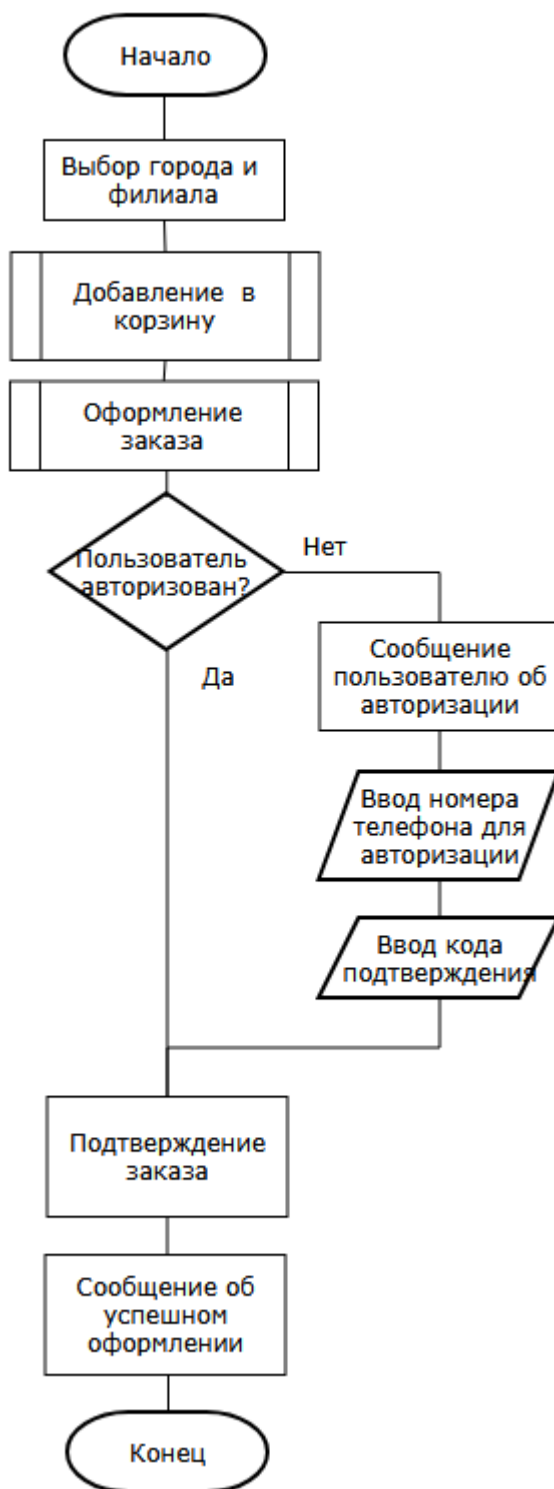


Рисунок 4 – Схема алгоритма (оформление заказа)

3. Разработка сценариев использования системы (use case)

В данном разделе представлена разработка ключевых сценариев использования системы (use case). Рассмотрены два сценария: добавление товара в корзину и подтверждение заявки сотрудником. Каждый описывает последовательность действий пользователя и системы, необходимых для

достижения конечной цели.

Сценарий добавление товара в корзину представленный в таблице 1 является одним из основных сценариев взаимодействия клиента с сайтом.

Таблица 1 – Сценарий использования «Добавление товара в корзину»

| | |
|---------------------------|---|
| Вариант использования | Добавление товара в корзину |
| Область действия | Клиентская часть веб-платформы |
| Уровень | Цели пользователя |
| Основное действующее лицо | Пользователь сайта |
| Участники | Пользователь сайта, Система |
| Предусловие | Пользователь открыл веб-платформу и перешел в раздел услуг |
| Результат | Товар добавлен в корзину. |
| Триггер | Пользователь хочет записаться на услугу |
| Описание | 1. Пользователь открывает главную страницу. 2. Пользователь выбирает пункт меню «Услуги». 3. Пользователь выбирает интересующую его услугу из доступных. 4. Пользователь нажимает кнопку добавления в корзину. 5. Товар добавлен в корзину. |
| Расширение | 4.1 Если комплектующих для оказания услуги нет на сайте, кнопка добавления в корзину неактивна и отображается надпись «Нет в наличии» |

Сценарий подтверждения заявки сотрудником, представленный в таблице 2 представляет собой один из основных сценариев взаимодействия сотрудника с сайтом.

