

А. О. ЛИСЕНКО, М. С. ТВЕРДА

РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ СТВОРЕННЯ ДОКУМЕНТАЦІЇ ДЛЯ ТЕСТУВАЛЬНИКІВ ТА ОЦІНКА ЙОГО ВПЛИВУ НА ЕФЕКТИВНІСТЬ ТЕСТУВАННЯ В ІТ-ПРОЕКТАХ

Розробка веб-застосунку для створення документації для тестувальників є важливим аспектом підвищення ефективності процесу тестування в ІТ-проектах. Сучасні ІТ-проекти стають дедалі складнішими, що зумовлює необхідність впровадження інструментів, які можуть забезпечити чітке й структуроване створення, підтримку та оновлення тестової документації. Така документація є невід'ємною частиною процесу забезпечення якості та відіграє ключову роль у взаємодії між учасниками команди. Метою статті є не лише опис функціональних можливостей розробленого веб-застосунку, але й оцінка його реального впливу на ефективність тестування в ІТ-проектах. Створення спеціалізованого веб-застосунку дозволяє автоматизувати рутинні процеси, зокрема документування тестових сценаріїв, звітів про помилки та результатів тестування. Це знижує час, що витрачається на ручне виконання цих завдань, і значно зменшує ризик людських помилок. Крім того, автоматизація сприяє підвищенню точності та ефективності тестування, оскільки дозволяє тестувальникам зосередитися на основних аспектах їхньої роботи замість механічних завдань. Розроблений веб-застосунок інтегрує функції управління проектами, що дає змогу ефективніше планувати, розподіляти та контролювати тестові завдання. Наприклад, функціональність для створення інтерактивних тестових сценаріїв, управління пріоритетами тестів і моніторингу прогресу дозволяє командам адаптуватися до змін у проекті в реальному часі. Це, своєю чергою, сприяє покращенню комунікації між тестувальниками, розробниками та іншими зацікавленими сторонами, забезпечуючи прозорість процесу тестування. У статті також розглядається вплив впровадження веб-застосунку на продуктивність команд. Було проведено порівняльний аналіз між часом, необхідним для ручного створення документації, і часом, який потрібен для створення тих самих документів за допомогою автоматизованого інструменту. Результати показали, що автоматизація дозволяє значно зменшити затрати часу на документування, одночасно підвищуючи якість і повноту створених звітів. Додатково обговорюються перспективи подальшого вдосконалення веб-застосунку та його інтеграції з іншими інструментами для управління ІТ-проектами.

Ключові слова: веб-застосунок, тестування, документація для тестувальників, оптимізація процесу тестування, зменшення часу на документування, ефективність тестування, ІТ-проект.

A. O. LYSENKO, M. S. TVERDA

DEVELOPMENT OF A WEB APPLICATION FOR CREATING DOCUMENTATION FOR TESTERS AND EVALUATION OF ITS IMPACT ON THE EFFECTIVENESS OF TESTING IN IT PROJECTS

The development of a web application for creating documentation for testers is a crucial aspect of improving the efficiency of the testing process in IT projects. Modern IT projects are becoming increasingly complex, necessitating tools that can ensure a clear and structured approach to creating, maintaining, and updating test documentation. This documentation is an integral part of the quality assurance process and plays a key role in the interaction between team members. The aim of this paper is not only to describe the functional capabilities of the web application but also to assess its real impact on testing efficiency in IT projects. The creation of a specialized web application enables the automation of routine processes, such as documenting test scenarios, bug reports, and testing outcomes. This reduces the time spent on manual execution of these tasks and significantly minimizes the risk of human error. Furthermore, automation enhances the accuracy and efficiency of testing by allowing testers to focus on critical aspects of their work instead of mechanical tasks. The developed web application integrates project management functionalities, allowing for more efficient planning, distribution, and control of testing tasks. For example, features such as creating interactive test scenarios, managing test priorities, and monitoring progress enable teams to adapt to project changes in real-time. This, in turn, improves communication between testers, developers, and other stakeholders, ensuring transparency in the testing process. The paper also examines the impact of implementing the web application on team productivity. A comparative analysis was conducted between the time required for manual documentation and the time needed to create the same documents using the automated tool. The results showed that automation significantly reduces the time spent on documentation while simultaneously improving the quality and completeness of the generated reports. Additionally, the prospects for further improvement of the web application and its integration with other IT project management tools are discussed.

