

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЙ
(на примере портала Саратовского госуниверситета)
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование
информационных систем
факультета компьютерных наук и информационных технологий
Масеева Никиты Евгеньевича

Научный руководитель:

доцент

Е. В. Кудрина

подпись, дата

Зав. кафедрой:

доцент, к. ф.-м. н.

М. В. Огнева

подпись, дата

Саратов 2025

ВВЕДЕНИЕ

Актуальность темы. В современном мире программное обеспечение используется практически во всех сферах жизни, гигантские суммы тратятся на разработку разнообразных программ, востребованных в промышленности, бизнесе, индустрии развлечений, образовании и медицине. Задача снижения стоимости разработки программного обеспечения и улучшения качества выпускаемой продукции является одной из наиболее актуальных в индустрии информационных технологий. Автоматизация тестирования позволяет значительно сократить затраты компаний разработчиков, сэкономить время и ресурсы, расходуемые на тестирование, снизить риск выпуска на рынок некачественного продукта.

Автоматизированное тестирование заключается в написании автотестов – программ, направленных на выполнение заложенных в них тестовых сценариев. Благодаря им разработчик может по истечению короткого периода времени получить информацию о том, были ли обнаружены в ходе выполнения данного сценария ошибки или тест пройден успешно. Для разработки и запуска автотестов используются специальные системы автоматизированного тестирования.

Главной идеей автоматизированного тестирования является полная замена человеческого труда. Программист единожды создает программу, которая проверяет все тестовые сценарии. Далее, сценарии только поддерживаются в актуальном рабочем состоянии. Создание таких программ позволяет снизить присутствие ручного тестирования в жизненном цикле разрабатываемого ПО. Данный подход экономит время сотрудников отдела по контролю качества, а также ощутимо ускоряет работу команды разработчиков, и весь процесс разработки ПО в целом. Многократно возрастает тестовое покрытие. Есть возможность проводить проверки нового или модифицированного приложения на соответствие установленным требованиям по функциональности и производительности.

В итоге, внедряя в проект этап тестирования, мы уменьшаем траты на создание продукта и повышаем его качество. Внедряя автоматизированное тестирование, мы экономим еще и время, а также повышаем процент качества еще больше, что в настоящее время является главным в процессе разработки ПО [1].

Цель бакалаврской работы – изучение методологии автоматизированного тестирования и её применение для UI сервиса SGU.RU

Поставленная цель определила **следующие задачи**:

1. Рассмотреть теоретические основы и методологии автоматизированного тестирования;
2. Описать инструментальные средства, применяемые для автоматизированного тестирования;
3. Разработать техническую документацию для автоматизированного тестирования портала sgu.ru;
4. Подобрать вспомогательные алгоритмы для автоматизированного тестирования портала sgu.ru;
5. Выполнить автоматизированное тестирование портала sgu.ru;
6. Проанализировать полученные результаты.

Методологические основы исследования в области автоматизированного тестирования представлены в работах Фирсова С. А., Пакшина А. В. [1], Геннадия Алпаева [2], Куликова С. С. [3], Кулакова К. А., Димитрова В. М. [4], Веденеева Г. С., Федосеева Д. А., Старовойта К. А. [5].

Практическая значимость бакалаврской работы заключается в проектировании и реализации системы автоматизированного отслеживания некорректных ссылок на официальном портале Саратовского госуниверситета www.sgu.ru.

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 8 приложений. Общий объем работы – 89 страниц, из них 53 страниц – основное

содержание, включая 11 рисунков и 2 таблицы, цифровой носитель в качестве приложения, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретическая часть» посвящен теоретическим основам и методологиям автоматизированного тестирования, а также инструментальным средствам, используемым для этого.

Автоматизированное тестирование программного обеспечения - процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически при помощи инструментов для автоматизированного тестирования.

Примеры методологий автоматизированного тестирования:

1. Test Driven Development (разработка через тестирование). Она основывается на создании тестов для написания кода, что позволяет убедиться в правильности работы программы на ранних стадиях разработки и снизить объём ошибок до 80 %;
2. Behavior Driven Development (разработка, основанная на поведении). Она основывается на понимании требований заказчика и написании тестов на языке, близком к естественному языку, такие как Cucumber или JBehave;
3. Acceptance Test Driven Development (разработка через приемочные тесты) - методология, которая предполагает написание автоматизированных тестов на основе требований к системе и проверку соответствия ее функционала этим требованиям.

Есть наиболее часто встречаемые методологии, объединённые в различные классификации:

1. Классификация по природе тестируемых приложений;
2. Классификация по целям и задачам;
3. Классификация по привлечению конечных пользователей;
4. Классификация по степени важности тестируемого функционала;
5. Классификация по уровню архитектуры приложения;

6. Классификация по запуску кода на исполнение.

Инструменты для автоматизированного тестирования могут представлять собой полнофункциональными средами или библиотеками.

