

Разработка инструмента автоматизированного тестирования

Д.В. Набиева
Факультет информатики и робототехники
Уфимский государственный авиационный
технический университет
Уфа, Россия
e-mail: dianotka97@gmail.com1

Г.Н. Верхотурова
Факультет информатики и робототехники
Уфимский государственный авиационный
технический университет
Уфа, Россия
e-mail: verhoturova.gn@yandex.ru2

Аннотация¹

Данная статья посвящена актуальной проблеме тестирования веб-интерфейсов приложений. Рассматриваются основные преимущества автоматизированного тестирования, подходы и инструменты автоматизированного тестирования, а также приведено основание для разработки собственного инструмента.

1. Введение

Тестирование программного обеспечения – это процесс исследования программного обеспечения с целью выявления дефектов и определения соответствия между реальным и ожидаемым поведением программного обеспечения [1]. В более широком понимании тестирование – это техника контроля качества программного обеспечения, включающая в себя создание тестов, выполнение тестирования и анализ полученных результатов.

Существует множество видов тестирования программного обеспечения. Все эти виды классифицируются по разным категориям. По степени автоматизации выделяют два вида тестирования:

- Ручное тестирование
- Автоматизированное тестирование

Автоматизированное тестирование – это метод тестирования программного обеспечения, который выполняется с использованием специальных программных средств автоматизированного тестирования для выполнения набора тестовых примеров. Ручное тестирование, напротив, выполняется человеком, сидящим перед компьютером и тщательно выполняющим шаги теста.

Последовательные циклы разработки программного обеспечения требуют многократного выполнения одного и того же набора тестов [2]. Используя

инструмент автоматизации тестирования, можно записать этот набор тестов и при необходимости воспроизводить его для разных данных, условий и ситуаций. После автоматизации набора тестов вмешательство человека не требуется. Цель автоматизации – уменьшить количество тестовых примеров, которые нужно выполнять вручную, а не полностью исключить ручное тестирование.

2. Программная реализация

Программные средства для автоматизации тестирования также позволяют вводить тестовые данные в тестируемую систему, сравнивать ожидаемые и фактические результаты и создавать подробные отчеты о тестах.

Автоматизированное тестирование – это лучший способ повысить эффективность процесса, расширить охват тестированием и увеличить скорость выполнения при проверке качества и надежности программного обеспечения [3].

Использование автоматизированного тестирования программного обеспечения необходимо по следующим причинам [4]:

- ручное тестирование всех рабочих процессов, всех полей, всех негативных сценариев требует значительных затрат времени и денег;
- многоязычные сайты сложно тестировать вручную;
- ручное тестирование может стать скучным и, следовательно, подверженным ошибкам;
- появляется возможность многократного запуска автоматических тестов по расписанию;
- увеличивается скорость выполнения тестов и охват тестированием.

Современная разработка программных продуктов зависит от автоматизированного тестирования с использованием тестовых фреймворков. Цель гарантировать, что исходный код программного обеспечения готов для включения в пакеты и передачи их конечным пользователям.

Мобильные приложения, веб-разработка, игры, e-commerce — те отрасли, где последовательное прохождение стадий проектирования, сборки и

Труды Восьмой всероссийской научной конференции "Информационные технологии интеллектуальной поддержки принятия решений", 6-9 октября, Уфа-Ставрополь, Ханты-Мансийск, Россия, 2020

тестирования должно быть максимально быстрым. На этапе разработки возможность непрерывного тестирования и доставки кода позволяет увеличить скорость и качество готовых решений.

Непрерывное (несколько раз в день) слияние рабочих копий программного кода в общую основную ветвь и тестирование результатов оформилось в «концепцию непрерывной интеграции и доставки софта» (англ. Continuous Integration & Continuous Delivery или сокращённо CI/CD). Впервые её предложил Гради Буч в 1991 году [5].

Главные цели CI/CD — свести к минимуму ошибки, ускорить сборку и повысить качество конечного продукта. В CI/CD тестирование проводится всякий раз, когда в код вносятся изменения. Обнаружение ошибок на ранней стадии разработки существенно экономит время и ресурсы команды, ведь чем позднее выявляется баг, тем труднее и дороже его исправлять. Что касается ускорения, то автоматизация CI/CD помогает значительно оптимизировать все рутинные процессы сборки.

Чтобы получить максимальную отдачу от тестирования, тесты также могут быть интегрированы в конвейеры управления версиями.

Но есть одна область, где автоматизированное тестирование сложно реализовать. Речь идет о необходимости взаимодействия с динамическими графическими пользовательскими интерфейсами и их тестировании. Тестирование веб-приложений - сложный процесс. Такие инструменты, как Selenium и webdriver, являются ключевыми элементами в автоматизации тестирования содержимого веб-страницы и приложения в целом [6].

Анализ программных средств создания автоматизированных тестов [7] позволил выделить среди них инструмент под названием Playwright.

31 января 2020 года Microsoft выпустила свой инструмент автоматизированного тестирования Playwright с открытым кодом в Github на основе JavaScript [8].

Playwright, в отличие от других инструментов, работает практически со всеми современными браузерами включая Google Chrome, Mozilla Firefox, Microsoft Edge и Apple Safari.

Также Playwright поддерживает Headless режим для всех браузеров.

Headless режим представляет собой запуск в браузере автоматизированного теста на работоспособность каких-либо составляющих пользовательского интерфейса (UI - user interface) без отображения этого самого UI. Такое тестирование запускает скрипты в браузере, но не запускает и не показывает сам по себе UI.

