

Д.А. Макуховська, здобувач

М.О. Українець, аспірант

І.Д. Лімінович, асистент

Д.Д. Плечистий, к.т.н., доц.

Державний університет «Житомирська політехніка»

В.В. Буданов, експерт

«WholeCall»

Моделі тестування при розробці web-платформи для обміну тривимірними моделями

У статті досліджуються моделі забезпечення тестування в розробки web-платформи для обміну тривимірними моделями. Мета даної роботи полягає у визначенні певних підходів до тестування, які будуть ефективними та дозволяють підвищити якість і надійність продукту, орієнтованого на роботу з тривимірними моделями. Для досягнення цієї мети проаналізовано детерміністичні та імовірнісні моделі надійності програмного забезпечення. Детерміністичні моделі використовуються для оцінки структури та складності коду та оцінювання покриття тестами, що сприяє вчасному виявленню певних проблем у програмному коді. У свою чергу імовірнісні моделі використовуються, щоб враховувати випадкові чинники, такі як непередбачувані помилки і дозволяють прогнозувати надійність роботи платформи в реальних умовах використання. Основна увага приділена аналізу даних моделей, за для того, щоб визначити які краще у розрізі web-платформ з обміну тривимірними моделями, забезпечать надійну роботу продукту. У результаті дослідження встановлено, що детерміністичні моделі ефективні для забезпечення певної структурованості та вчасно визначення певних помилок, тоді як імовірнісні підходи краще підходять для моделювання реальних умов використання. Обидва методи доповнюють один одного, створюючи комплексний підхід до тестування. Отримані результати можуть бути використані для розробки, спрямованих на покращення продуктивності web-платформ до сучасних вимог. Це забезпечує значний внесок у розробку надійного продукту, web-ресурсу для роботи з тривимірною графікою.

Ключові слова: тестування; обмін; web-платформа; ресурси; моделі.

Актуальність теми. У сучасному світі розробка надійної та ефективної web-платформи для обміну 3D-моделями стала важливим завданням для багатьох розробників. Забезпечення якості таких платформ, стає першочерговим для покращення взаємодії з користувачем і сприяння розвитку творчості. На сьогоднішній час із розвитком технологій та зростанням інтересу до 3D-дизайну в різних галузях враховуючи і створення моделей, платформи для обміну 3D-моделями стають важливими. Такі ресурси дають можливість користувачам економити свій час та швидко знаходити готові моделі для своїх проєктів. Також такі web-платформи сприяють розвитку професійних спільнот. Вони дозволяють розробникам 3D-моделей, дизайнерам, художникам та іншим, обмінюватися своїми роботами, оцінювати їх, вдосконалювати свої навички. Це дозволяє швидко освоїти певні знання у цій галузі. Для багатьох компаній та окремих осіб простіше купити або завантажити готову 3D-модель, ніж створювати її з нуля. Такі web-платформи мають легко взаємодіяти з програмами для моделювання та дизайну, адже це полегшує роботу користувачів і дозволяє їм швидко використовувати моделі у своїх проєктах.

Аналіз останніх досліджень та публікацій, на які спираються автори. Тестування програмного забезпечення залишається критично важливим етапом у життєвому циклі розробки, спрямованим на забезпечення якості, надійності та продуктивності продуктів. Останні дослідження підкреслюють необхідність оптимізації процесів тестування, особливо у контексті динамічних середовищ та складних систем. Дослідження [12, с. 346] зосереджується на розробці математичної моделі, яка враховує масштаб проєкту для оптимізації процесу тестування. Автори наголошують на важливості адаптивного підходу до планування тестування залежно від обсягу проєкту та складності завдань. Запропонована модель дозволяє визначити оптимальні ресурси та часові витрати для підвищення ефективності тестування великих програмних систем. Ця праця [4, с. 13] є фундаментальним курсом, який охоплює основні принципи забезпечення якості ПЗ. Автори детально розглядають методи тестування, такі як модульне, інтеграційне та системне тестування, а також пропонують рекомендації щодо використання різних підходів залежно від специфіки проєкту. Значну увагу приділено управлінню дефектами та моніторингу якості упродовж усього життєвого циклу розробки. У цьому дослідженні [15, с. 209] автори пропонують методiku визначення критерію достатності тестування, який дозволяє оцінити, коли тестування можна вважати завершеним. Основним показником є покриття коду тестами та виконання ключових сценаріїв використання системи.

