

Задаваясь (по рисунку 2, на уровне  $p_{\tilde{f}} \geq 10^{-7}$ ) интервалом  $[R_{min}, E_{max}] = [3,6 \text{ МН}, 5,9 \text{ МН}]$ , получаем:

$$P_{\tilde{f}} = 2/3(5,9 - 3,6) \times 1,28 \times 10^{-6} = 1,96 \times 10^{-6},$$

а именно – совпадающую с результатами численного моделирования величину полной вероятности отказа анализируемой конструкции.

Значения  $E_{max}$  и  $R_{min}$  можно вычислить из трансцендентного уравнения, вытекающего из  $p_{\tilde{f}}(r) = C_{\tilde{E}}(r) \times f_{\tilde{R}}(r) = e^{a-br} \times \beta/\lambda^\beta \times (r-\xi)^{\beta-1}$ , вида  $p_{\tilde{f}}(R_{min}) = p_{\tilde{f}}(E_{max}) = 0,1 \text{ max } p_{\tilde{f}} = 1,18 \times 10^{-7}$  – путем его решения относительно  $r$ . [2]

В рассматриваемом примере такое уравнение будет иметь два решения:

$$r_1 = R_{min} = 3,66 \text{ МН}$$

$$r_2 = E_{max} = 6,46 \text{ МН}.$$

При этом  $P_{\tilde{f}} = 2,2 \times 10^{-6}$ , что на 12,25% превышает (т.е. отличается в сторону запаса надежности) значение, полученное в результате численного моделирования.

#### Список использованных источников

1. Шпете, Г. Надежность несущих строительных конструкций [Текст]: Перевод с нем. О.О.Андреева. – М., Стройиздат, 1994. – 288с.
2. S.S.Derechennik, V.V.Tur. Adaptive Estimation of the In Situ Characteristic Concrete Strength // Solid State Phenomena. – 2019 (292). – P. 257-263.

УДК 004.054

## ПРЕИМУЩЕСТВА ФРЕЙМВОРКА PLAYWRIGHT ДЛЯ ФУНКЦИОНАЛЬНОГО ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ

Д.С. Жукович

Брестский государственный технический университет, Брест, Беларусь,  
zhukovich14@gmail.com

*This article is about automated software testing. Key features of Playwright framework for end-to-end automated testing of web applications are described. This framework helps to solve the problem of cross-browser testing. It also gives an opportunity for developers to select the most suitable programming language from four possible ones and write automated tests on any computer platform (Windows, Linux, macOS) for any platform including mobile (Android, iOS).*

Сегодня невозможно представить нашу жизнь без различных онлайн-сервисов и приложений. Зачастую они направлены на повышение эффективности работы компании, автоматизацию процессов, обработку больших объемов данных и многое другое. Каждый день мы пользуемся веб-приложениями, которые открываем в браузере. И каждое такое приложение необходимо проверять на соответствие определенным требованиям к его функционированию, взаимодействию с пользователем.

Перед командой, которая работает над созданием или поддержкой программного продукта, ставится вопрос о необходимости тестирования приложения. В процессе разработки программного продукта создается множество различного функционала, который должен работать стабильно. Чем объемнее продукт, тем больше ресурсов необходимо для тестирования.

Тестирование программного обеспечения – процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта [1]. Во время разработки программного продукта тестировщики пишут тест-сценарии, которые проходят вручную на этапе тестирования созданного функционала. Далее необходимо поддерживать работоспособность существующего функционала, осуществляя проверку по созданным ранее тестовым сценариям. Для каждого продукта существует свой набор тестов, покрывающий базовый функционал позитивными сценариями. Такие тесты необходимо проводить каждый раз при доставке заказчику нового функционала, а также на этапе тестирования вновь разработанного функционала.

Актуальным на сегодняшний день является кросс-браузерность веб-приложений и единообразие пользовательского интерфейса на устройствах с различными форм-факторами. К числу основных веб-браузеров можно отнести: Chromium (Chrome), Microsoft Edge, Safari, Firefox. Объем работы для ручного тестирования кратно увеличивается из-за необходимости проверки одного и того же функционала для каждого такого веб-браузера. Автоматизированное кросс-браузерное тестирование позволит существенно сократить время проверки веб-приложения и сэкономит ресурсы. А статистика успешно пройденных и «упавших» тестов поможет определить, что именно не работает.

