

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра математического обеспечения вычислительных комплексов и
информационных систем

**РАЗРАБОТКА И РЕАЛИЗАЦИЯ АВТОМАТИЧЕСКИХ ТЕСТОВ ДЛЯ
WEB-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛИЗМА**
АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 273 группы
направления 01.04.02 Прикладная математика и информатика
факультета компьютерных наук и информационных технологий
Ефимова Ильи Сергеевича

Научный руководитель:

доцент
(должность, уч.степень, звание)

_____ (подпись, дата)

Борзов И. А.
(ФИО)

Зав. кафедрой:

профессор, д.ф.-м.н.
(должность, уч.степень, звание)

_____ (подпись, дата)

Андрейченко Д.К.
(ФИО)

Саратов 2018 г.

ВВЕДЕНИЕ

Актуальность темы. Сейчас существует множество web-приложений разнообразной тематики. Популярностью пользуются не только развлекательные порталы, но и важные интернет-ресурсы цифровой экономики.

Один из таких интернет-ресурсов – справочно-информационный интернет-портал «Госуслуги». Актуальность выбора обусловлена тем фактом, что в России, согласно с общемировой практикой, проводится административная реформа, направленная на оптимизацию исполнения функций органами исполнительной власти. В целях повышения удобства при очном взаимодействии заявителей с государственными органами, ведётся работа по созданию многофункциональных центров предоставления государственных и муниципальных услуг, которые обеспечивают предоставление комплекса взаимосвязанных между собой государственных услуг федеральными органами исполнительной власти, органами исполнительной власти субъектов Российской Федерации и органами местного самоуправления по принципу «единого окна». При этом межведомственное взаимодействие, необходимое для оказания государственной услуги (включая необходимые согласования, получение выписок, справок и др.), должно проходить без участия заявителя [1].

На данном портале обрабатывается массив персональных данных значительных масштабов, и, с учётом предоставляемого доступа к информации такого рода, возрастает и важность контроля качества предоставляемых населению услуг.

Автоматизация тестирования позволяет значительно сократить время выполнения тестирования в сравнении с ручным тестированием.

Актуальность автоматизации кросс-браузерного тестирования в современном мире и определила цель и задачи магистерской работы.

Цель магистерской работы: разработка и реализация автоматических тестов для web-приложения с использованием параллелизма

(на примере системы предоставления государственных и муниципальных услуг порталом государственных услуг Российской Федерации).

Задачи магистерской работы:

1. Систематизировать знания в области тестирования интернет-порталов;
2. Составить тестовый план с подробным описанием тестовой стратегии, инструментов тестирования, оценкой рисков и объёмом работ;
3. Подготовить и провести автоматизированное тестирование справочно-информационного портала «Госуслуги»;
4. Предоставить результаты тестирования;
5. Использовать технологии параллелизма при кросс-браузерном автоматическом тестировании.

Методологические основы автоматизации тестирования web-приложений представлены в работах Куликова [2], Майерса [3], Котлярова [6], Адеркина [7], а также в книге Янга «Написание параллельных тестов» [17].

Практическая значимость магистерской работы. В ходе выполнения практической части магистерской работы был реализован фреймворк параллельных автоматических тестов для тестирования интернет-портала «Госуслуги». Отчёт об обнаруженных дефектах web-приложения был отправлен в службу технической поддержки портала.

Структура и объем магистерской работы. Работа состоит из введения, пяти разделов, заключения, списка использованных источников информации и 2 приложения. Общий объем работы – 67 страниц, из них 52 страницы – основное содержание, включая 21 рисунок и 11 таблиц, цифровой носитель в качестве приложения, список использованных источников информации – 24 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические основы тестирования программного обеспечения» посвящен описанию теоретических основ

тестирования программного обеспечения в целом, а также более подробному изучению unit-тестирования. Раздел содержит несколько подразделов.

Основные определения – включает в себя основные определения (термины), связанные с тестированием программного обеспечения.

История тестирования – подраздел посвящен изучению теоретических основ тестирования программного обеспечения и истории их развития.

Жизненный цикл тестирования включает в себя описание полного цикла тестирования программного обеспечения на примере современной модели разработки программного обеспечения, а также дана оценка роли автоматизации в ней.

Второй раздел «Планирование тестирования» посвящен составлению тест-плана. Тестплан – это документ, описывающий и регламентирующий перечень работ по тестированию, а также соответствующие техники и подходы, стратегию, области ответственности, ресурсы, расписание и ключевые даты.

