

технологій, як периферійні обчислення, хмарні середовища, відкриті бібліотеки та фреймворки ШІ, вимагає нових підходів до навчання, повторного використання ресурсів і забезпечення прозорості розробки.

Перехід до емпірично обґрунтованої розробки ПЗ, що базується на великих даних та відкритих платформах, створює основу для побудови стійких програмних екосистем. Наукове програмне забезпечення більше не є просто інструментом, а стає повноцінним науковим артефактом, розробка якого потребує глибокої інтеграції знань з ІТ, інженерії та фундаментальних наук.

Список використаних джерел

1. Prus O. V., Maidaniuk V. P., Arseniuk I. R. Analysis of the tools for the multi-project environment management : optimization of the software development (Ukr). *Scientific Works of Vinnytsia National Technical University*. 2024. No. 1. URL: <https://doi.org/10.31649/2307-5376-2024-1-29-36> (date of access: 10.04.2025).

2. С. О. Цибульник К. С. Барандич. ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЧАСТИНА 1. ЖИТТЄВИЙ ЦИКЛ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. Київ : КПІ ім. Ігоря Сікорського, 2022. 270 с.

3. Addi Malviya Thakur. The New Science of Scientific Software Development. *Oak Ridge National Laboratory*. 2021. DOI: 10.13140/RG.2.2.14173.49124.

Науковий керівник: Гращенко І. С., к.е.н., доцент.

***Папіж Любомир Миколайович, аспірант
факультет інформаційних систем та технологій,
ПВНЗ "Європейський університет"
l.m.papizh@gmail.com***

АНАЛІЗ ТЕСТОВАНOSTІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА МЕТОДИ ОЦІНКИ ЕФЕКТИВНОСТІ ТЕСТУВАННЯ

Тестування програмного забезпечення є ключовим етапом у процесі розробки інформаційних систем, спрямованим на забезпечення якості продукту. У сучасних умовах, коли складність програмних продуктів зростає, а вимоги до їхньої надійності та ефективності підвищуються, питання тестованості набуває особливої актуальності. Тестованість визначається як властивість програмного забезпечення, що полегшує його перевірку на відповідність заданим вимогам [1]. Оцінка ефективності тестування, у свою чергу, вимагає системного підходу, який враховує як внутрішні характеристики продукту, так і зовнішні фактори його розробки та використання.

Дослідження тестованості включає аналіз факторів, що впливають на можливість виявлення дефектів у програмному забезпеченні. Ця властивість залежить від структури коду, його модульності та прозорості внутрішньої архітектури [2]. Високий рівень тестованості сприяє скороченню часу на тестування та підвищенню достовірності результатів. Проте в багатьох проєктах виникають труднощі через недостатню документованість або складність коду, що ускладнює розробку ефективних тестових сценаріїв. Тому важливим є створення підходів, які дозволяють оцінювати тестованість ще на початкових етапах життєвого циклу розробки.

Оцінка якості тестування зосереджується на тому, наскільки процес тестування здатен виявляти дефекти та відповідати вимогам до продукту. Ефективність тестування залежить від повноти покриття коду, якості тестових сценаріїв та їхньої здатності виявляти критичні помилки [3]. У контексті гнучких методологій тестування інтегрується в усі етапи розробки, що вимагає гнучких методів оцінки, здатних адаптуватися до змін у структурі програмного забезпечення чи

специфікаціях. Такий підхід дозволяє своєчасно реагувати на нові виклики та вдосконалювати процес тестування.

Технології штучного інтелекту відкривають нові можливості для підвищення тестованості та ефективності тестування. Алгоритми машинного навчання можуть застосовуватися для автоматизації створення тестових випадків, аналізу коду та прогнозування потенційних проблемних зон [4]. Це дозволяє зменшити залежність від людського фактора та оптимізувати ресурси, спрямовані на тестування. Інтелектуальні системи здатні аналізувати взаємозв'язки між компонентами програмного забезпечення, сприяючи розробці стратегій тестування, які враховують як функціональні, так і нефункціональні характеристики продукту.

Розробка методів оцінки якості тестування на основі формальних моделей є ще одним перспективним напрямом. Такі моделі можуть охоплювати аналіз покриття коду, оцінку складності тестових сценаріїв або статистичні методи для прогнозування ймовірності пропуску дефектів [5]. Формальні підходи забезпечують об'єктивність у порівнянні різних стратегій тестування та допомагають обирати оптимальні рішення залежно від особливостей проєкту. Однак важливо уникати надмірної формалізації, яка може ускладнити практичне впровадження таких методів, зберігаючи їхню застосовність у реальних умовах.

Гнучкі системи розробки суттєво впливають на організацію тестування, оскільки воно часто відбувається паралельно з програмуванням [3]. Це вимагає інструментів, які забезпечують моніторинг тестованості в реальному часі та надають розробникам оперативний зворотний зв'язок. Ідеєю, вартою уваги, є використання графових моделей для оцінки тестованості: побудова графу залежностей між модулями програмного забезпечення дозволяє виявити критичні точки, що потребують детального тестування, та оптимізувати розподіл зусиль на перевірку.