Практичне значення цієї роботи полягає у можливості мінімізувати витрати на тестування, не жертвуючи якістю програмного продукту. Стаття [1, с. 226] присвячена аналізу методів тестування web-додатків із застосуванням моделей нечіткої логіки для моніторингу процесів. Запропоновано підхід, що дозволяє враховувати змінні та нечіткі інтервали часу при моніторингу продуктивності та безпеки web-додатків. Така методика є особливо корисною для систем з високою кількістю одночасних користувачів, де необхідно забезпечити стабільність навіть за умов високої навантаженості. Таким чином, різні аспекти тестування програмного забезпечення – від оптимізації ресурсів до визначення критеріїв достатності та застосування сучасних методів моніторингу – є ключовими для створення високоякісних програмних продуктів, особливо в умовах зростаючої складності сучасних систем.

Метою дослідження є аналіз та порівняння детерміністичних та імовірнісних моделей тестування для забезпечення надійності, продуктивності та безпеки web-ресурсу та формулювання рекомендації щодо вибору оптимальних моделей.

Викладення основного матеріалу. Тестування web-платформ є важливим етапом розробки, який дозволяє виявити потенційні проблеми та забезпечити надійність, безпеку та вирішення певних задач та проблем. Але незалежно від типу платформи, існує перелік загальних проблем, які впливають на ефективність тестування. Сперечаюся на тематику проекту у контексті web-платформи для обміну 3D-моделлями, ці проблеми стають ще більш актуальними завдяки контенту web-ресурсу, а саме 3D-моделі. Одна з проблем це складність тестування крос-браузерної сумісності. Однією з ключових проблем web-платформ є забезпечити їх роботу у різноманітних браузерах та операційних системах. Відмінності у рендерах браузерів можуть призвести до відображення елементів інтерфейсу по-різному або некоректно, що спричинить деякі проблеми. Це стає особливо важливим для web ресурсу, що працює з 3D-моделлями, оскільки технології для рендерингу 3D-графіки (наприклад WebGL) можуть мати різну продуктивність в залежності від використання різних браузерів. Також однією з проблем є забезпечення швидкості завантаження. Завантаження та рендеринг 3D-моделей можуть суттєво впливати на продуктивність сайту, особливо якщо користувачі мають певні обмеження в ресурсах. Погана продуктивність може негативно вплинути на досвід користувача у використанні такої web-платформи. Основні проблеми продуктивності включають:

- проблеми у завантаженні великих 3D-файлів, адже вони можуть бути великими за розміром, що призводить до значного часу завантаження;
- проблеми з рендерингом у браузері, адже обробка складних моделей з текстурами може навантажувати браузер, що призводить до зависання платформи.

Тестування продуктивності включає перевірку швидкості завантаження сторінок, рендерингу моделей і роботи платформи під навантаженням, а також оптимізацію використання ресурсів браузера. Наступною проблемою web-платформ включаючи і з обміну 3D-моделлями є безпека. Це один з критично важливих аспектів тестування, особливо для сайту, що має містити контент, створений користувачами. Web-платформи для обміну 3D-моделлями, дозволяють користувачам завантажувати та обмінюватись контентом, що підвищує ризик атак типу SQL ін'єкції, XSS (cross-site scripting) або шкідливих скриптів, завантажених разом із 3D-моделлями. Щоб зменшити ризики, потрібно здійснювати постійне тестування безпеки, яке включає перевірку обробки та захист даних, файлів і т. д. Варто зазначити проблему, як тестування інтерфейсу користувача. Тестовані web-програми можуть мати складні функції та складну інфраструктуру. У цьому випадку можуть виникнути величезні проблеми з тестуванням інтерфейсу користувача, які виникають під час тестування всіх компонентів і функцій програми по черзі. Процес може бути тривалим і ресурсозатратним, що є серйозною проблемою для команди. Якщо команда тестування та розробників вирішить часто змінювати інтерфейс користувача, процес тестування може бути невинновано складним і важким для команди. Останнє, що потрібно розглянути з проблем це регресійне тестування після оновлень. З розвитком сайтів і їх функцій, часто виникають проблеми з регресіями – це ситуація, коли нові оновлення викликають непередбачувані ситуації, проблеми у вже працюючих функцій. Для платформ, що пов'язані з 3D-моделлями, регресійні помилки можуть впливати на рендеринг моделей та взаємодії з ними або завантаження нових. Регресійне тестування полягає у перевірці того, що нові зміни не порушили роботу раніше протестованих і функціонуючих частин платформи. Недостатнє або несвоєчасне регресійне тестування може призвести до серйозних проблем на платформі та погіршити користувацький досвід. Отже, основні проблеми тестування web-платформ, що включають роботу з 3D-моделлями це необхідність забезпечити сумісності з різними пристроями, браузерами, також підтримки високої продуктивності, захист від атак, забезпечення уникнення регресійних помилок. Для web-платформ, орієнтованих на обмін 3D-моделлями, ці проблеми ще більш загострюються через складність роботи з 3D-контентом та високу вимогливість до продуктивності системи.