Таблица 2 – Сценарий использования «Подтверждение заявки сотрудником»

| | |
|---------------------------|--|
| Вариант использования | Подтверждение заявки сотрудником |
| Область действия | Часть веб-платформы для сотрудников |
| Уровень | Цели пользователя (сотрудника) |
| Основное действующее лицо | Сотрудник |
| Участники | Сотрудник, Система |
| Предусловие | Сотрудник открыл веб-платформу и перешел в раздел заявок |
| Результат | Заявка подтверждена |
| Триггер | Клиент оформил заявку на услугу |
| Описание | 1. Сотрудник открывает главную страницу портала 2. Сотрудник выбирает пункт меню «Заказы». 3. Сотрудник переходит в раздел «Новые заявки» 4. Сотрудник подтверждает новую заявку. |

| | |
|------------|---|
| | 5. Заявка подтверждена. |
| Расширение | 1.1 Если сотрудник не авторизован будет предложено авторизоваться для доступа к portalу 4.1 Если новых заявок нет, то будет сообщение «Нет новых заявок» |

Представленные сценарии использования образуют базовый функционал для пользовательского опыта в системе, который обеспечивает эффективность ее применения.

4. Общие понятия тестирования и его виды

Тестирование программного обеспечения — это процесс проверки соответствия разработанного продукта предъявляемым требованиям, выявления дефектов и недостатков, а также оценки его работоспособности и удобства использования. Тестирование направлено на обеспечение качества продукта, минимизацию рисков и повышение уверенности в том, что система будет выполнять свои функции корректно.

Цели тестирования:

- **Выявление ошибок:** Основная цель тестирования заключается в обнаружении багов, дефектов или несоответствий в работе программы до её выпуска.
- **Проверка соответствия требованиям:** Убедиться, что приложение выполняет все функции, описанные в техническом задании (ТЗ) или спецификациях.
- **Обеспечение качества:** Гарантировать, что конечный продукт соответствует ожиданиям пользователей и отраслевым стандартам.
- **Подтверждение стабильности системы:** Убедиться, что приложение работает корректно в различных условиях, включая нагрузку, стрессовые ситуации и взаимодействие с другими системами.
- **Снижение затрат на поддержку:** Раннее выявление ошибок позволяет снизить затраты на их исправление после выпуска продукта.

В зависимости от целей и методов выполнения тестирование можно разделить на несколько основных типов:

1. Функциональное тестирование

- Проверка корректности работы функций приложения согласно требованиям.

- Пример: Проверка, что кнопка "Отправить" в форме действительно отправляет данные на сервер.

2. Нагрузочное (производительностное) тестирование

- Оценка производительности системы под различными нагрузками (количество пользователей, объем данных).

- Пример: Проверка, как сайт справляется с 1000 одновременными посетителями.

3. Регрессионное тестирование

- Проверка, что изменения в коде (например, добавление новой функции) не нарушили работу уже существующих компонентов.

- Пример: После обновления мобильного приложения убедиться, что старые функции работают так же, как и раньше.

4. Юнит-тестирование

- Проверка отдельных модулей или компонентов кода на предмет их корректной работы.

- Пример: Тестирование функции сложения двух чисел в программе.

5. Интеграционное тестирование

- Проверка взаимодействия между различными модулями или компонентами системы.

- Пример: Убедиться, что база данных корректно передает данные в пользовательский интерфейс.

6. Тестирование пользовательского интерфейса (UI)

- Проверка удобства и корректности работы интерфейса для конечного пользователя.

– Пример: Проверка, что все кнопки и поля ввода на сайте доступны и работают правильно.

7. Безопасностное тестирование

– Проверка устойчивости системы к угрозам безопасности (взломы, утечки данных, несанкционированный доступ).

– Пример: Проверка, что пароль пользователя нельзя восстановить через базу данных.

5. Автоматизированное тестирование

Автоматизированное тестирование — это процесс выполнения тестов с использованием специализированных инструментов, скриптов или программных решений для проверки работоспособности, производительности и соответствия требованиям программного обеспечения. В отличие от ручного тестирования, где тестировщик выполняет действия вручную, автоматизация позволяет выполнять тесты без прямого участия человека.

Отличия от ручного тестирования:

1. **Скорость выполнения:** Автоматизированные тесты выполняются значительно быстрее, так как они не зависят от скорости работы человека.

2. **Повторяемость:** Автоматизированные тесты можно запускать многократно с одинаковыми входными данными, что исключает человеческий фактор.

3. **Объем покрытия:** Автоматизация позволяет охватить большее количество тестовых сценариев за короткое время.

4. **Точность:** Автоматизированные тесты минимизируют риск ошибок, связанных с невнимательностью тестировщика.

5. **Интеграция с CI/CD:** Автоматизированные тесты легко интегрируются в процессы непрерывной интеграции и доставки, что невозможно при ручном тестировании.

