УКД 004.054


А.А. Краснов, Д.В. Кочкин
Вологодский государственный университет

РАЗРАБОТКА СИСТЕМЫ ТЕСТОВ ДЛЯ ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЯ

В статье представлены результаты исследования по выбору методов тестирования, применимых при создании системы тестов веб-приложения. Определены методы тестирования веб-приложения. Проведен анализ применимости методов и их порядок выполнения.

Методы тестирования, тестирование веб-приложения, инструменты тестирования.

В настоящее время для большинства современных организаций спрос на разработку веб-приложений стремительно растет. Это связано как с автоматизацией бизнес-процессов организации, так и с текущей политикой внедрения цифровой трансформации [1].

Разработанное веб-приложение должно соответствовать всем техническим и функциональным требованиям, быть удобным в использовании, соответствовать современным техническим стандартам и безошибочно выполнять требуемую бизнес-логику. Зачастую мы можем столкнуться с некачественно реализованным веб-приложением по причинам отсутствия этапа тестирования в жизненном цикле разработки программного обеспечения (ПО), несистематизированном процессе тестирования [2, 4].

Попытка игнорирования данного этапа с целью экономии бюджета и неправильно выстроенная методика проведения тестирования, как правило, оборачиваются высокой тратой временных и финансовых ресурсов [3, 5]. Кроме этого, нарушения в работе функционала, бизнес-логики веб-приложения, связано как с репутационными рисками для организации, так и с различными штрафными санкциями.

Разрабатываемые тесты предназначены для единой информационной системы в сфере закупок (ЕИС). Официальный сайт ЕИС в сфере закупок в информационно-телекоммуникационной сети Интернет (рис. 1) предназначен для обеспечения свободного и безвозмездного доступа к полной и достоверной информации о контрактной системе в сфере закупок и закупках товаров, работ, услуг отдельными видами юридических лиц, а также для формирования, обработки и хранения такой информации.

На текущее время существует множество методов тестирования программного обеспечения. Правильный выбор методов в подавляющем большинстве случаев определяет успешность проведения тестирования на проекте в целом [7, 8]. Методы тестирования веб-приложений представлены на рисунке 2.

