

# АНАЛІЗ ПІДХОДІВ ДО ПРІОРИТЕЗАЦІЇ ТЕСТ КЕЙСІВ

<sup>1</sup>Вінницький національний технічний університет

## Анотація

В роботі розглянуто застосування різних методів пріоритезації тест кейсів для підвищення якості програмного забезпечення за мінімальний час перевірки.

**Ключові слова:** тест кейси, розробка, програмне забезпечення.

## Abstract

The paper considers the different methods of prioritizing test cases and increasing the quality of developed software.

**Keywords:** test case, development, software.

## Вступ

Програмне забезпечення займає все більше й більше місця у нашому сучасному житті. Від правильності роботи програм залежить значна частина нашого життя, починаючи від зручності нашого повсякденного ритму до безпосереднього впливу на безпеку людей. Тому важливість тестування програмного забезпечення не втрачає своєї значущості.

Один із головних інструментів для забезпечення якості програмного забезпечення є ефективні тест кейси, що дають можливість провести стандартизований процес тестування. Систематизація та пріоритезація тест кейсів – це одна з найважливіших складових частин процесу тестування програмного забезпечення. Враховуючи той факт, що тестування сучасних комерційних програм може потребувати виконання від 2 до 10 тисяч тест-кейсів [1] питання ефективної систематизації та управління тест-кейсами є досить актуальними.

## Результати дослідження

Стрімкий розвиток ПЗ призвів до того, що тестувальникам потрібно працювати швидше та більш ефективно. Велика кількість тест кейсів для перевірки нового ПЗ чи регресії вже існуючого пропорційно збільшує час, необхідний для їх виконання, що впливає на швидкість релізу розроблюваного ПЗ. Тому тест-кейси потребують систематизації та управління, щоб зменшити час, необхідний для їх виконання, та забезпечити найбільш результативне тестування.

Систематизація тест-кейсів дозволяє зменшити навантаження на тестувальників, оскільки вони можуть працювати з меншою кількістю тест кейсів, які охоплюють більш широкий спектр тестових випадків. Крім того, систематизація тест-кейсів дозволяє:

- забезпечити більш точну оцінку покриття тестування та якості ПЗ;
- зменшити ризики при швидкій розробці ПЗ;
- забезпечити більш ефективне управління процесом тестування;
- швидше зорієнтуватися в тестовій документації та знайти необхідний тест кейс для проведення перевірки.

Існує чимало методів, які забезпечують менеджмент тестової документації. Згідно з однією з методологій, тестові кейси розбиваються на групи, що відповідають окремим вимогам до програмного забезпечення, після чого виконується аналіз кореляції між цими вимогами та тестовими кейсами. На основі отриманих даних визначається важливість кожного тестового кейсу в контексті окремої вимоги та в контексті всіх вимог загалом. Використання такого способу пріоритезації тестових сценаріїв дозволяє зменшити кількість тестових кейсів на 25%, при цьому час тестування зменшується приблизно на 21% [2]. Також завдяки такому методу можна знайти більше помилок у програмному забезпеченні за менший час.

За іншим підходом при проведенні пріоритизації тест-кейсів необхідно враховувати ряд факторів, які можуть бути поділені на 4 групи: фактори часу, фактори дефекту, фактори вимог та фактори складності [3]. Фактори часу включають час виконання і час перевірки отриманого результату з очікуваним і оцінюються за 10-бальною шкалою. Фактори дефекту кращим тест-кейсом визначають той, який дозволяв раніше виявляти дефекти, при чому тут вагу має і серйозність виявлених дефектів. За факторами вимог кращими будуть вважатись тест-кейси, які дозволяють виявляти дефекти у новій функціональності, які дозволяють покрити більшу кількість вимог або вимог, які визначені замовником як більш важливі. Фактори складності оцінюють наскільки тест-кейс є складним для виконання. Основним недоліком такого підходу є суб'єктивність оцінок, що залежить від досвіду тестувальників. Тому, на думку дослідників визначальними все ж таки є фактори дефектів, оскільки для їх вимірювання може бути використана більш об'єктивна інформація, взята з баг-трекінгових систем.