Этапы внедрения автоматизации:

1. **Анализ требований:** Определение целей автоматизации, выявление критичных для тестирования областей.

2. Выбор инструментов: Подбор подходящих инструментов и фреймворков (например, Selenium, JUnit, TestNG).

3. Разработка тестовых сценариев: Создание скриптов для автоматического выполнения тестов.

4. Настройка среды: Подготовка тестовой среды, включая серверы, базы данных и другие компоненты.

5. Запуск тестов: Выполнение тестов в автоматическом режиме.

6. Анализ результатов: Проверка отчетов об ошибках и их исправление.

7. Поддержка и обновление: Регулярное обновление тестов для поддержания их актуальности.

Преимущества автоматизированного тестирования:

1. Ускорение процесса тестирования:

– Автоматизированные тесты выполняются намного быстрее, чем ручные, особенно для регрессионного тестирования.

– Пример: Тестирование API может занять несколько минут вместо часов при ручном подходе.

2. Снижение затрат времени и ресурсов:

– После первоначальной настройки автоматизированные тесты могут быть запущены многократно без дополнительных усилий.

– Это освобождает тестировщиков для выполнения более сложных задач, таких как исследовательское тестирование.

3. Высокая точность и воспроизводимость результатов:

– Автоматизированные тесты выполняются строго по заданному алгоритму, что исключает ошибки, связанные с человеческим фактором.

– Каждый запуск теста дает одинаковый результат при одних и тех же входных данных.

4. Возможность масштабирования:

– Автоматизация позволяет тестировать большие системы с множеством функциональностей и различных конфигураций.

– Пример: Тестирование мобильного приложения на сотнях устройств с

помощью инструментов, таких как Appium.

5. Поддержка непрерывной интеграции и доставки (CI/CD):

- Автоматизированные тесты легко интегрируются в CI/CD-процессы, такие как Jenkins, GitLab CI или CircleCI.

- Это позволяет быстро получать обратную связь о состоянии продукта после каждого изменения кода.

Ограничения автоматизации:

1. Высокие начальные затраты на настройку:

- Разработка и настройка автоматизированных тестов требуют значительных временных и финансовых вложений.

- Пример: Написание сложных тестовых сценариев может занять недели или даже месяцы.

2. Необходимость технической подготовки команды:

- Для создания и поддержки автоматизированных тестов требуется знание языков программирования (например, Python, Java) и тестовых фреймворков.

- Командам, состоящим из тестировщиков без технического опыта, может быть сложно внедрить автоматизацию.

3. Невозможность полностью заменить ручное тестирование:

- Автоматизация не может охватить все аспекты тестирования, такие как исследовательское тестирование, проверка удобства использования (UX/UI) или тестирование субъективных характеристик.

- Пример: Оценка визуального дизайна или удобства интерфейса требует человеческого взгляда.

4. Зависимость от изменений в коде:

- При изменении функциональности или интерфейса приложения автоматизированные тесты могут перестать работать и потребовать обновления.

- Пример: Изменение структуры HTML-элементов на странице может "сломать" тесты, написанные с использованием Selenium.

5. Ограниченная эффективность для новых проектов:

— На ранних этапах разработки, когда требования часто меняются, автоматизация может быть неэффективной из-за необходимости постоянного обновления тестов.

6. Популярные инструменты автоматизированного тестирования

6.1. Инструменты для юнит-тестирования

Юнит-тестирование — это проверка отдельных модулей или компонентов кода на предмет их корректной работы. Для этого используются специализированные фреймворки, которые позволяют создавать и выполнять тесты.

1. JUnit (Java):

— Один из самых популярных фреймворков для юнит-тестирования в экосистеме Java.

— Поддерживает аннотации (`@Test`, `@Before`, `@After`) для организации тестовых сценариев.

— Пример использования: Тестирование методов классов в Java-приложениях.

2. NUnit (.NET):

— Аналог JUnit для платформы .NET.

— Предоставляет широкие возможности для написания и выполнения тестов, включая параметризованные тесты.

— Пример использования: Проверка логики бизнес-процессов в C#-приложениях.

3. PyTest (Python):

— Современный фреймворк для тестирования Python-приложений.

— Упрощает написание тестов благодаря своей простоте и поддержке плагинов.

— Пример использования: Тестирование функций, обрабатывающих данные в аналитических приложениях.

6.2. Инструменты для функционального тестирования

Функциональное тестирование проверяет, что приложение работает в соответствии с требованиями и выполняет свои задачи корректно. Для этого используются инструменты, которые взаимодействуют с интерфейсом приложения.