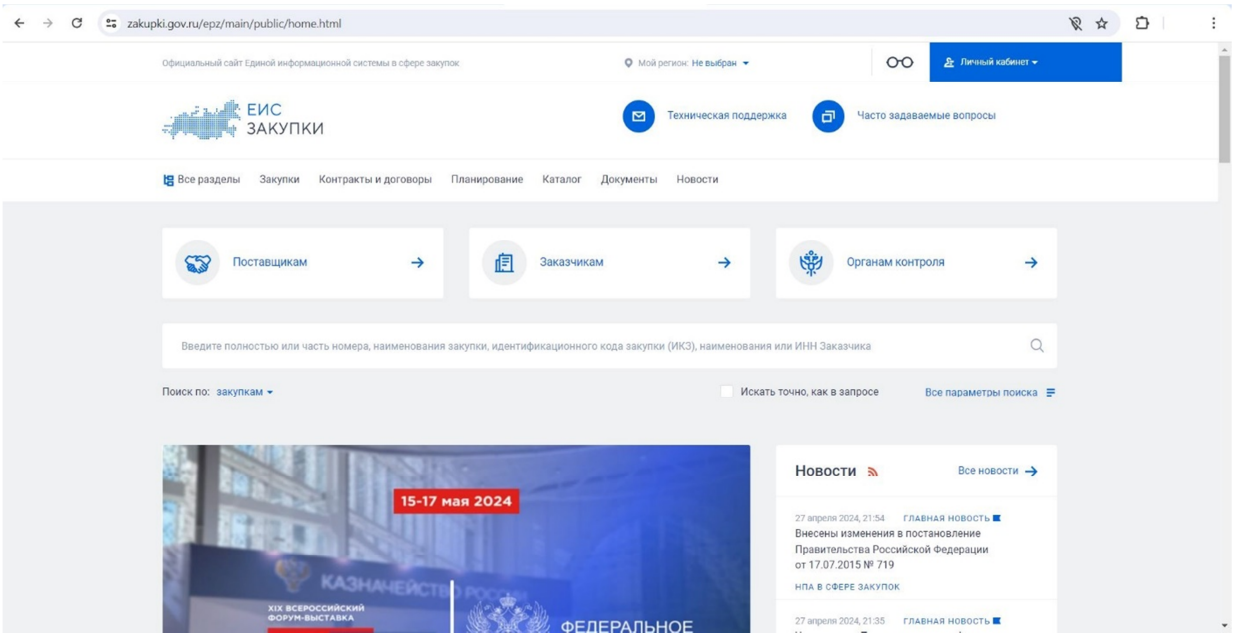

Рис. 1. Сайт ЕИС в сфере закупок

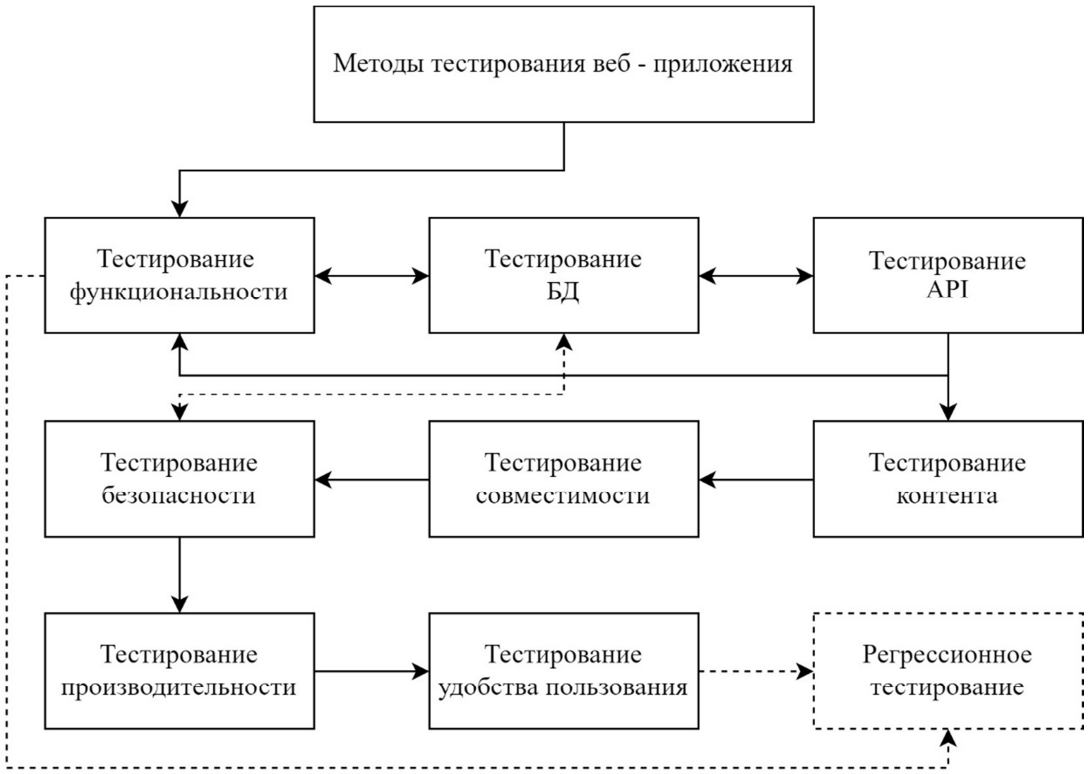

Рис 2. Методы тестирования веб-приложения

Исходя из представленной схемы видно, что тестирование функциональности, тестирование базы данных, тестирование API и тестирование контента связаны между собой – эти методы являются первоочередными в начальном этапе тестирования веб-приложения. Выполнение данных методов создаст уверенность, что весь реализованный функционал веб-приложения работоспособен, соответствует требованиям документации и графически корректно отображается на веб-странице.

Следующим шагом проводится тестирование совместимости. Тестирование совместимости создаст уверенность, что веб-приложение правильно загружается и корректно работает в требуемых браузерах, операционных системах и устройствах.

Далее проводится тестирование безопасности. Данный метод может использоваться совместно с тестированием базы данных. Тестирование безопасности обеспечивает гарантию защищенности веб-приложения. После того, как тестирование безопасности выполнено, проводится тестирование производительности. Оно обеспечивает стабильность, надежность и высокую производительность веб-приложения в соответствии с требованиями к системе.