Keywords: web application, testing, documentation for testers, optimizing the testing process, reducing the time for documentation, testing efficiency, IT project.

Вступ. Вступ. Стрімкий розвиток інформаційних технологій та зростання масштабів сучасних проектів висувають дедалі вищі вимоги до процесів тестування програмного забезпечення. Від ефективності цього процесу значною мірою залежить якість кінцевого продукту, дотримання термінів випуску та рівень задоволеності клієнтів. Однак традиційні підходи до створення та ведення документації для тестувальників нерідко не відповідають сучасним викликам, що спричиняє затримки, підвищення кількості помилок і перевитрати ресурсів. У цьому контексті впровадження спеціалізованих веб-застосунків для оперативного створення, оновлення та управління тестовою документацією може стати вирішальним кроком для підвищення ефективності тестування та оптимізації процесів забезпечення якості.

Актуальність роботи. Тестування програмного забезпечення відіграє критично важливу роль у виявленні дефектів на ранніх етапах розробки, забезпечуючи стабільність та якість кінцевого продукту. Це є одним із ключових етапів життєвого циклу будь-якого ІТ-проекту. Із зростанням складності сучасних проектів і скороченням термінів розробки зростає потреба у швидкому та ефективному створенні й веденні тестової документації. Використання веб-застосунків із вбудованими шаблонами для часткової автоматизації цього процесу дозволяє не лише оптимізувати роботу тестувальників, але й значно скоротити час, витрачений на рутинні завдання. Це сприяє підвищенню продуктивності команд, які займаються забезпеченням якості, і дозволяє їм зосередитися на складніших та творчих аспектах тестування.

©, А. О. Лисенко, М. С. Тверда, 2024

Постановка задачі. Метою даної роботи є аналіз розробленого веб-застосунку для створення документації для тестувальників, а також оцінка його впливу на ефективність тестування в ІТ-проектах. Особливу увагу приділено визначенню можливостей скорочення часу, витрат та зусиль, необхідних для написання й підтримки тестової документації. У роботі також розглядаються аспекти інтеграції веб-застосунку в існуючі процеси тестування, його здатність підвищувати продуктивність команди та зменшувати вплив людського фактору на якість документації.

Вирішення задачі. Розробка веб-застосунку для створення документації для тестувальників передбачає часткову автоматизацію та спрощення процесів документування, що спрямовані на підвищення точності й ефективності тестування. Автоматизація процесів створення, оновлення та управління тестовою документацією залишається актуальною темою досліджень і практичного впровадження впродовж багатьох років [1-6].

Тестування є одним із найважливіших етапів забезпечення якості програмного забезпечення, адже саме воно дозволяє виявити й усунути дефекти до того, як продукт потрапить у середовище «продакшн». Помилки, виявлені на ранніх етапах розробки, наприклад на етапі специфікації, є відносно недорогими для виправлення. Натомість помилки, що залишаються невиправленими до пізніх етапів розробки або навіть «продакшн», можуть значно підвищити витрати, як з точки зору часу, так і ресурсів.

Дослідження Г. Брайніка [8] підтверджують, що використання автоматизованих інструментів у процесах документування та тестування сприяє зменшенню ризику виникнення помилок, підвищенню доступності й зручності роботи з документацією. Такі інструменти допомагають запобігти людським помилкам, забезпечують оперативне оновлення тестової документації та підтримують її актуальність, що має особливе значення для тестувальників.