Полнофункциональные среды для автоматизированного тестирования представляют собой инструменты с возможностями записи, редактирования, запуска тестов и формирования отчётности:

1. HP Quick Test Professional – это основной инструмент автоматизации функционального тестирования компании HP (ныне HP Unified Functional Testing);
2. Geb – фреймворк тестирования для браузеров;
3. JWebUnit – фреймворк тестирования на основе Java и одно из расширений JUnit, используемый для интеграционного, регрессионного и функционального тестирования;
4. Smart Bear Test Complete - полнофункциональная система, которая позволяет автоматизировать тестирование приложений.

Библиотеки для автоматизированного тестирования - представляют собой наборы функций и методов, которые используются для написания тестовых скриптов:

1. Selenium - одна из самых популярных библиотек, которая занимает уже не первый год лидирующие места;
2. TestNG - среда тестирования, основанная на JUnit и NUnit - конструкции Java, которые позволяют добавлять некоторые метаданные к классам, полям и методам;
3. Apache HttpClient - библиотека на Java для обработки HTTP-запросов и ответов.

Второй раздел «Практическая часть» содержит подробное описание этапов практической части работы.

Были определены некоторые причины появления некорректных ссылок: Удаление страницы, Переход на HTTPS без редиректа с HTTP,

Человеческий фактор, Изменение URL без редиректа, Технические проблемы.

На основе анализа требований заказчика была сформирована техническая документация: тест-план, чек-листы и тест-кейсы для проведения тестирования различных типов ссылок, таких как файлы, телефоны, почты, обычные ссылки и изображения.

Были сформированы критерии приемки и качества, которые являются общепринятыми в IT кампаниях:

1. Приемочные критерии: успешное прохождение 100 % тест-кейсов уровня дымового (Smoke) тестирования и 90 % тест кейсов уровня критического пути (Critical Path) при условии устранения % дефектов критической и высокой важности. Итоговое покрытие требований тест-кейсами должно составлять не менее 90 %;
2. Критерии начала тестирования: выход сборки согласно расписанию;
3. Критерии приостановки тестирования: переход к тесту критического пути только при успешном прохождении 100 % тест-кейсов Smoke теста;
4. Критерии возобновления тестирования: направление более 50 % обнаруженных на предыдущей итерации дефектов;
5. Критерии завершения тестирования: выполнение 100 % запланированных на итерацию тест-кейсов.

Общий объем разработанных документов составляет 9,13 Мбайт. Количество чек-листов составляет 17. Количество тест-кейсов также составляет 17. Объем архива с проектом на Java составляет 8,78 Мбайт, а объем архива с проектом на Python — 33,5 Кбайт. Количество строк кода в тестовых классах в промежутке от 45 до 270.

Планируется, что разработанные тестовые классы будут регулярно использоваться командой разработчиков — раз в две недели или раз в месяц — для проверки работоспособности приложения.

Структуру веб-приложения можно описать с помощью теории графов, где узлы представляют страницы или элементы интерфейса, а ребра — переходы между ними. При работе с порталом www.sgu.ru необходимо использовать алгоритмы обхода для проверки всех ссылок на объекты, избегая повторной проверки и игнорируя переходы на внешние ресурсы, чтобы предотвратить бесконечное выполнение алгоритма. Для ускорения обхода следует фильтровать ссылки, удаляя параметры, не влияющие на уникальность страницы, например, параметры сессии или перенаправления. Также в процессе разработки тестовых классов необходимо было исключить страницы с сегментом /en в URL по просьбе команды разработчиков из-за технических проблем. Важно учитывать систему защиты от автоматических запросов на портале: для избежания блокировки при переходе к следующей ссылке необходимо выделять время ожидания.

При выполнении практической части использовался следующий стек технологий: Selenium WebDriver, JUnit, Apache HttpClient, Requests.

При работе с инструментом Selenium WebDriver и использованием алгоритма обхода в глубину время полного обхода портала заняло 17 часов 48 минут, а при использовании алгоритма обхода в ширину время выполнения составляло 16 часов 54 минут. Было принято решение выбрать алгоритм обхода в ширину и применить алгоритм Рабина-Карпа для хеширования страниц, поскольку до этого сравнение ссылки с посещенной было по строкам, что замедляло время выполнения алгоритма. Переход к сравнению по хешам позволил ускорить проверку ссылок в результате чего время выполнения алгоритма уменьшилось до 15 часов 3 минут. Однако алгоритм обхода в глубину оказался более эффективным по сравнению с обходом в ширину, поскольку большинство некорректных ссылок располагалось во вложенных разделах и внутренних элементах структуры страницы, а не на её лицевой части.

Структура системы автоматизированного тестирования портала sgu.ru, разработанная на Java, содержит шесть подкаталогов. В каждом из них размещены наборы тестовых классов, предназначенных для проведения тестирования веб-приложения www.sgu.ru в браузере Google. Также в рамках проекта, созданного на Python, были разработаны четыре дополнительных набора тестовых классов.