В данном разделе описан перечень условий, при выполнении которых команда приступает к тестированию и завершает его. Описаны методы и уровни тестирования, способ документирования обнаруженных дефектов приложения.

Третий раздел «Описание тест-кейсов» посвящен созданию тест-кейсов, чье выполнение будет автоматизировано. Тест-кейс – это набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.

Четвёртый раздел «Применение автоматических тестов на практике» посвящен разработке фреймворка автоматических тестов и анализу полученного отчёта.

Для автоматизации был написан фреймворк автоматизированных тестов на языке Java с применением тестовых библиотек TestNG.

Выбор Java в качестве языка программирования обусловлен тем, что это самый популярный язык программирования десктопных приложений, согласно опросу Stackoverflow [11].

Среди двух наиболее популярных фреймворков тестирования: TestNG или JUnit, был выбран первый, т.к. этот фреймворк позволяет создавать зависимости между группами тестов, а также параллелизм выполнения тестов указывается в настройках тест-сюэтов, таким образом, в данной работе конфигурации параллелизма будут более наглядными. Разработчики TestNG изначально предполагали возможность параллельного выполнения тестов.

Параметризация тестов из таблицы приложения А проходила с помощью аннотации TestNG `@DataProvider`. Пример её использования, а также описание классов других разработанных автоматических тестов и код приложения целиком, приведены в приложении В.

Применялся паттерн проектирования Page Object [10], для того, чтобы инкапсулировать работу с отдельными элементами страницы, что позволяет уменьшить количество кода и его поддержку. Таким образом получено разделение кода тестов и описания страниц, а также объединение всех действий по работе с веб-страницей в одном месте. Суть данного паттерна проектирования в том, что для каждой страницы тестируемого приложения создаётся отдельный объект, методы которого инкапсулируют логику работы с отдельными элементами. Таким образом, Page Object позволяет избежать дублирования локаторов в тестах.

Для поиска веб-элемента на странице был реализован метод неявных ожиданий [12]. Неявные ожидания конфигурируют экземпляр веб-драйвера Selenium делать многократные попытки найти элемент на веб-странице в течении заданного периода времени, если элемент не найден сразу. Только по истечении этого времени WebDriver бросит исключение `ElementNotFoundException`.

Результатом решения поставленной задачи является готовый набор автоматизированных тестов с применением инструмента для автоматизированного управления браузерами Selenium.

После выполнения автоматических тестов был составлен отчёт, который преобразован в виде веб-страницы с помощью генератора отчётов Allure от Яндекс. Обнаружено 5 дефектов приложения.

После перепроверки результатов был составлен подробный отчёт дефектов и отправлен по адресу технической поддержки портала «Госуслуги». Сообщение было отправлено 12 апреля 2018 года. Ответного сообщения или признаков исправления обнаруженных дефектов на момент написания данной работы нет.

Пятый раздел «Исследование возможностей параллелизма выполнения автоматических тестов» посвящен сравнению двух способов параллелизма выполнения автоматических тестов. Параллельное тестирование – это процесс одновременного запуска нескольких тестов по нескольким комбинациям операционных систем и браузеров. Процесс автоматизирован и часто выполняется на виртуальных машинах.

Поскольку параллельное тестирование выполняется быстрее последовательного, оно позволяет протестировать совместимость на большем количестве примеров за одинаковое время. Если речь идёт о выпуске нового приложения и есть необходимость протестировать функциональность пользовательского интерфейса (UI), отделу тестирования не нужно ограничивать себя тестированием только самых популярных комбинаций ОС и браузера или мобильных устройств – можно протестировать гораздо больше комбинаций, которые важны для конечного пользователя.

По ходу написания автоматических тестов для веб-портала «Госуслуги» использовался Apache Maven – программная платформа для автоматизации сборки проектов на основе описания их структуры в файлах

на языке POM (Project Object Model, модель объекта проекта), являющемся подмножеством XML.

POM-файл является базовым модулем Maven. Это специальный XML-файл, который всегда хранится в базовой директории проекта и называется pom.xml.

Файл POM содержит информацию о проекте и различных деталях конфигурации, которые используются Maven для создания проекта.

Требования к сборке сильно зависят от проекта. Плагины – это дополнение к проекту, способ расширить функциональность maven в больших диапазонах.

Для того, чтобы задать параметры параллелизма в POM-файле, достаточно в конфигурации плагина Maven Surefire задать, что именно будет распараллелено: методы (methods) или классы (classes), а также число потоков (threadCount).