Преимущество использования Headless режима заключается в том, что он использует настоящий

браузер, но намного быстрее запускает ваши тесты, чем, например, Selenium.

Автоматизированное тестирование программного обеспечения предоставляет большое преимущество перед ручным тестированием. Но для написания автоматизированных тестов нужно иметь достаточно хорошие знания в программировании, в фреймворках и в паттернах программирования. Обычно у инженеров по тестированию программного обеспечения такие знания и умения отсутствуют. Для решения данной проблемы существуют такие виды инструментов автоматизированного тестирования, называемые record and playback. Эти инструменты позволяют запоминать действия пользователя в браузере, далее переводить эти команды в программный код и в последующем воспроизводить их.

Рассмотрим, как это работает. Любая веб-страница состоит из html элементов. Веб-элемент – это индивидуальная сущность, генерирующаяся на веб-странице. Элементы – это все то, что пользователь видит (а иногда и не видит) на странице – заголовки, кнопки "ОК", поля ввода, текстовые блоки... Элементы в HTML определяются через имя тэга, атрибуты и содержание. У них также могут быть дочерние элементы – например, таблицы. CSS (формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки) может применяться к элементам и менять их цвета, размеры и расположение.

При запуске данного инструмента открывается браузер, и пользователь в этом браузере совершает действия, описанные в тестовом сценарии. Для каждого веб элемента в странице, с которым пользователь взаимодействовал, инструмент составляет локатор.

Локатор веб-элемента – это объект, который находит и возвращает веб-элементы на странице по заданному запросу. Инструмент автоматизированного тестирования запоминает все локаторы и действия, совершенные над ним такие, как клик по элементу, ввод в поле элемента и другие.

Далее эти действия и локаторы сохраняются в виде команд в программном коде. В итоге получается автоматизированный тест, который в точности будет повторять все действия, совершенные пользователем.

Существуют несколько подобных инструментов, например, такие как Selenium IDE, Katalon Studio, TestComplete и другие.

Наряду с преимуществами каждый из перечисленных инструментов имеет свои недостатки. Например, Selenium IDE работает только на браузере Mozilla Firefox, новые версии которого не поддерживаются. Следовательно, записывать автоматизированные тесты на новых версиях браузера не получится.

Katalon Studio работает на браузерах Google Chrome, Mozilla Firefox, но не работает на Microsoft Edge и Apple Safari.

Для работы с TestComplete необходимо приобрести лицензию. Также работает только на Google Chrome, Mozilla Firefox.

Исходя из вышеуказанных сведений о существующих инструментах, было принято решение разработать собственный инструмент автоматизированного тестирования на основе фреймворка Playwright, который позволит работать со всеми современными браузерами: Google Chrome, Mozilla Firefox, Microsoft Edge и Apple Safari.

На рисунке 1 показан интерфейс инструмента.

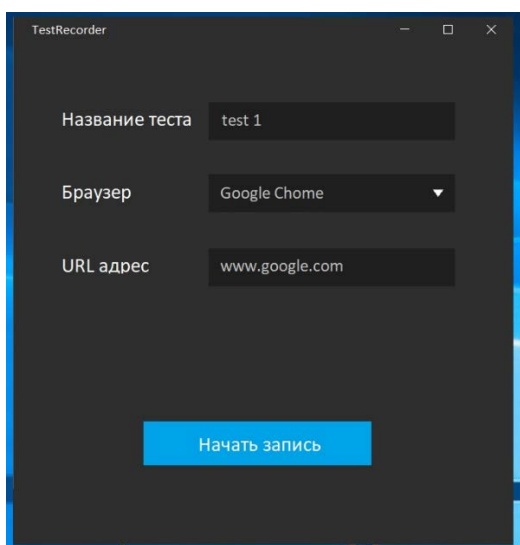


Рис 1. Интерфейс инструмента

3. Заключение

Разрабатываемый инструмент автоматизированного тестирования реализует все преимущества данного фреймворка. В итоге у инженеров тестировщиков появляется инструмент, с которым легко автоматизировать тестирование приложений без знаний в области программирования.

Список используемых источников

1. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. — Минск: Четыре четверти, 2017. — 312 с.
2. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: Пер. с англ. / Сэм Канер, Джек Фолк, Енг Кек Нгуенов. — К.: Издательство «ДиаСофт», 2001. — 544с.
3. Статья про автоматизацию тестирования [Электронный ресурс]. URL: <http://www.protesting.ru/automation/functional/whytoauto.html> (дата обращения: 15.02.2021)
4. Статья про целесообразность внедрения автоматизированного тестирования [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/otsenka-tselesoobraznosti-vnedreniya-avtomatizirovannogo-testirovaniya/viewer> (дата обращения 20.02.2021)
5. Что такое непрерывное развертывание (CD)? [Электронный ресурс]. URL: <https://www.jetbrains.com/ru-ru/teamcity/ci-cd-guide/continuous-deployment/> (дата обращения 20.02.2021)
6. Официальный сайт инструмента Selenium [Электронный ресурс]. URL: <http://seleniumhq.org/> (дата обращения: 26.02.2021).
7. Повышение эффективности процесса тестирования веб-приложений за счет создания инструмента для разработки автоматизированных тестов / Д.В. Набиева, Г.Н. Верхотурова // VI Международная научно-практическая конференция (школы-семинара) молодых ученых «Прикладная математика и информатика: современные исследования в области естественных и технических наук», Россия, Тольятти, 2020, С. 896-899.
8. Официальный сайт инструмента Playwright [Электронный ресурс]. URL: <https://playwright.dev> (дата обращений 28.02.2021).