Следующим шагом проводится тестирование удобства пользования. Тестирование удобства пользования позволит определить, насколько веб-сайт привлекателен, комфортен и полезен для пользователя.

Выполнение регрессионного тестирования даст уверенность, что исправления в системе или реализованный новый функционал не привел к наведенным дефектам. Регрессионное тестирование может быть необязательным в случае, если не ожидается внесение каких-либо изменений в систему веб-приложения.

Если были исправления в системе или реализован новый функционал, который не требует проверок со стороны совместимости, безопасности, производительности и удобства пользования, то после выполнения первоочередных методов тестирования перечисленные методы могут быть пропущены.

Далее рассмотрим более подробно представленные методы тестирования:

I. Тестирование функциональности – является базовым методом тестирования веб-приложения. Функциональное тестирование гарантирует, что реализованный функционал веб-приложения соответствует технической документации и требованиям заказчика.

Метод функционального тестирования содержит выполнение следующих этапов:

1. Модульное тестирование – тестирование отдельных функций, классов, методов классов, взаимодействие классов, небольших библиотек для демонстрации того, что программа выполняется согласно спецификации [12]. Проводится разработчиком при готовности кода какой-либо отдельной подсистемы/модуля.

Примеры инструментов модульного тестирования: JUnit, NUnit, PHPUnit и др. Данные инструменты обеспечивают возможность создания тестовых сценариев и автоматического выполнения тестов.

2. Компонентное тестирование – тестирование небольших отдельных подсистем/модулей веб-приложения для того, чтобы на ранних этапах разработки снизить вероятность появления более серьезных ошибок в дальнейшем. Проводится тестировщи-

ком, когда модульное тестирование выполнено, билд готов и успешно установлен на тестовый стенд. Необходимо убедиться, что функционал подсистемы/модуля работоспособен и полностью соответствует требованиям спецификаций.

Примеры инструментов компонентного тестирования представлены в таблице.

Таблица

Примеры инструментов компонентного тестирования

Чек-лист (тест-дизайн)	Документ, содержащий в себе список проверок
Техники тест дизайна	1. Классы эквивалентности и граничные значения 2. Попарное тестирование 3. Таблица принятия решений 4. Диаграмма переходов и состояний и др.
Баг-трекинговые системы управления проектами	1. Jira с плагином Zephyr 2. Testrail 3. Testlink и др.
СУБД	1. Oracle SQL 2. DBeaver 3. MySQL и др.

Рассматривая инструменты из таблицы, чек-лист является одним из видов тестовой документации, составляется тестировщиком в соответствии со структурой спецификации и применением указанных в таблице техник тест-дизайна. В ЕИС тест-дизайн составляется при помощи инструмента Google Таблицы и содержит в себе: краткое описание задачи; окружение; трудозатраты; сводная статистика по проверкам, дефектам; перечень выполняемых проверок; время на проверку; приоритет; результат; ID бага / Комментарии QA / Проверяющий; версия спецификации; реализация разработкой и комментарий.

Пример чек-листа (тест-дизайна) представлен на рисунке 3.

Баг-трекинговые системы позволяют управлять разработкой веб-приложения, оформлять дефекты, создавать тест-кейсы, коллекции с прогонами тест-кейсов определенных подсистем. В ЕИС используется Jira с плагином Zephyr, где проводится работа по управлению проектом, реализации доработок, дефектам и инцидентам. Осуществляется работа с тест-кейсами и тестовыми прогонами.

При помощи СУБД проверяется соответствие требованиям реализованных таблиц, их атрибутов и типов данных. Для ЕИС используются инструменты СУБД: Oracle SQL и DBeaver.

3. Интеграционное тестирование – тестирование взаимодействий отдельных подсистем/модулей веб-приложения между собой. Проводится при готовности нескольких взаимосвязанных подсистем/модулей. Необходимо убедиться, что подсистемы между собой работают корректно, без ошибок и данные успешно передаются из одной подсистемы в другую.

Инструменты интеграционного тестирования: интеграционные схемы; приложения: SoupUI, Postman; интеграционная шина; брокеры сообщений: Apache Kafka, RabbitMQ, ActiveMQ и др.