В результате проведения первого этапа автоматизированного тестирования было выявлено 689 некорректных ссылок: 442 некорректные ссылки на файлы, 71 некорректный телефон и почтовый адрес, 172 некорректные ссылки и 4 некорректные ссылки на изображения.

Виды ошибок на стороне клиента: 403, 404, 418.

Виды ошибок на стороне сервера: 500, 502, 504.

Время выполнения тестовых классов на первом этапе:

1. Тестирование файлов - 17 часов 52 минуты;
2. Тестирование телефонов и почт - 7 часов 36 минут;
3. Тестирование обычных ссылок - 13 часов 46 минут;
4. Тестирование ссылок на изображения - 11 часов 14 минут.

Результаты выполнения тестовых классов были получены в текстовых файлах, проанализированы и оформлены в excel таблице. Полученный отчет был передан команде разработчиков для исправления ошибок. Процесс исправления занял один месяц. После процесса исправления была получена обратная связь для начала проведения второго этапа тестирования, с целью выявления новых или оставшихся проблем.

В результате проведения второго этапа автоматизированного тестирования было выявлено 267 некорректных ссылок: 69 некорректных ссылок на файлы, 18 некорректных телефонов и почтовых адресов, 169 некорректных ссылок и 11 некорректных ссылок на изображения.

Виды ошибок на стороне клиента: 403, 404.

Виды ошибок на стороне сервера: 500.

Время выполнения тестовых классов на первом этапе:

1. Тестирование файлов - 10 часов 32 минуты;
2. Тестирование телефонов и почт - 4 часа 48 минут;
3. Тестирование обычных ссылок - 12 часов 29 минут;
4. Тестирование ссылок на изображения - 10 часов 42 минуты.

Полученные результаты также были оформлены в виде excel таблицы и отправлены разработчикам для исправления ошибок. Исправления ошибок после первого этапа привели к заметному снижению времени выполнения тестовых классов. Также уменьшилось количество ошибок на стороне клиента благодаря тому, что количество ошибок с файлами сократилось в 6 раз, а с телефонами и почтами — в 3 раза. Поскольку платформа СГУ активно используется различными пользователями, невозможно избавиться от всех ошибок, но можно снизить вероятность появления большого числа новых ошибок, регулярно проводя этапы тестирования, что позволит уменьшить нагрузку на команду разработчиков.

ЗАКЛЮЧЕНИЕ

Задачи выпускной квалификационной работы были выполнены. В результате проведенных исследований, было выполнено автоматизированное тестирование портала Саратовского госуниверситета с применением инструментов для автоматизированного тестирования. На основе двух этапов тестирования была собрана и передана разработчикам полная информация о некорректных ссылках в виде отчетов по тестированию. Проекты с наборами тестовых классов были переданы команде разработке для поддержания работоспособности рассмотренной платформы. Результаты выпускной квалификационной работы были представлены на студенческой научно-практической конференции факультета компьютерных наук и информационных технологий 21 апреля 2025 года.

Отдельные части бакалаврской работы были опубликованы/представлены на конференции: Масеев Н. Е. АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЙ (на примере портала Саратовского госуниверситета) // «СНК 2025 ИиП».

Основные источники информации:

1. Фирсова С.А., Пакшин А.В. Организация процесса автоматизированного тестирования WEB-приложений на базе Фреймворка Selenium. // ОгарёвOnline, Компьютерные и информационные науки : [сайт]. – 2021. – URL: <https://cyberleninka.ru> (дата обращения: 16.09.2024);
2. Геннадий Алпаев. Автоматизация тестирования от «А» до «Ы». Москва: ДМК Пресс, 2015. — 82 с. — URL: <https://alpaev.com/TestingAutomation.pdf> (Дата обращения 20.09.2024). Загл. с экр. Яз. Рус;
3. Куликов. С. С. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. — 3-е изд. — Минск: Четыре четверти, 2020. — 312 с. (Дата обращения 21.09.2024). Загл. с экр. Яз. Рус;

4. Кулаков, К. А., Димитров, В. М. Основы тестирования программного обеспечения: Учебное электронное пособие для обучающихся Института математики и информационных технологий / К. А. Кулаков, В. М. Димитров. — Петрозаводск: Издательство ПетрГУ, 2018. — 57 с. (Дата обращения 24.09.2024). Загл. с экр. Яз. Рус;
5. Веденеев Г. С., Федосеев Д. А., Старовойт К. А. Анализ инструментов для проведения автоматизированного функционального тестирования программного обеспечения // Киберленинка. — 2017. — 13 с. — URL: <https://cyberleninka.ru/article/n/analiz-instrumentovdlya-provedeniya-avtomatizirovannogo-funktsionalnogo-testirovaniyaprogrammnogo-obespecheniya/viewer> (дата обращения: 10.12.2024).