Крім того, можливість групування та структурованого представлення документації дозволяє тестувальникам швидше виявляти потенційно пов'язані помилки, запобігаючи їх накопиченню на наступних етапах розробки. Це не лише покращує процес тестування, але й знижує витрати, пов'язані з виправленням дефектів, та підвищує загальну якість програмного забезпечення.

Дослідження Г. А. Ді Лукка та його колег [9] демонструють, що для ефективного тестування веб-застосунків необхідно враховувати особливості взаємодії з браузером, клієнтськими технологіями та їхню поведінку в різних умовах. Недостатньо ретельне тестування може залишити серйозні помилки непоміченими, що, своєю чергою, може спричинити збої у роботі веб-застосунків, зниження продуктивності та негативний користувацький досвід.

М. Н. Алалфі [10] наголошує на важливості наявності чіткої навігаційної моделі веб-застосунків

як основи для створення тестових сценаріїв. Така модель дозволяє тестувальникам краще структурувати тестові кейси, враховуючи логіку та взаємодію елементів користувацького інтерфейсу. Впровадження цих підходів сприяє підвищенню якості тестування, забезпечуючи покриття різних сценаріїв використання веб-застосунку, включаючи крайні випадки та нестандартні умови.

Без комплексної моделі тестування неповна або неякісна документація може залишити важливі аспекти програми неперевіреними, що значно підвищує ризик випуску веб-продукту з критичними помилками. Щоб уникнути дорогих виправлень на пізніх стадіях розробки або після випуску продукту, необхідно забезпечити ефективне та всебічне тестування, яке базується на добре організованій та актуальній документації.

Для цього було поставлено завдання створити інструмент, який частково автоматизує процеси написання та оновлення тестової документації, забезпечуючи її швидке й зручне редагування. У рамках реалізації цього завдання було проведено аналіз потреб користувачів, а також досліджено сучасні підходи до тестування програмного забезпечення та написання тестової документації.

Розроблений веб-застосунок відповідає сучасним вимогам до програмного забезпечення та пропонує систему управління документацією, яка автоматизує основні процеси роботи з документами, зокрема створення, редагування, видалення та перегляд. Інтеграція з шаблонами дозволяє стандартизувати документи та прискорити їх створення (рис. 1). Такий підхід забезпечує зручність використання та підвищує ефективність роботи тестувальників.

Веб-застосунок включає розширений функціонал управління шаблонами, що дозволяє додавати, редагувати, видалити шаблони, а також налаштовувати доступ для окремих користувачів і команд. Крім того, система підтримує повний цикл роботи з користувачами, включаючи їхнє додавання, видалення, зміну ролей і організацію команд для ефективного розподілу завдань.

Система забезпечує надійні механізми авторизації, реєстрації та безпечного виходу користувачів, що гарантує захист даних. Важливим доповненням є інструменти глобального пошуку, які дозволяють швидко знаходити необхідну інформацію, оптимізуючи робочі процеси. Це рішення підвищує ефективність колаборації між учасниками команди та забезпечує високий рівень безпеки даних.

Хоча веб-застосунок орієнтований на використання в ІТ-сфері, його функціонал робить його універсальним інструментом, придатним для будь-якої галузі, де потрібно створювати, організовувати та управляти документацією. Основними користувачами є спеціалісти з тестування програмного забезпечення, розробники, керівники проєктів, а також вся ІТ-команда. Завдяки своїй гнучкості, застосунок може бути адаптований під потреби різних команд та організацій.

