УДК 004.415.532.3

Р. В. Слободян
І. В. Богач

АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ ВЕБ-ДОДАТКІВ ПЕРЕД ЇХ РОЗГОРТАННЯМ

Вінницький національний технічний університет

Анотація

В роботі проаналізовано існуючі методи та інструменти для автоматизованого тестування веб-додатків. Удосконалено методи тестування компонентів веб-додатків під час процесу їх розгортання. Розроблено алгоритм та програмний засіб для реалізації.

Ключові слова: тестування, автоматизація, розгортання, веб-додаток.

Abstract

The report analyzes existing methods and instruments for automated testing of web applications. Methods for components of web applications testing during their deployment process have been improved. An algorithm and a software tool for implementation have been developed.

Keywords: testing, automatization, deployment, web-application.

Вступ

Стрімкий розвиток та удосконалення технологій створення веб-додатків, спричинений зростаючими потребами та вибагливістю користувачів, посилив необхідність у автоматизованому тестуванні програмного продукту не тільки на ключових етапах життєвого циклу додатку, а й і протягом усього процесу, включаючи рутинні розгортання проміжних версій.

Необхідність підтвердження стабільності роботи додатку з урахуванням змін, що були внесені під час роботи команди розробки за максимально короткий термін є надзвичайно високою для нинішнього IT-ринку. Тому, обрана тема є актуальною на сьогоднішній день, так як налагоджений процес автоматизованого тестування у цілому, та веб-додатків зокрема, протягом усіх етапів життєвого циклу продукту є запорукою успішності окремого проекту, та одним із ключових факторів успіху та конкурентоздатності організації в цілому.

Результати дослідження

В доповіді здійснено аналіз існуючих засобів для автоматизованого тестування веб-додатків. Поміж них виділено ті, котрі можуть бути застосовані для більшості проектів без необхідності обмеження команди розробки до певних інструментів, мови програмування, операційної систем, бібліотеки, тощо, як от Selenium, Ranorex, Jest, Mocha [1-14].

Проведено дослідження можливостей використання вище згаданих засобів для автоматизації тестування як для компонентів користувацького інтерфейсу, так і їх контролерів під час різних фаз розробки та впровадження веб-додатку. Під час дослідження було виділено ті, котрі можуть бути застосовані для тестування серверної складової веб додатку лише за певних умов, як от використання Node.js для її реалізації; запропоновано альтернативні рішення для таких мов програмування, як Java та C# [15-20].

Розглянуто сучасні методи тестування веб-додатків, включаючи ті, які можна та необхідно застосовувати за відсутності можливості локального розгортання, виокремлено Jest, Mocha та Selenium як найбільш гнучкі та ефективні для тестування сучасних веб-компонентів [21-24].

Проаналізовано методики створення тестів окремих модулів за допомогою Jest та Mocha, їх відтворення, збору та аналізу отриманих результатів.

Досліджено можливості інтеграції автоматизованих тестів компонентів інтерфейсу користувача, реалізованих за допомогою Jest та Mocha, у системи контролю версій, хостингу та автоматизованого розгортання, серед яких виділено Jenkins як найбільш гнучку. Виділено можливість нативної інтеграції тестів, що реалізовані за допомогою Jest у проекти, що використовують NPM, за рахунок використання хук-ефекту під час безпосередньої фіксації внесених змін у системі контролю версій, що базується на Git [25-27].

На основі проаналізованої інформації було удосконалено метод тестування веб-додатків шляхом впровадження локального тестування елементів інтерфейсу користувача використовуючи тести, реалізовані за допомогою Jest під час хук-ефекту безпосередньої фіксації внесених змін у системі контролю версій, що базується на Git.

Розроблено алгоритм та програмний засіб для подальшої реалізації використовуючи Jest як інструмент для тестування елементів інтерфейсу користувача та Jenkins у якості сервісу для автоматизації тестування та розгортання.

Висновки

Проаналізовано існуючі методи автоматизованого тестування веб-додатків. Удосконалено метод роботи як з елементами користувацького інтерфейсу, які з’являються у результаті асинхронної клієнт-серверної взаємодії, так і їх серверною частиною.