Веб-приложение, как правило, состоит из двух модулей: серверная часть и пользовательский интерфейс. Для серверной части необходимо тестирование API веб-сервисов, которое может в себя включать и тестирование базы данных; для пользовательского интерфейса – тестирование внешней части приложения, интеграции с серверной частью.

Автоматизированные тесты включают в себя как проверки отдельных функциональных частей приложения, так и реальные пользовательские сценарии – сквозные тесты (end-to-end), которые управляют браузером и имитируют действия пользователя. Такие сценарии должны быть проверены с каждой новой версией продукта [2].

Playwright является на сегодняшний день наиболее подходящей бесплатной технологией, которая доступна в четырех наиболее популярных языках программирования (JavaScript, Java, C#, Python). Она предлагает огромный спектр методов, позволяющих проводить кросс-браузерное автоматизированное тестирование веб-приложений. Playwright поддерживает все современные механизмы рендеринга, включая Chromium, WebKit и Firefox. К тому же поддерживается нативная мобильная эмуляция Google Chrome для Android и Mobile Safari для IOS. Один и тот же механизм рендеринга работает и на рабочем столе, и в облаке [3].

Playwright предлагает набор для тестирования как серверной части, так и пользовательского интерфейса, тем самым нет необходимости подключать для этих целей отдельные библиотеки. Данный фреймворк кроссплатформенный, что позволяет использовать его на Windows, Linux или macOS, локально или в CI (continuous integration).

Использование протокола DevTools делает Playwright более эффективным, поддержка опции headless позволяет настроить запуск теста в окне браузера или же без его явного открытия [2].

Playwright имеет повышенную точность тестирования за счет функции автоматического ожидания, которая повторяет соответствующие проверки элементов и запрошенных действий до тех пор, пока все они не станут пригодны-

ми для действий. У данного фреймворка также есть встроенные функции, называемые трассировками (Traces), которые могут делать автоматические снимки экрана, видеозапись теста, повторную попытку теста и механизм ведения журнала. Playwright предоставляет инструмент инспектора, который помогает отслеживать и отлаживать каждый шаг исполнения, видеть точки кликов и проверять локаторы страниц на лету (во время выполнения теста) [4].

Playwright предоставляет также функцию контекста браузера, которая помогает сохранять и передавать состояние браузера в любой другой новый браузер. Имеется возможность воспользоваться функцией генерации кода, которая записывает все шаги в веб-приложении и далее преобразует в написанные тесты на любом поддерживаемом языке программирования [4].

Сегодня фреймворк Playwright для автоматизированного тестирования от Microsoft набирает популярность и активно развивается. Благодаря всем вышеперечисленным основным функциям и возможностям, Playwright является основным конкурентом других инструментов автоматизации, например, Selenium и Cypress.

Очевидно, что выбор технологии Playwright для основы фреймворка автоматизации тестирования является обоснованным. Используя набор выбранных технологий планируется создать эффективный универсальный фреймворк для автоматизированного кросс-браузерного тестирования веб-сайтов. Готовых решений не существует, т.к. каждый программный продукт индивидуален. Поэтому для успешного создания фреймворка автоматизированного тестирования необходимо учитывать логику работы, набор API веб-сервисов, а также реализацию внешнего интерфейса, что позволит ускорить процесс непрерывного тестирования веб-приложения.

#### **Список использованных источников**

1. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. – 3-е изд. – Минск: Четыре четверти, 2020. – 312 с..
2. Playwright – драматургия от Microsoft и новый инструмент для тестирования [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/jugru/blog/487294> – Дата доступа: 15.10.2022.
3. Playwright [Электронный ресурс]. – Режим доступа: <https://playwright.dev> – Дата доступа: 15.10.2022.
4. Playwright JS Automation Testing from Scratch with Framework [Электронный ресурс]. – Режим доступа: <https://www.classcentral.com/course/udemy-playwright-tutorials-automation-testing-98095> – Дата доступа: 15.10.2022.

УДК 655.1

## **ОБНАРУЖЕНИЕ АНОМАЛИЙ В ДАННЫХ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА**

А.Л. Михняев, А.В. Пархоць, И.В. Абоимов, Д.А. Дворанинович  
Брестский государственный технический университет, Брест, Беларусь,  
[bstucon1@bstu.by](mailto:bstucon1@bstu.by)

*The main task of this work is the detection of anomalies for their further analysis, classification and possible ways of elimination.*