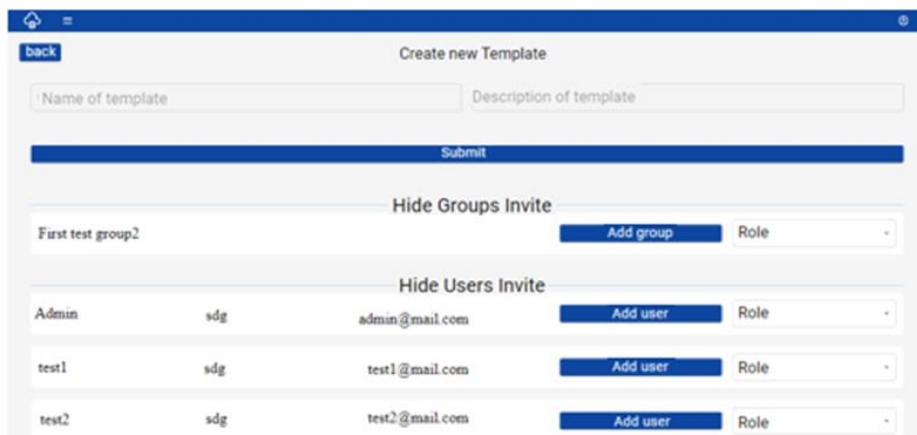


Рис. 1. Форма створення шаблону в веб-застосунку «TestDoc»

Застосунок підтримує три рівні ролей: адміністратор, редактор і переглядач. Адміністратор має розширені повноваження, включаючи управління користувачами (додавання, блокування, розблокування), а також повний доступ до створення, редагування та видалення документів для всіх команд. Редактор отримує права на редагування лише тих документів, до яких йому надано доступ, тоді як переглядач обмежується можливістю перегляду документів у межах своєї зони доступу. Такий розподіл ролей забезпечує чіткий контроль доступу, підвищує безпеку даних і сприяє ефективному управлінню процесами в команді.

Одним із ключових аспектів ефективного тестування програмного забезпечення є оптимізація процесу створення тестової документації. У традиційному підході, коли документи створюються вручну, тестувальники витрачають значну кількість часу на повторювані рутинні завдання. Це не лише знижує продуктивність команди, а й збільшує витрати компанії.

Результати дослідження показують, що впровадження часткової автоматизації написання документації дозволяє суттєво скоротити витрати часу на ці завдання. Було виконано розрахунок економії часу для команди з трьох тестувальників, який демонструє річний обсяг заощаджених годин. Крім того, було оцінено потенційну фінансову вигоду для компанії, яка виникає внаслідок оптимізації робочих процесів і підвищення загальної ефективності команди. Ці результати підтверджують економічну доцільність

впровадження автоматизованих інструментів у процеси створення тестової документації.

У рамках дослідження було проведено два експерименти для порівняння часу, необхідного для написання тестової документації вручну та з використанням розробленого веб-застосунку «TestDoc».

Перший експеримент полягав у створенні тестової документації з нуля у форматі Word-файлу. Було зафіксовано час, витрачений на написання документів вручну. Другий експеримент передбачав створення аналогічної документації за допомогою веб-застосунку з використанням заздалегідь підготовлених шаблонів. Це забезпечувало структурованість, стандартизацію й оптимізацію процесу створення документів.

Для об'єктивного порівняння в обох експериментах використовували однакові дані: 10 схожих багів і 10 тестових випадків, які потребували написання баг-репортів і тест-кейсів. Обсяг і зміст даних залишалися незмінними, що гарантувало точність та достовірність результатів.

Результати експериментів було представлено у вигляді порівняльної таблиці середнього часу написання баг-репорту (рис. 2). Згідно з отриманими даними, середній час написання одного баг-репорту вручну становив 16,61 хвилини, тоді як використання веб-застосунку скоротило цей час до 12,14 хвилини. Таким чином, завдяки «TestDoc» вдалося заощадити в середньому 4 хвилини 47 секунд на кожному баг-репорті.