Интеграционные схемы позволяют составить интеграционный пакет, проверить валидность и соответствие элементов пакета интеграционным схемам. Инструменты SoupUI, Postman необходимы для проведения тестирования веб-сервисов и передачи интеграционных пакетов через протокол HTTP(S). Интеграционная шина является набором программных модулей, входящих в подсистему интеграции, реализующий функционал взаимодействия между подсистемами и внешними системами. Брокеры сообщений позволяют отслеживать отправку, маршрут и доставку сообщения от отправителя к получателю.

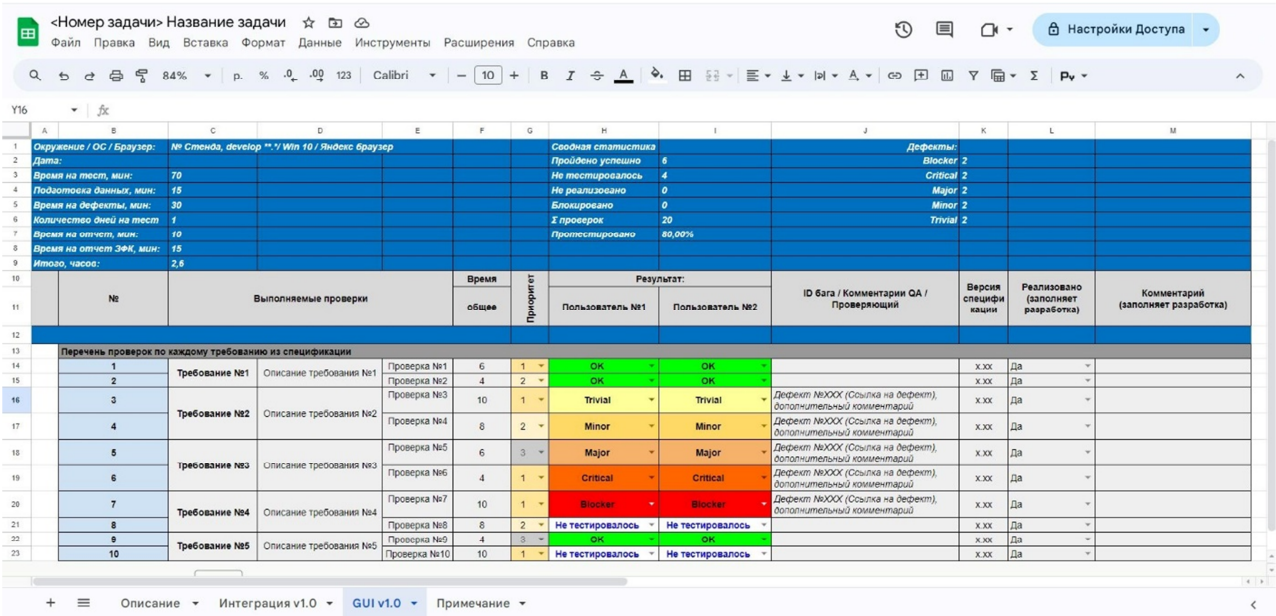

Рис. 3. Чек-лист (тест-дизайн)

В ЕИС используются инструменты тестирования, такие как: интеграционные схемы; приложения: SoupUI, Postman; интеграционная шина и брокер сообщений ActiveMQ.

4. Дымовое тестирование – тестирование критически важных функций, в случае неработоспособности которых дальнейшее тестирование проводить нецелесообразно. Выполняется после модульного, компонентного и интеграционного тестирования, когда уже готова первая версия приложения или выполнено обновление приложения. Проводится посредством прогонов тест-кейсов дымового уровня, когда тест-кейсы написаны только на основной бизнес-процесс веб-приложения.

Инструменты дымового тестирования: багтрекинговые системы управления проектами, которые указаны в таблице.

В ЕИС дымовое тестирование применяется после стабилизации версии (отладки системы) и при выпуске хотфикса.

II. Тестирование БД – тестирование баз данных гарантирует, что созданные таблицы с атрибутами, их связи между собой, соответствуют требованиям. Значения корректно сохраняются в БД и успешно производится репликация данных. Этот метод осуществляется совместно с компонентным и интеграционным тестированием.

Примеры инструментов тестирования БД перечислены в таблице.