Дослідження взаємозв'язку між тестованістю та якістю програмного забезпечення залишається актуальним завданням [2]. Хоча високий рівень тестованості сприяє швидкому виявленню дефектів, він не усуває їх повністю. Якість тестування визначає, наскільки достовірно продукт оцінюється перед випуском, впливаючи на управління ризиками. Це особливо важливо для систем, де помилки можуть призвести до значних наслідків, підкреслюючи роль тестування як інструмента контролю та прогнозування. Для узагальнення підходів до аналізу тестованості та оцінки ефективності тестування наведено таблицю 1.

Таблиця 1. Основні аспекти тестованості та оцінки ефективності тестування

Аспект	Опис	Джерело
Тестованість	Властивість програмного забезпечення, що полегшує виявлення дефектів	[1]
Фактори тестованості	Структура коду, модульність, прозорість архітектури	[2]
Ефективність тестування	Здатність тестів виявляти дефекти та відповідати вимогам	[3]
Штучний інтелект у тестуванні	Автоматизація тестів, аналіз коду, прогнозування проблемних зон	[4]
Формальні моделі	Аналіз покриття, оцінка складності, статистичні методи	[5]

Отже, аналіз тестованості програмного забезпечення та розробка методів оцінки ефективності тестування є важливими напрямками сучасних інформаційних технологій. Вони потребують поєднання теоретичних підходів, практичних інструментів і новітніх технологій, таких як штучний інтелект [4]. Висновок: інтеграція графових моделей із методами машинного навчання може підвищити точність оцінки тестованості, сприяючи створенню надійніших програмних продуктів.

Список використаних джерел

1. Alenezi, M. (2022). Investigating Software Testability and Test Cases Effectiveness. *International Journal of Advanced Computer Science and Applications (IJACSA)*. DOI: 10.48550/arXiv.2201.10090
2. Garousi, V., & Mäntylä, M. V. (2016). A systematic literature review of literature reviews in software testing. *Information and Software Technology*. DOI: 10.1016/j.infsof.2016.09.002
3. Rashid, J., Mahmood, T., & Nisar, M. W. (2019). A Study on Software Metrics and its Impact on Software Quality. *Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan*, 24(1), 62-70. DOI: 10.48550/arXiv.1905.12922
4. Minimol Anil Job. (2021). Automating and Optimizing Software Testing using Artificial Intelligence Techniques. *International Journal of Advanced Computer Science and Applications (IJACSA)*. DOI: 10.14569/IJACSA.2021.0120571
5. Akila, V., Vasuki, A., Christaline, J. A., Sathiya, R., Rishi, P., & Edward, A. S. (2023). Enhancing Software Testing with Machine Learning Techniques. *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. DOI: 10.1109/ICSCDS56580.2023.10105028.

Науковий керівник: Улічев О.С., к.т.н., доцент.

Печенкова Е. Є., студентка
Київський національний економічний
університет імені Вадима Гетьмана
elvirapechenkova1234@gmail.com

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ ЕЛЕКТРОННОЇ ТОРГІВЛІ В УМОВАХ ЦИФРОВОЇ ЕКОНОМІКИ

Електронна торгівля сьогодні – це не просто продаж товарів через Інтернет. Це цілий цифровий простір, у якому бізнес має бути гнучким, швидким та зручним для клієнта. Але на практиці багато онлайн-магазинів стикаються з типовими проблемами: ручне опрацювання замовлень, дублювання даних, повільне обслуговування, відсутність аналітики й персоналізованого підходу до покупця. У результаті – втрачені продажі, незадоволені клієнти, зайві витрати.

В умовах цифрової економіки питання автоматизації електронної комерції стає стратегічним. Саме тому було проведено дослідження, спрямоване на створення ефективної, гнучкої та зручної в користуванні інформаційної системи для онлайн-магазину одягу [1]. Мета – не лише спростити рутинні операції, а й створити умови для масштабування бізнесу, покращення клієнтського досвіду та підтримки управлінських рішень на основі даних.

У рамках дослідження було проаналізовано структуру типового Інтернет-магазину, виявлено «вузькі місця» в організації процесів, а також розроблено власну концепцію інформаційної системи, орієнтовану на модель роботи бренду Casual.trend. Рішення охоплює весь цикл взаємодії з клієнтом – від перегляду товару до підтвердження доставки.

Ключові можливості системи:

- зручна реєстрація користувачів та їх персоналізація;
- швидкий пошук і фільтрація товарів;
- оформлення замовлення з інтеграцією платіжних API (LiqPay);
- облік залишків на складі в режимі реального часу;
- аналітика продажів і поведінки користувачів;
- інтеграція з логістичними сервісами.

Систему побудовано на базі архітектури Model-View-Controller (MVC), що забезпечує гнучке