	Вручну	З застосунком
Баг репорт 1	15,4	11,3
Баг репорт 2	16,5	12,1
Баг репорт 3	14,5	12,2
Баг репорт 4	17,5	12,2
Баг репорт 5	18	13,1
Баг репорт 6	17	12,5
Баг репорт 7	18,2	12,5
Баг репорт 8	14,3	11
Баг репорт 9	16,5	12,2
Баг репорт 10	18,2	12,3
Середнє значення	16,61	12,14

Рис. 2. Порівняльна таблиця середнього часу написання баг-репортів

Також було представлено порівняльну таблицю середнього часу написання тест-кейсу (рис. 3). Згідно з отриманими результатами, середній час написання одного тест-кейсу вручну перевищував час, необхідний для створення аналогічного тест-кейсу за

допомогою веб-застосунку, на 5 хвилин 16 секунд. Це демонструє, що використання «TestDoc» дозволяє значно скоротити час на виконання рутинних завдань, сприяючи підвищенню продуктивності тестувальників і зменшенню витрат часу на створення документації.

	Вручну	З застосунком
Тест-кейс 1	22,5	16,4
Тест-кейс 2	20,4	15,5
Тест-кейс 3	24,5	19,5
Тест-кейс 4	19,4	15,5
Тест-кейс 5	22,4	17,4
Тест-кейс 6	22,3	18,2
Тест-кейс 7	21,5	17,4
Тест-кейс 8	20,5	15,4
Тест-кейс 9	22,4	16,2
Тест-кейс 10	22,5	15,3
Середнє значення	21,84	16,68

Рис. 3. Порівняльна таблиця середнього часу написання тест-кейсів

На основі отриманих даних було проведено розрахунок потенційної економії часу та коштів для команди з трьох тестувальників. Щоденна економія часу для одного тестувальника на написання баг-репортів становить 13 хвилин 41 секунду, а на написання тест-кейсів – 10 хвилин 32 секунди. Загалом це дозволяє заощадити 24 хвилини 13 секунд на день для одного тестувальника. У місячному вимірі, за умови 22 робочих днів, економія часу для одного тестувальника становить 64,35 години. Для команди з трьох тестувальників цей показник досягає 193,05 години на місяць.

Враховуючи середню зарплату тестувальника у 25 000 грн і погодинну ставку 156,25 грн, економія коштів для команди з трьох тестувальників становить приблизно 30 164 грн на місяць. Ці результати підкреслюють значну фінансову вигоду від використання автоматизованих інструментів для створення тестової документації.

Висновок. Розроблений веб-застосунок «TestDoc» продемонстрував високу ефективність у скороченні часу, необхідного для створення та ведення тестової документації. Це дозволяє тестувальникам зосередитися на вирішенні критичних завдань, що сприяє підвищенню загальної продуктивності команди. Оптимізація процесів документування також зменшує кількість помилок, пов'язаних із ручним введенням даних, що позитивно впливає на якість тестування.

Заощаджений час і ресурси еквівалентні суттєвому скороченню витрат, що підтверджує економічну доцільність впровадження автоматизованих інструментів у процеси тестування. Крім того, застосунок має значний потенціал для подальшого вдосконалення, зокрема шляхом інтеграції з іншими популярними інструментами управління тестуванням і проектами. Це зробить його ще більш зручним, функціональним та корисним для

користувачів, сприяючи ширшому впровадженню в ІТ-компаніях.