Розроблено алгоритм та програмний засіб для його реалізації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. What is Selenium? [Електронний ресурс] – Режим доступу: http://docs.seleniumhq.org/
2. Burns D. Selenium 2.0: Beginner`s Guide / D. Burns – London : Packt Publishing, 2012. – 232 с.
3. Best Practices for Coded UI Tests [Електронний ресурс] – Режим доступу: https://learn.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2017/test/best-practices-for-coded-ui-tests
4. Introduction to QTP [Електронний ресурс] – Режим доступу: https://goo.gl/Z6dRff
5. Ranorex – Automated Testing Tool for Desktop, Web & Mobile Applications [Електронний ресурс] – Режим доступу: https://goo.gl/uF6TmU
6. Ranorex – Test Automation [Електронний ресурс] – Режим доступу: https://goo.gl/bc9q4e
7. Jest [Електронний ресурс] – Режим доступу: https://jestjs.io/uk/
8. Початок роботи з Jest [Електронний ресурс] – Режим доступу: https://jestjs.io/uk/docs/getting-started
9. Тестування фреймворків використовуючи Jest [Електронний ресурс] – Режим доступу: https://jestjs.io/uk/docs/testing-frameworks
10. Karma [Електронний ресурс] – Режим доступу: https://karma-runner.github.io/latest/index.html
11. Karma – How It Works [Електронний ресурс] – Режим доступу: https://karma-runner.github.io/6.4/intro/how-it-works.html
12. Karma – Public Api [Електронний ресурс] – Режим доступу: https://karma-runner.github.io/6.4/dev/public-api.html
13. Mocha [Електронний ресурс] – Режим доступу: https://mochajs.org/
14. Mocha – Asynchronous code [Електронний ресурс] – Режим доступу: https://mochajs.org/#asynchronous-code
15. Jenkins [Електронний ресурс] – Режим доступу: https://www.jenkins.io/
16. Jenkins Pipeline [Електронний ресурс] – Режим доступу: https://www.jenkins.io/doc/book/pipeline/
17. Bitbucket [Електронний ресурс] – Режим доступу: https://bitbucket.org/
18. Bitbucket Pipelines [Електронний ресурс] – Режим доступу: https://bitbucket.org/product/features/pipelines
19. GitHub [Електронний ресурс] – Режим доступу: https://github.com/
20. GitHub – Using workflows [Електронний ресурс] – Режим доступу: https://docs.github.com/en/actions/using-workflows
21. Cloud-based vs. Web-based Applications: What You Need to Know [Електронний ресурс] – Режим доступу: https://www.linkedin.com/pulse/cloud-based-vs-web-based-applications-what-you-need-know-john-tomblin/
22. Difference Between Cloud-Based App v/s Web-Based App: Full Comparison [Електронний ресурс] – Режим доступу: https://www.wscubetech.com/blog/cloud-based-application-vs-web-based-application-which-one-is-better/
23. What is multi-tenancy? [Електронний ресурс] – Режим доступу: https://www.techtarget.com/whatis/definition/multi-tenancy

24. Platform Multitenant Architecture [Електронний ресурс] – Режим доступу: https://architect.salesforce.com/fundamentals/platform-multitenant-architecture
25. Node.js [Електронний ресурс] – Режим доступу: https://nodejs.org/uk/
26. Improving NodeJS workflow with GIT hooks [Електронний ресурс] – Режим доступу: https://medium.com/@satya164/improving-nodejs-workflow-with-git-hooks-40996830619f
27. Node.js – Pre-commit hooks [Електронний ресурс] – Режим доступу: https://www.npmjs.com/package/pre-commit

Слободян Роман Віталійович – *старший інженер з розробки програмного забезпечення, керівник команди з розробки, ФОП, м. Вінниця, e-mail: romich.prof@gmail.com.*

Богач Ілона Віталіївна *– к.т.н., доцент кафедри Автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м.Вінниця, e-mail: ilona.bogach@gmail.com.*

Slobodian Roman V. – *senior software engineer, development team lead, PE, Vinnytsia, e-mail: romich.prof@gmail.com.*

Bogach Ilona Vitaliivna - *Associate Professor of Automation and Intelligent Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com.*