ФРЕЙМОВРК АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Мищенко А. В., Мельников А. В.

Пискун Г.А. – канд. техн. наук, доцент

В настоящее время возрастает потребность внедрения автоматизации тестирования на крупных проектах по разработке программного обеспечения, так как это в первую очередь позволяет снизить затраты на обеспечение качества программного обеспечения путем сокращения затрат на ручное тестирование. С увеличением потребности в автоматизации, часто стала возникать проблема затрат ресурсов на разработку фреймворка автоматизации, а не на тестирование, в связи с этим в данной статье рассматривается подход, который позволит сократить время на второстепенные операции и больше времени уделять непосредственно тестированию.

В процессе разработки программного обеспечения, на каждом из этапов, начиная с составления требований, в программы неоднократно вносятся изменения, обусловленные как необходимостью исправления существующих ошибок, так и исправления выявленных в процессе их исполнения дефектов, или же желанием внести в программу дополнительные изменения. Контроль над изменениями приобрел статус критического фактора для сохранения актуальности программ [1].

При разработке программного обеспечения часто встречаются ситуации, когда новая версия продукта выходит раз в одну-две недели. Чем чаще выходит обновление продукта, тем чаще необходимо проводить тестирование его качество. Зачастую тестирование сайта занимает 2-4 часа, при этом определенное время занимает исправление дефектов и выпуск патча, который в свою очередь также требует проведения тестирования. Существовавший стандартный процесс разработки программного обеспечения подвергся некоторой модернизации. Ввиду того, что поведение новой версии программы должно совпадать с поведением предыдущей версии, за исключением ситуаций, обусловленных внесением изменений, соответствующих новым требованиям к системе, регрессионные системные тесты можно рассматривать как частичные требования к новым версиям системы. В большинстве случаев вместо регрессионного тестирования для проверки того, что качество новой версии программы не ухудшилось, выполняется множество всех тестов, используемых на этапе системного и функционального тестирования продукта [1]. Одним из очевидных решений здесь является автоматизация процесса тестирования, помогающая ускорить создание продукта и улучшить его качество. Автоматизация тестирования зачастую позволяет сократить время на тестирование до 1 часа, а также проводить тестирование на регулярной основе, запуская автоматические тесты ночью и затем анализирую их результаты в течении короткого промежутка времени.

Одним из самых популярных и востребованных видов тестирования является тестирования веб-сайтов. В зависимости от требований и специфики сайта, могут требоваться различные виды тестирования:

– нагрузочное тестирование;

– тестирование локализации;

– функциональное тестирование

Наиболее востребованной и эффективной является автоматизация функционального тестирования. При автоматизации функционального тестирования наибольшую эффективность показало применение паттернов Page Object и Page Factory. Данные паттерны строятся по принципу, когда каждая страница сайта в коде автотеста описывается в отдельном классе или модуле. Подход в автоматизированном тестировании, с использованием Page Objects, заключается в том, что весь код низкоуровневой работы со страницей (например, набор текста и нажатия мышкой по элементам) выносится в отдельные классы. Теперь тесты не работают со страницей напрямую, вызывая низкоуровневые методы программы управления браузером, а используют более высокоуровневые операции, специфичные для каждой страницы [2]. Это сокращает количество строк кода в тестах, тем самым делая код более читабельным, понятным и надёжным.

Практика показывает, что в процессе автоматизации по данному принципу около 60% трудозатрат приходится на создание структуры и разделения страниц на Page Object, поиск и написание уникальных локаторов элементов и т.д. Таким образом, решив вопрос автоматической генерации Page Object и поиска локаторов, можно больше чем в два раза сократить затраты на автоматизацию тестирования. Было разработано решение, которое позволяет автоматически генерировать код и искать локаторы. На основе SWD Recordera [3], был разработан инструмент, позволяющий, с помощью встроенного в браузер Javascript кода, автоматически выбирать необходимые элементы на странице и, используя готовые шаблоны генерировать готовые Page Object классы.

Данная программа, позволяет не только найти необходимый локатор, но и оптимизировать его (сократить размер) и сгенерировать весь необходимый код для последующей декларации элемента в коде автотеста. Принцип работы программы устроен на том, что в браузер встраивается специально разработанный javascript код, который позволяет при выделении элемента по клику мышкой сохранить его локатор и в дальнейшем добавить найденный элемент в приложение. После заполнения поля имени и нажатии на кнопку Add Element – найденный элемент сохранится в дереве Page Object. По завершению работы над добавлением

элементов в Page Object, на вкладке Source Code можно сгенерировать Page Object-класс на языке программирования Java.

На рисунке 1 представлен внешний вид главного окна приложения.