III. Тестирование API – тестирование набора правил, с помощью которых веб-приложения обмениваются/общаются между собой.

Этапы тестирования API: тестируются методы API; тестируется взаимодействие API с другим API; проверяется доступ к API при авторизации/аутентификации; проверяется обработка ошибок при некорректных данных; тестируется производительность API и безопасность.

Данный метод проводится совместно с тестированием функциональности и тестированием БД.

Инструменты тестирования API: Postman, SoupUI и др. SoupUI позволяет осуществить взаимодействие с веб-приложением по SOAP-протоколу. Postman позволяет осуществить взаимодействие с веб-приложением как по SOAP-протоколу, так и по REST-архитектурному стилю.

В ЕИС используется SOAP протокол и инструменты: SoupUI, Postman.

IV. Тестирование контента – процесс проверки текстов, изображений, видео, аудио и других мультимедийных элементов на веб-странице. Данный метод проводится совместно с тестированием функциональности.

Инструменты тестирования контента: Grammarly, W3C Markup Validation Service, ImagesLoaded, DevTools и др. Данные инструменты позволяют проверить корректность и валидность HTML-кода веб-страницы, CSS стилей внешнего вида веб-страницы. С помощью DevTools в ЕИС осуществляется просмотр параметров элемента веб-страницы (шрифт, размер, цвет и т.д.) и сравнение параметров элемента с макетами спецификаций.

V. Тестирование совместимости – тестирование на предмет проверки корректной работы веб-приложения в различных средах, таких как операционные системы, браузеры, устройства или сетевые условия.

Этапы тестирования совместимости: определить, на каких операционных системах, браузерах и устройствах должно работать веб-приложение; осуществить выбор подходящего инструмента тестирования совместимости; провести тестирование; выявить и осуществить валидацию дефектов.

Инструменты тестирования совместимости: DevTools, BrowserStack, CrossBrowserTesting, LambdaTest и др. Данные инструменты позволяют тестировать веб-приложение на различных браузерах и устройствах без необходимости наличия всех этих конфигураций физически.

В ЕИС тестирование совместимости проводится на веб-браузерах Яндекс.Браузер и Chromium GOST, операционных системах Windows, Linux, Android, iOS.

По итогу проведения функционального тестирования, БД, API, контента, совместимости формируется отчет. Он содержит в себе: общую информацию о тестировании (окружение, стенд, дата проведения тестирования, тестировщики); тестовое покрытие; информация по проблемам (качество в %, актуальные дефекты); все найденные дефекты и уточнения; проблемы при тестировании.

VI. Тестирование безопасности – важный этап тестирования веб-приложения. Необходимо убедиться, что веб-приложение выдерживает попытки взлома, несанкционированного доступа к информации.

Этапы тестирования безопасности: проводится сканирование уязвимостей на наличие известных сигнатур уязвимостей специальным инструментом ПО; выявляются слабые стороны сети и системы; имитируется атака на веб-приложение; выявляются и валидируются дефекты; повторно имитируется атака на веб-приложение; оцениваются риски, проводится аудит безопасности и оценка состояния системы; формируется отчет о проведенном тестировании.

Тестирование безопасности может проводится совместно с тестированием БД.

Инструменты тестирования безопасности: Burp Suite, Metasploit, Wireshark и др. Данные инструменты позволяют сканировать уязвимости веб-приложения, произвести взлом сессии, перехват и анализ трафика, модификацию запросов.

VII. Тестирование производительности – тестирование на предмет оценки быстроты работы веб-приложения под определенной нагрузкой. Воздействие производится на веб-сервер, сервер БД, балансировщик. Метод включает в себя: нагрузочное, стресс, тестирование стабильности и масштабируемости.

Этапы тестирования производительности представлены на рисунке 4. Основываясь на схеме этапов

тестирования в первую очередь необходимо выявить требования по производительности к веб-приложению. На этом этапе определяется, что именно необходимо проверить в процессе тестирования, например: максимальную нагрузку или время отклика. Проводится анализ инструментов, наиболее применимых к системе. Следующим шагом подготавливаются тестовые сценарии и настраивается тестовая среда в соответствии с выбранным инструментом тестирования [6, 9]. Далее производится запуск тестов и анализ полученных результатов.