TestNG – это платформа тестирования. Для того, чтобы задать параметры параллелизма в TestNG-файле, необходимо задать атрибуты у тега определения набора тестов (test-suite). Конкретно, что именно тестировать parallel (возможны значения methods, classes и tests), число потоков thread-count и число потоков у data provider – data-provider-thread-count. Атрибут preserve-order означает сохранение порядка выполнения тестов.

Для наглядной демонстрации преимуществ параллельного запуска автоматических тестов были рассмотрены кросс-браузерные параметризованные тесты страницы «Помощь и Поддержка». То есть два теста (поиск с помощью кнопки «поиск» и с помощью всплывающей подсказки) с разными наборами данных (в реализованной параметризации на данном проекте оба эти теста должны будут запускаться для трёх разных параметров – поисковых запросов) запускаются для трёх разных браузеров: Google Chrome, Mozilla Firefox и Microsoft Edge.

Рассмотрены две параллельные реализации с помощью тегов в POM-файле и атрибутов в TestNG-файле. Прохождение тестов проводилось с помощью процессора Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz 3.60 GHz.

Параллельное выполнение с помощью настроек в POM-файле или конфигураций TestNG-файла прошло примерно с одинаковым временем, однако помимо того, что по мнению автора, конфигурирование TestNG-файла более удобно и наглядно, есть важное отличие – TestNG позволяет параллелизм выполнения параметризованных тестов с помощью атрибута `data-provider-thread-count` у тест-сюта.

Согласно наблюдениям, можно сделать вывод, что суммарное время параллельного выполнения тест-сюта не может быть меньше времени выполнения самого длинного тест-кейса.

Ускорение выполнения тестов значительно замедлилось при достижении выполнения с четырьмя потоками (на четырёх-ядерном процессоре). В дальнейшем время выполнения могло даже возрасти, потому что требовались дополнительные вычислительные ресурсы для планирования.

Самое минимальное среднее время выполнения было достигнуто на TestNG при двух потоках выполнения тестов и ещё двух потоков параллелизма параметризации тестов (в данном случае использовались значения `thread-count="2"` `data-provider-thread-count="2"`) – одна минута и сорок семь секунд. Так как при конфигурации в POM-файле параллелизм выполнения параметризованных автоматических тестов невозможен (то есть принимается, что этот поток всего лишь один), то в данном случае это заняло в среднем три минуты и семнадцать секунд, то есть в полтора раза дольше.

Таким образом, самым оптимальным выбором в вопросе об инструментах параллелизма выполнения автоматических тестов оказался TestNG, потому что по сравнению с настройками параллельного запуска в POM-файле, можно указать число потоков при параметризованном

выполнении тестов (один и тот же тест перезапускается с разными наборами данных).

ЗАКЛЮЧЕНИЕ

В ходе выполнения магистерской работы были выполнены все поставленные задачи, а именно был разработан набор параллельных кросс-браузерных автоматических тестов на объектно-ориентированном языке программирования Java для тестирования основного функционала портала «Госуслуги».

Были рассмотрены методы параллелизма выполнения автоматического тестирования и необходимые требования к автоматическим тестам.

В данной работе было приведено сравнение двух способов параллельного выполнения автоматических тестов с последовательным запуском.

По тематике научной работы были представлены доклады:

«Анализ покрытия кода unit-тестами Dotcover» на международной научно-практической конференции «Информационные технологии в образовании-2016» в Саратовском национальном исследовательском государственном университете имени Н.Г.Чернышевского 2 ноября 2016 [23].

«Автоматизация тестирования государственных электронных услуг» на научно-практической конференции «Цифровая экономика: перспективы и вызовы» в секции «Цифровая экономика в современной России: социально-правовое измерение» в Саратовском социально-экономическом институте РЭУ им. Г.В. Плеханова 17 апреля 2018 [24].