Список літератури

1. Szatmári A., Gergely T., Beszédés Á. ISTQB-based Software Testing Education: Advantages and Challenges. 2023 *IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, м. Dublin, Ireland., 16–20 квіт. 2023 р. 2023. Doi: 10.1109/icstw58534.2023.00072.
2. Badgett T., Myers G. J., Sandler C. *Art of Software Testing*. Wiley & Sons, Incorporated, John, 2011. 256 p.
3. Barker T. *Perspectives on Software Documentation: Inquiries and Innovations* / ed. by T. T. Barker. Routledge, 2020. Doi: 10.4324/9781315223919.
4. Committee I. C. S. S. & S. E. S. IEEE standard for software and system test documentation. New York, NY : Institute for Electrical and Electronics Engineers, 2008. 118 p.
5. Blokdyk G. *Software test documentation: Standard Requirements*. CreateSpace Independent Publishing Platform, 2018. 134 p.
6. Shi Y., Peng Z., Hong, L., Yu Q. SoC-constrained team formation with self-organizing mechanism in social networks. *Knowledge-Based Systems*. 2017, vol. 138, pp. 1-14. doi: 10.1016/j.knosys.2017.09.018.
7. U. S. General Accounting Office Staff. *Greater Emphasis on Testing Needed to Make Computer Software More Reliable and Less Costly*. Independently Published, 2019.
8. Brajnik G. *Using Automatic Tools in Accessibility and Usability Assurance Processes. User-Centered Interaction Paradigms for Universal Access in the Information Society*. Berlin, Heidelberg, 2004. P. 219–234. Doi: 10.1007/978-3-540-30111-0_18.
9. Di Lucca G. A. *Testing Web-based applications: the state of the art and future trends. Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC 2005)*, Edinburgh, 26–28 July 2005. 2005. Doi: 10.1109/compsac.2005.95.
10. Alalfi M. H., Cordy J. R., Dean T. R. *A Survey of Analysis Models and Methods in Website Verification and Testing. Lecture Notes in Computer Science*. Berlin, Heidelberg. P. 306–311. Doi: 10.1007/978-3-540-73597-7_25.

References (transliterated)

1. Szatmári A., Gergely T., Beszédés Á. ISTQB-based Software Testing Education: Advantages and Challenges. 2023 *IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, м. Dublin, Ireland., 16–20 квіт. 2023 р. 2023. Doi: 10.1109/icstw58534.2023.00072.
2. Badgett T., Myers G. J., Sandler C. *Art of Software Testing*. Wiley & Sons, Incorporated, John, 2011. 256 p.

3. Barker T. Perspectives on Software Documentation: Inquiries and Innovations / ed. by T. T. Barker. Routledge, 2020. Doi: 10.4324/9781315223919.
4. Committee I. C. S. S. & S. E. S. IEEE standard for software and system test documentation. New York, NY : Institute for Electrical and Electronics Engineers, 2008. 118 p.
5. Blokdyk G. Software test documentation: Standard Requirements. CreateSpace Independent Publishing Platform, 2018. 134 p.
6. Shi Y., Peng Z., Hong, L., Yu Q. SoC-constrained team formation with self-organizing mechanism in social networks. *Knowledge-Based Systems*. 2017, vol. 138, pp. 1-14. doi: 10.1016/j.knosys.2017.09.018.
7. U. S. General Accounting Office Staff. Greater Emphasis on Testing Needed to Make Computer Software More Reliable and Less Costly. Independently Published, 2019.
8. Brajnik G. Using Automatic Tools in Accessibility and Usability Assurance Processes. *User-Centered Interaction Paradigms for Universal Access in the Information Society*. Berlin, Heidelberg, 2004. P. 219–234. Doi: 10.1007/978-3-540-30111-0_18.
9. Di Lucca G. A. Testing Web-based applications: the state of the art and future trends. *Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC 2005)*, Edinburgh, 26–28 July 2005. 2005. Doi: 10.1109/compsac.2005.95.
10. Alalfi M. H., Cordy J. R., Dean T. R. A Survey of Analysis Models and Methods in Website Verification and Testing. *Lecture Notes in Computer Science*. Berlin, Heidelberg. P. 306–311. Doi: 10.1007/978-3-540-73597-7_25.

Надійшла (received) 18.11.2024

Відомості про авторів / About the Authors

Лисенко Антон Олександрович (Lysenko Anton) – старший викладач кафедри управління проектами в інформаційних технологіях, кандидат технічних наук, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; e-mail: anton.lysenko@khpri.edu.ua, ORCID: <https://orcid.org/0009-0000-5588-8340>

Тверда Марія Сергіївна (Tverda Mariia) – студентка, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; e-mail: mariya2002tv@gmail.com, ORCID: <https://orcid.org/0009-0005-2701-705>