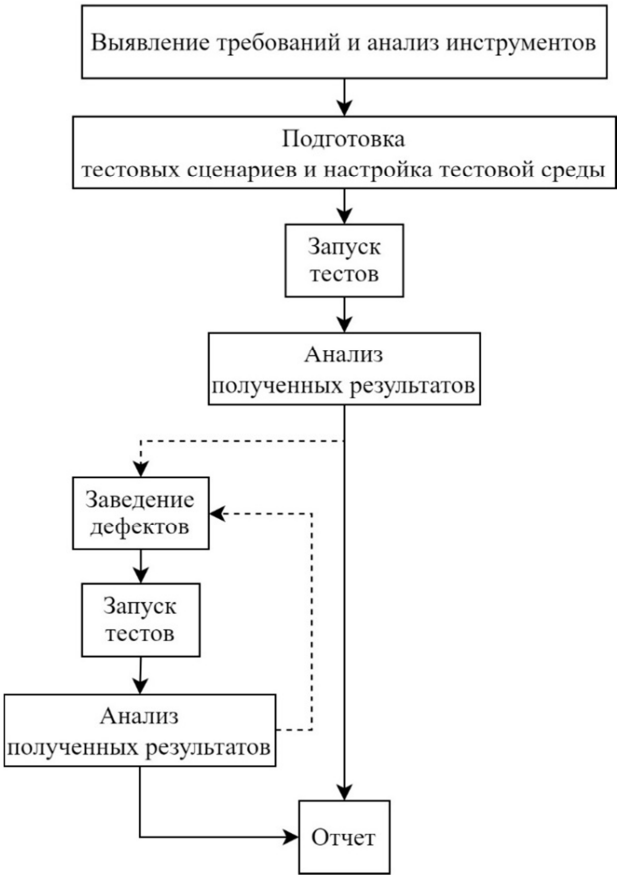

Рис. 4. Этапы тестирования производительности

В случае, если были обнаружены ошибки во время тестирования, то необходимо завести дефекты и после их исправлений, повторно запускаются тесты. Анализируются результаты до тех пор, пока все дефекты не будут исправлены и в полученных результатах отсутствуют ошибки. Если при анализе полученных результатов все тесты прошли успешно, то выполняется переход к заключительному этапу – написанию отчета о проведенном тестировании.

Инструменты тестирования производительности: JMeter, LoadRunner, Gatling и др. Данные инструменты позволяют смоделировать доступ к веб-приложению определенного количества пользователей.

VIII. Тестирование удобства пользования – это установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей веб-приложения в контексте заданных условий.

Этапы тестирования удобства пользования: необходимо определить, с помощью какого сервиса будет проводится тестирование; необходимо разместить проект на выбранный сервис и запустить тестирование; по представленному отчету от сервиса, анализируются результаты тестирования, документируются выявленные недостатки и предоставляются проектной команде на дальнейший анализ.

Сервисы тестирования удобства пользования: Usabilitylab, Точка качества, UsabilityHub, Hotjar, UserTesting. Данные сервисы позволяют провести тестирование интерфейса веб-приложения на представителях целевой аудитории для оценки удобства пользования.

IX. Регрессионное тестирование – это проверка ранее реализованного функционала при внесении в него изменений. В случае, если были внесены изменения в систему, то необходимо проверить, что функционал по-прежнему работоспособен, отсутствуют какие-либо наведенные ошибки и сбои. Тестирование производится ручным или автоматизированным методом по тест-кейсам дымового уровня. По окончании тестирования составляется отчет.

Инструменты регрессионного тестирования: багтрекинговые системы управления проектом, TestComplete, Selenium, Soap UI, Postman, Appium, RFT, QTP/UFT, UI Automator. Данные инструменты позволяют создать коллекции ручных или автоматизированных тестов GUI и API, запуск автотестов и отчет о проведении тестирования.

Для ЕИС в сфере закупок регрессионное тестирование проводится во время стабилизации версии – отладки системы. Создаются тестовые прогоны по подсистемам в Jira Zephyr. Прохождение тест-кейсов осуществляется как ручным, так и автоматизированным способом. По окончании тестирования составляется отчет, где указывается: общее качество системы; статус и качество всех задач, выпущенных за версию; перечень актуальных дефектов и уточнений; полный перечень всех заведенных дефектов и уточнение; статистика по дефектам и уточнениям в виде графика.

