**И.О. Альтшулер, А.П. Кондратюк**

БрГУ имени А. С. Пушкина(г. Брест, Беларусь)

**Функциональное тестирование WEB-ПРИЛОЖЕНИЙ В ИНТЕГРИРОВАННОЙ СРЕДЕ SELENIUM IDE**

Тестирование занимает важное место в жизненном цикле программного обеспечения. Современный этап развития тестирования характеризуется глубокой интеграцией с процессом разработки, широким использованием автоматизации, большим набором технологий и инструментов. В данной статьерассматриваются возможности Selenium ID как средствафункционального автоматизированного тестирования web-приложений.Специалисты по тестированию обычно рекомендуют проводить в качестве базовых проверок web-приложений функциональное тестирование, тестирование графического интерфейса пользователя, совместимости, производительности и др. [1].

В частности, функциональное тестирование включает тесты, которые имитируют фактическое использование web-приложений и концентрируются только на выходных данных, полученных в ответ на пользовательский ввод и условия исполнения сценариев.Как известно, почти любой тип тестирования ПО может быть выполнен как вручную, так ис использованием средств автоматизации.Автоматизированное тестированиепредставляет собой набор техник, подходов и инструментальных средств, позволяющих исключить человека из выполнения отдельных задач в процессе тестирования. Автоматизированное функциональное тестирование web-прило-жений заключается в тестировании функциональных требований и особенностей приложения посредством инструментов для автоматизирован-ного тестирования. В ходе тестирования функциональности web-приложенийрекомендуетсяавтоматизировать предельно рутинные действия [2]. Автома-

тизация наиболее применима при тестировании валидационных форм, базовых операций, часто используемой функциональности приложений и др.

Одним из известных способов разработки автоматизированных тест-кейсов является тестирование под управлением ключевыми словами, в котором за пределы тест-кейса выносится набор входных данных и ожидаемых результатов, а также логика поведения тест-кейса, при этом данные и особенности поведения могут быть изменены без изменения кода [2].

Примером инструментального средства автоматизации тестирования, реализующего тестирование под управлением ключевыми словами, является Selenium IDE. Следует отметить, что все семейство Selenium – это фреймворк для тестирования web-приложений, включающий набор различных программныхинструментов: Selenium IDE, WebDriver,Selenium Grid.Selenium поддерживает автоматизацию всех основных браузеров с помощью WebDriver, который представляет собой API и протокол, определяющий не зависящий от языка интерфейс для управления поведением web-браузеров.Selenium Grid позволяет запускать тестовые примеры на разных машинах и разных платформах.Selenium IDEкак расширение для Chrome илиFirefox используется для записи, сохранения и воспроизведения сценариев тестирования web-страниц и приложений.При этомдействия пользователей записываютсяв браузере с использованием существующих команд Selenium с параметрами, определяемыми контекстом этого элемента[3].Язык написания скриптов в Selenium IDE **–**selenese (команды type, open, click, assert, verify и др.). С помощью Selenium IDE можно экспортировать тестовые случаи в различные форматы, а также в WebDriver.

В нашем проекте Selenium IDE используется для тестирования web-приложений в Firefox.Приведем пример автоматизированного теста для формы авторизациина сайте Национальной библиотеки Беларуси.Наиболее распространенным методом создания тестовых сценариев в Selenium IDE является запись.Затемсценарий выполняетсяс помощью функции воспроизведения и при необходимости корректируется. Ниже приводится часть тест-кейса по авторизации на сайте с введением невалидных данных в поле «Логин» после экспорта его в WebDriver(с выборомJava JUnit).

@Test

public void invaledName()

{

driver.get("https://nlb.by/");

driver.manage().window().setSize(new Dimension(1600, 860));

driver.findElement(By.cssSelector(".header-logo-phone> .col-xs-12 .svg-logo-animate")).click();

driver.findElement(By.cssSelector(".hide-1280")).click();

driver.findElement(By.name("USER\_PASSWORD")).sendKeys("i12345");

driver.findElement(By.name("USER\_LOGIN")).sendKeys("InvalidN");

driver.findElement(By.cssSelector(".button\_wide")).click();

driver.findElement(By.name("USER\_LOGIN")).click();

}

СПИСОК ЛИТЕРАТУРЫ

1. Вишневская, Т.И. Тестирование программного обеспечения как учебная дисциплина / Т.И. Вишневская // Образовательные ресурсы и технологии. – 2014. – № 1(4). – С. 83–89.

2. Куликов, С.C. Тестирование программного обеспечения. Базовый курс / С.С. Куликов. – Минск: Четыре четверти, 2017. – 312 с.

3. Selenium IDE. Автоматизация тестирования записи и воспроизведения с открытым исходным кодом для Интернета [Электронный ресурс].– Режим доступа: <https://www.selenium.dev/selenium-ide/>. – Дата доступа: 10.10.2020.