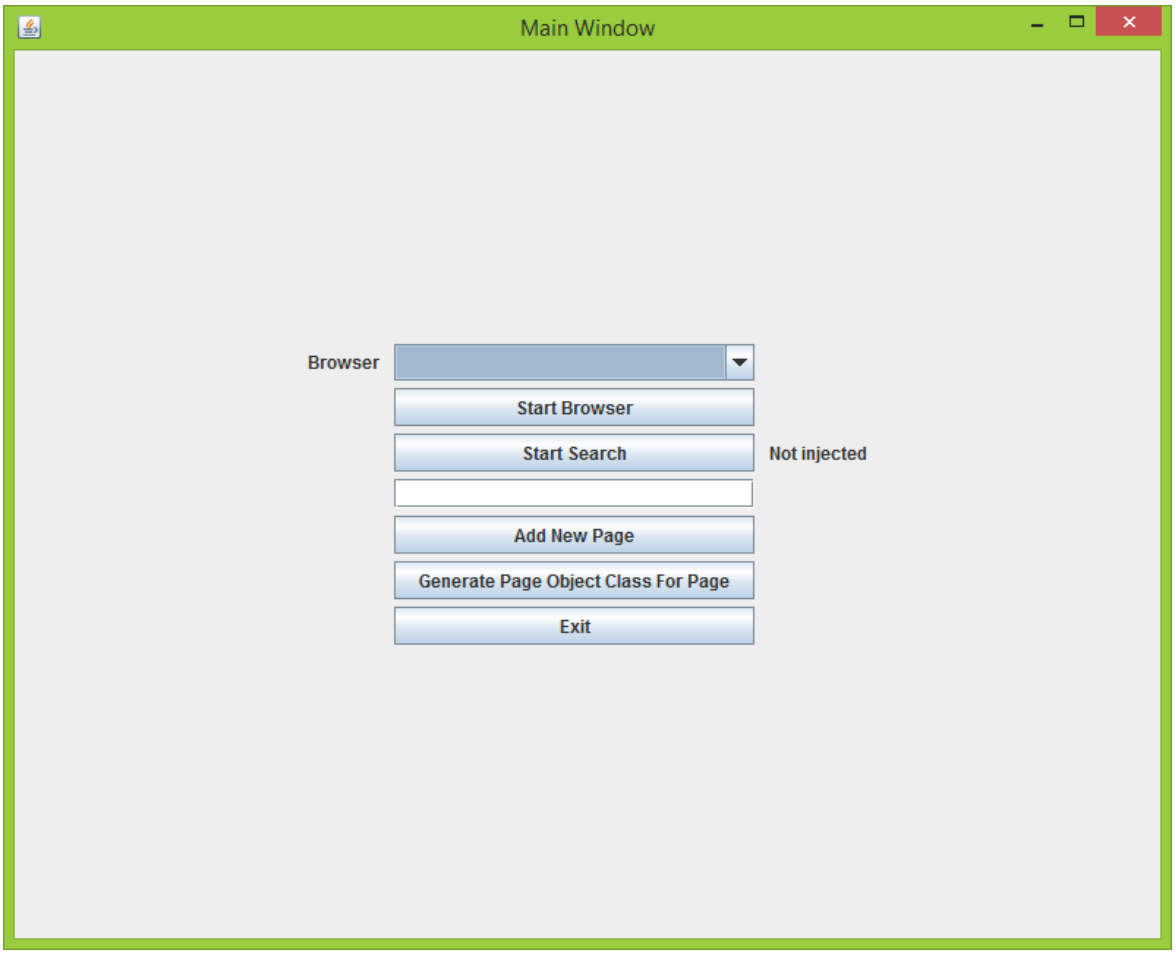

Рисунок 1 -.Главное окно программы

В исходный код приложения также была добавлена возможность добавлять собственные стратегии генерации локаторов, после чего для генерации можно выбрать любой из них, наиболее подходящий под конкретную специфику автотеста.

Данное приложение не может полностью генерировать код автотеста, и основная работа по-прежнему выполняется инженером по автоматизации, однако оно предоставляет возможность ускорить рутинную работу и сократить затраты на выполнение работ, непосредственно несвязанных с тестированием приложения.

Список использованных источников:

[1] Важность тестирования в разработке ПО [Электрон. ресурс] / Мой компьютер онлайн. – Электрон. дан. Режим доступа: https://mycompplus.ru/news/53-obzor-softa/2420-vazhnost-testirovaniya-v-razrabotke-po.html

[2] Использование паттерна Page Object. [Электрон. ресурс] / Selenium WebDriver. – Электрон. дан. Режим доступа: https://www.gitbook.com/book/kreisfahrer/selenium-webdriver

[3] Жарий Д. Быстрый старт автоматизации тестирования UI / Хабрахабр. – Электрон. дан. Режим доступа: https://habrahabr.ru/post/208822/