Таким образом, применение указанных методов на этапе тестирования позволит удостовериться, что разработанное веб-приложение соответствует предъявляемым к нему требованиям и готово к выпуску [10, 11].

В статье проведено исследование на тему разработки системы тестов для тестирования веб-приложения. Для достижения данной цели были выполнены следующие задачи: рассмотрены виды и методы тестирования программного обеспечения; проведен анализ видов и методов тестирования в целях применения их для тестирования веб-приложения; определен порядок выполнения методов тестирования; приведено описание методов с примерами инструментов тестирования.

Литература

1. Сертифицированный тестировщик. Программа обучения Базового уровня: сайт / Rstqb – URL:

/https://www.rstqb.org/ru/ (Дата обращения 17.02.2024). Текст: электронный.

2. Кочкин, Д. В. Моделирование информационно-телекоммуникационной системы предприятия раскрашенными сетями Петри / Д. В. Кочкин, В. А. Горбунов // Вестник Череповецкого государственного университета. – 2024. – № 1(118). – С. 48–58.

3. Кочкин, Д. В. Проектирование и конструирование программного обеспечения : учебное пособие / Д. В. Кочкин, А. Н. Швецов. – Вологда : Вологодский государственный университет, 2023. – 127 с.

4. Куликов, С. С. Тестирование программного обеспечения. Базовый курс / С.С. Куликов. – URL: https://svyatoslav.biz/software_testing_book (Дата обращения 26.02.2024). Текст: электронный. – С. 67–110.

5. Интеллектуальные информационно-телекоммуникационные системы / А. Н. Швецов, А. А. Суконщиков, И. А. Андрианов [и др.]. – Вологда : Вологодский государственный университет, 2023. – 127 с.

6. Шагина, А. А. Разработка специализированной программы по обучению медицинского персонала работе в медицинской информационной системе / А. А. Шагина, Д. В. Кочкин // Вестник Вологодского государственного университета. Серия: Технические науки. – 2023. – № 1(19). – С. 67–70.

7. Модели и методы построения нейро-нечетких интеллектуальных агентов в информационно-телекоммуникационных системах / А. А. Суконщиков, И. А. Андрианов, С. В. Дианов [и др.]. – Курск : Закрытое акционерное общество «Университетская книга», 2021. – 152 с.

8. Принципы построения самоорганизующихся информационно-телекоммуникационных систем / А. А. Суконщиков, А. Н. Швецов, И. А. Андрианов, Д. В. Кочкин // Вестник Череповецкого государственного университета. – 2021. – № 1(100). – С. 56–67.

9. Building self-organizing information and telecommunications systems / A. A. Sukonschikov, A. N. Shvetsov, I. A. Andrianov, D. V. Kochkin // Journal of Physics: Conference Series, Krasnoyarsk, Russian Federation, 25 сентября – 04 2020 года. Vol. 1679. – Krasnoyarsk, Russian Federation: Institute of Physics and IOP Publishing Limited, 2020. – P. 32013.

10. Modeling the Elements of an Enterprise Infocommunication System Using Colored Petri Nets / A. Sukonschikov, D. Kochkin, A. Shvetsov [et al.] // Conference of Open Innovations Association, FRUCT. – 2020. – No. 26. – P. 660–666.

11. Кочкин, Д. В. Информационные сети и телекоммуникации / Д. В. Кочкин, А. А. Суконщиков. Том часть 1. – Курск : Закрытое акционерное общество «Университетская книга», 2016. – 233 с.

12. Duplicate and Plagiarism Search in Program Code Using Suffix Trees over Compiled Code / I. Andrianov, S. Rzheutskaya, A. Sukonschikov [et al.] // Conference of Open Innovations Association, FRUCT. – 2020. – No. 26. – P. 16–22.

A.A. Krasnov, D.V. Kochkin
Vologda State University

DEVELOPMENT OF TEST SYSTEM FOR WEB APPLICATION TESTING

The article presents the results of a study on the choice of testing methods used to create a test system for a web application. The methods of testing the web application are defined. The analysis of the applicability of the methods and their execution procedure is carried out.

Testing methods, web application testing, testing tools.