СПИСОК ЛИТЕРАТУРЫ

1. Официальный интернет-портал правовой информации. Распоряжение от 6 мая 2008 г. N 632-р Электронная версия: <http://pravo.gov.ru/proxy/ips/?docbody=&nd=102121853&rdk=0> (дата проверки: 20.03.2018) Загл с экр. Яз. рус.
2. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. – Минск: Четыре четверти, 2017. – 312 с. ISBN 978-985-581-125-2
3. Майерс, Г., Баджетт, Т., Сандлер, К. Искусство тестирования программ, 3-е издание The Art of Software Testing, 3rd Edition. – М.: Изд-во «Диалектика», 2012. – 272 с. – ISBN 978-5-8459-1796-6
4. Брукс, Ф.П. Как проектируются и создаются программные комплексы. – М.: Мир, 1979.
5. Макконнелл, С. Rapid Development: Taming Wild Software Schedules. Изд-во «Microsoft Press», 1996. – ISBN 1-55615-900-5.
6. Котляров, В.П., Коликова, Т.В. – Основы тестирования программного обеспечения. – Изд-во «Бином. Лаборатория знаний», 2009. – 285 с. – ISBN 5-94774-406-4
7. Адеркин, А. Тестирование программного обеспечения [Электронный ресурс]. [Электронный ресурс] : [сайт]. URL: <http://aderkin.ru/> (дата обращения: 24.04.2013) Загл с экр. Яз. рус.
8. Определение и принципы тестирования [Электронный ресурс]. URL: <http://www.rae.ru/monographs/141-4632> (дата обращения: 19.04.2013).
9. К.Бэк, Agile-манифест разработки программного обеспечения. Электронная версия: <http://agilemanifesto.org/iso/ru/manifesto.html> (дата проверки: 4.11.2018) Загл с экр. Яз. рус.
10. М. Черняк, Оценка эффективности автоматизации тестирования. Электронная версия: <http://www.a1qa.ru/blog/otsenka-effektivnosti-avtomatizatsii-testirovaniya/> (дата обращения: 10.06.2018) Загл с экр. Яз. Рус.
11. Stackoverflow. Most Popular Technologies Электронная версия: <https://insights.stackoverflow.com/survey/2018/#technology> (дата обращения:

- 17.06.2018) Загл с экр. Яз. Англ.
12. Getting started with Page Object Pattern for your Selenium tests [Электронный ресурс] URL: <https://www.pluralsight.com/guides/software-engineering-best-practices/getting-started-with-page-object-pattern-for-your-selenium-tests> (дата обращения: 4.11.2018) Загл с экр. Яз. англ.
 13. SeleniumHQ. Kissmetrics Blog. WebDriver: Advanced Usage. [Электронный ресурс] URL: https://www.seleniumhq.org/docs/04_webdriver_advanced.jsp (дата обращения: 4.11.2018) Загл с экр. Яз. англ.
 14. Allure – Технологии Яндекса. [Электронный ресурс] URL: <https://tech.yandex.ru/allure/> (дата обращения: 4.11.2018) Загл с экр. Яз. рус.
 15. Kissmetrics Blog. How Loading Time Affects Your Bottom Line. [Электронный ресурс] URL: <https://blog.kissmetrics.com/loading-time/> (дата обращения: 4.11.2018) Загл с экр. Яз. англ.
 16. «7 reasons to move to parallel testing» [Электронный ресурс]. URL: <https://devops.com/7-key-reasons-make-move-sequential-parallel-testing/> (дата обращения: 04.06.2018) Загл. с экр. Яз. англ.
 17. Янг, Д. – Parallel Spelling Tests. – Изд-во « Hodder Education», 1998. – 40 с. – ISBN 978-0340730935
 18. Шреста, П. – Validating Parallel Language Tests: A Study of Two 'Parallel' National Reading Tests in Nepal. – Изд-во « VDM Verlag», 2009. – 100 с. – ISBN 5-94774-406-4
 19. Линк, Дж. – Unit Testing in Java: How Tests Drive the Code (The Morgan Kaufmann Series in Software Engineering and Programming). – Изд-во « Morgan Kaufmann», 2003. – 376 с. – ISBN 978-1558608689
 20. Кочиаро, К. – Selenium Framework Design in Data-Driven Testing: Build data-driven test frameworks using Selenium WebDriver, AppiumDriver, Java, and TestNG. – Изд-во «Packt Publishing - ebooks Account», 2018. – 354 с. – ISBN 978-1788473576
 21. Бёуст, К. – Next Generation Java Testing: TestNG and Advanced Concepts.

- Изд-во «Addison-Wesley Professional», 2007. – 512 с. – ISBN 978-0321503107
22. GitHub. Yefimov – EGT (E-Government Testing). [Электронный ресурс] URL: <https://github.com/Yefimov/EGT> (дата обращения: 4.11.2018) Загл с экр. Яз. англ.
23. Ефимов, И.С. Анализ покрытия кода unit-тестами Dotcover [Текст] // Информационные технологии в образовании: Материалы VIII Международ. научно-практ. конф. – Саратов: ООО «Издательский центр «Наука»», 2016. ISSN 978-5-9758-1650-4. С. 470-474.
24. Ефимов, И.С. Автоматизация тестирования государственных услуг [Текст] // Научно-практический журнал «Наука и общество». Серия «Прикладная математика и информатика». 2018. № 1 (30). ISSN 2223-9774. С. 184-188.