Неабияким ефективним вважається підхід, що базується на генетичних алгоритмах пріоритизації, для визначення оптимального порядку виконання тестових випадків. Генетичні алгоритми – це еволюційні алгоритми, які моделюють природні процеси еволюції, такі як селекція, схрещування і мутація, для пошуку рішень. У випадку пріоритизації тестових випадків, генетичний алгоритм використовується для знаходження оптимального порядку виконання тестів, щоб зменшити час виконання тестової рутини і збільшити його ефективність.

В результаті численних дослідження, основою яких став мультиоб'єктивний генетичний алгоритм для вирішення проблеми пріоритизації тестових випадків, використовувалися дві метрики, щоб максимізувати покриття коду тестовими випадками, мінімізувати кількість помилок, які можуть виникнути як результат розробки нового модуля ПО, максимізувати кількість виявлених помилок: "критичність" тестового випадку і "кількість виконаних рядків коду". Критичність тестового випадку відображає його важливість для тестування програми. Кількість виконаних рядків коду відображає кількість коду, який буде виконуватися під час виконання тесту. Обидві метрики використовуються як цільові функції для генетичного алгоритму [4]. В результаті експерименту виявлено, що використання пріоритетів тестування дозволяє зменшити кількість необхідних тестових випадків на 70-80% при збереженні високої ефективності тестування.

## Висновки

Дослідження показують, що систематизація тест-кейсів може допомогти підвищити швидкість перевірки ПЗ, зменшити кількість помилок та знизити час, необхідний для виявлення та виправлення дефектів.

Розробка системи пріоритизації тест кейсів дозволяє визначити тестові сценарії, які мають найбільшу важливість для ПЗ і потребують перевірки в першу чергу. А систематизація тест кейсів дозволяє підтримувати їх в організованому та структурованому вигляді, забезпечуючи повторне використання тестових сценаріїв та зменшення кількості дублювання коду при їх розробці. Загалом, систематизація і пріоритизація тест кейсів є необхідними процесами, які допомагають забезпечити якість ПЗ та зменшити час і зусилля, необхідні для його тестування.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Романюк О. В., Латуша А. В. Аналіз основних проблем автоматизованого тестування програмного забезпечення // Електронні інформаційні ресурси: створення, використання, доступ. Пам'яті Олексія Петровича Стахова. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 9-10 листопада 2021 р. – Суми/Вінниця: НІКО/ВНТУ, 2021. – 224 с. – ISBN 978-617-7422-16-6. – С. 174-176.
2. Test case prioritization based on requirement correlations. Tingting Ma, Hongwei Zeng, Xiaolin Wang 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). [Електронний ресурс]. – Режим доступу: <https://ieeexplore.ieee.org/document/7515934>.
3. Побігай О. О. Аналіз факторів пріоритизації тест-кейсів [Електронний ресурс] / О. О. Побігай, О. В. Романюк // Матеріали міжнародної науково-практичної інтернет-конференції «Молодь в технічних науках: дослідження, проблеми, перспективи», м. Вінниця, 12-17 червня 2017 р. – 2017. – Режим доступу: <http://conf.inmad.vntu.edu.ua/fm/index.php?page=materials&line=29&mat=425>.

4. Test Evaluation Metrics and Test Case Prioritization in the Dynamics of Software Testing. Shantanu Mukherjee, Sandip Roy, Pinaki Pratim Acharjya, 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT) [Електронний ресурс]. – Режим доступу: <https://ieeexplore.ieee.org/document/9616658>.

**Райська Людмила Василівна** – студент групи ІІІ-19Б, Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: [milaraiska@gmail.com](mailto:milaraiska@gmail.com).

**Романюк Оксана Володимирівна** – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com).

**Liudmyla Raiska** – student of group ІІІ – 19Б, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: [milaraiska@gmail.com](mailto:milaraiska@gmail.com).

**Oksana Romaniuk** – Candidate of Technical Sciences, Associate Professor of the Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com).