1. Selenium:

- Самый популярный инструмент для автоматизации тестирования веб-приложений.
- Поддерживает множество языков программирования (Java, Python, C#) и браузеров.
- Пример использования: Проверка работоспособности форм регистрации и авторизации на сайте.

2. Cypress:

- Современный фреймворк для тестирования веб-приложений, который отличается высокой скоростью и удобством использования.
- Встроенные инструменты отладки упрощают поиск ошибок.
- Пример использования: Тестирование динамического контента и взаимодействия с элементами страницы.

3. TestComplete:

- Коммерческий инструмент для автоматизации тестирования как веб-, так и десктоп-приложений.
- Поддерживает скриптовые языки (JavaScript, Python) и запись действий пользователя.
- Пример использования: Тестирование сложных рабочих процессов в корпоративных приложениях.

6.3. Инструменты для нагрузочного тестирования

Нагрузочное тестирование оценивает производительность системы под различными нагрузками (например, количество пользователей или объем данных).

1. Apache JMeter:

- Открытый инструмент для нагрузочного тестирования веб-приложений, API и баз данных.
- Поддерживает создание сложных сценариев нагрузки и генерацию отчетов.
- Пример использования: Проверка производительности сайта при 1000 одновременных пользователей.

2. Gatling:

- Современный инструмент для нагрузочного тестирования, ориентированный на производительность и удобство использования.
- Использует DSL (Domain-Specific Language) на основе Scala для написания тестов.
- Пример использования: Тестирование RESTful API на предмет их способности обрабатывать большое количество запросов.

3. LoadRunner:

- Коммерческий инструмент для нагрузочного тестирования крупных систем.
- Предоставляет широкие возможности для моделирования реальных условий работы приложения.
- Пример использования: Тестирование банкоматов или банковских систем на устойчивость к пиковой нагрузке.

6.4. Инструменты для тестирования мобильных приложений

Тестирование мобильных приложений направлено на проверку их работоспособности на различных устройствах и операционных системах.

1. Appium:

- Популярный инструмент для автоматизации тестирования мобильных приложений на iOS и Android.
- Поддерживает написание тестов на популярных языках программирования (Java, Python, JavaScript).

- Пример использования: Проверка функциональности приложения на разных версиях Android.

2. Espresso:

- Фреймворк для тестирования Android-приложений, разработанный Google.

- Предоставляет удобные инструменты для тестирования пользовательского интерфейса.

- Пример использования: Тестирование навигации между экранами в Android-приложении.

3. XCTest:

- Фреймворк для тестирования iOS-приложений, встроенный в Xcode.

- Поддерживает как юнит-тестирование, так и UI-тестирование.

- Пример использования: Проверка работоспособности кнопок и полей ввода в iOS-приложении.

Заключение

Автоматизированное тестирование занимает ключевую роль в современной разработке программного обеспечения, обеспечивая высокое качество, надежность и стабильность приложений. Оно позволяет командам разработчиков и тестировщиков сократить время на проверку функциональности, выявлять ошибки на ранних этапах и поддерживать процессы непрерывной интеграции и доставки (CI/CD). Автоматизация становится неотъемлемой частью жизненного цикла ПО, особенно в условиях растущей сложности систем и сжатых сроков разработки.

В ходе исследования были рассмотрены основные виды тестирования, такие как функциональное, нагрузочное, регрессионное и юнит-тестирование, а также их роль в обеспечении качества продукта. Особое внимание уделено преимуществам автоматизированного тестирования, включая ускорение процесса, снижение затрат времени и ресурсов, высокую точность и возможность масштабирования. Однако важно учитывать ограничения автоматизации, такие как высокие начальные затраты и невозможность полностью заменить ручное тестирование.

Анализ популярных инструментов, таких как Selenium, JUnit, Apache JMeter, Appium и Cucumber, показал, что выбор инструмента зависит от типа приложения, языка программирования и специфики проекта. Грамотное внедрение этих инструментов позволяет значительно повысить эффективность тестирования и минимизировать риски выпуска некачественного продукта.

Список использованных источников

1. Петренко А. В., Махненко А. С. Основы тестирования программного обеспечения. — Москва: ДМК Пресс, 2019. — 240 с.
2. Боев Н.В. Проектирование информационных систем. — Москва: Издательский дом «Питер», 2019. — 384 с.
3. Иванов А. П. Автоматизированное тестирование веб-приложений // Программные продукты и системы. — 2020. — № 3. — С. 45-52.
4. Ковалевский А. Введение в автоматизированное тестирование [Электронный ресурс]. — URL: <https://www.testing.ru/articles/automation-testing> (дата обращения: 18.03.2025).