Найефективніші моделі на проекті платформи. Моделі надійності програмного забезпечення дозволяють прогнозувати, оцінювати та поліпшувати роботу системи. На рисунку 1 зображено, що існують як детерміністичні (статичні), так і імовірнісні (динамічні) моделі [15, с. 209].

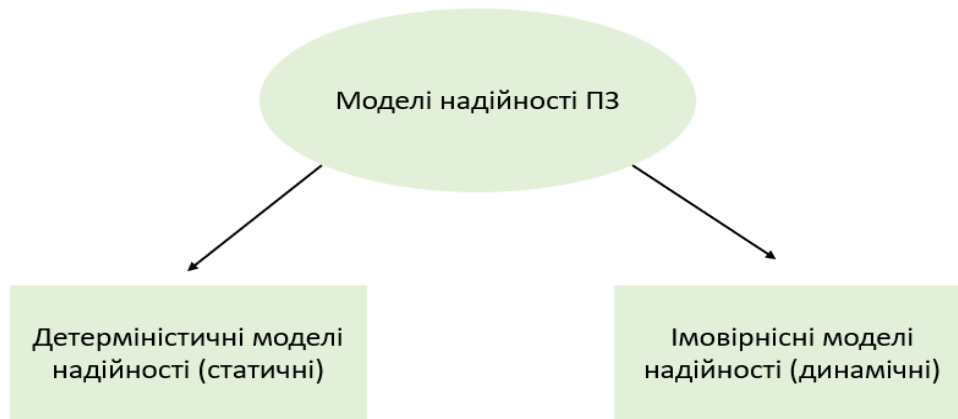


Рис. 1. Моделі надійності ПЗ

Детерміністичні моделі використовуються для оцінки складності програмного коду та передбачення можливих проблем, зокрема:

- підрахунку кількості операторів, операндів та інструкцій у програмі;
- вивчення потоку управління програми шляхом аналізу розгалужень та трасування шляхів виконання;
- аналізу потоку даних програми через спільне використання даних та їх передачу між частинами програми.

Оцінка продуктивності детерміністичних моделей проводиться через аналіз вихідного коду програми і не враховує жодних випадкових чинників або подій. Ці моделі надають кількісні оцінки на основі самого коду. Альтернативним підходом є теорія динаміки програмних систем, яка розглядає вплив вторинних помилок і не базується на ймовірнісних методах. Ця теорія також створює детерміністичні моделі, але динамічні, тобто такі, що враховують зміни в програмній системі з часом. Згідно з дослідженнями, ця модель є однією з найточніших для оцінки надійності програмного забезпечення і показує однаково надійні результати для різних програмних систем.

Імовірнісні моделі розглядають появу помилок і відмов у системі як випадкові події. Ці моделі класифікуються за кількома критеріями. Серед них виділяються моделі:

- висівання помилок – моделі, що передбачають видалення помилок на основі випадкових процесів;
- інтенсивність відмов – оцінюють ймовірність виникнення відмов у системі з плином часу;
- зростання надійності – передбачають збільшення надійності системи з часом, оскільки з кожним циклом тестування виявляються та усуваються помилки;
- архітектурні моделі – моделюють надійність програмного забезпечення на основі його архітектури;
- моделі неоднорідного пуассонового процесу – використовують пуассонівський розподіл для опису появи помилок у системі.

Узагальнені класифікації ймовірнісних моделей також включають характеристики, такі як час, обмеження на кількість помилок у програмі, типи розподілу випадкових відмов, а також форму функції для оцінки потоку відмов. Існують різні класифікаційні схеми, які дозволяють краще зрозуміти характеристики ймовірнісних моделей та їх застосовність для оцінки надійності програмного забезпечення.

Моделі, що використовуються у проєкті web-платформи з обміну 3D-моделей, зображені на рисунку 2 [14, с. 209].

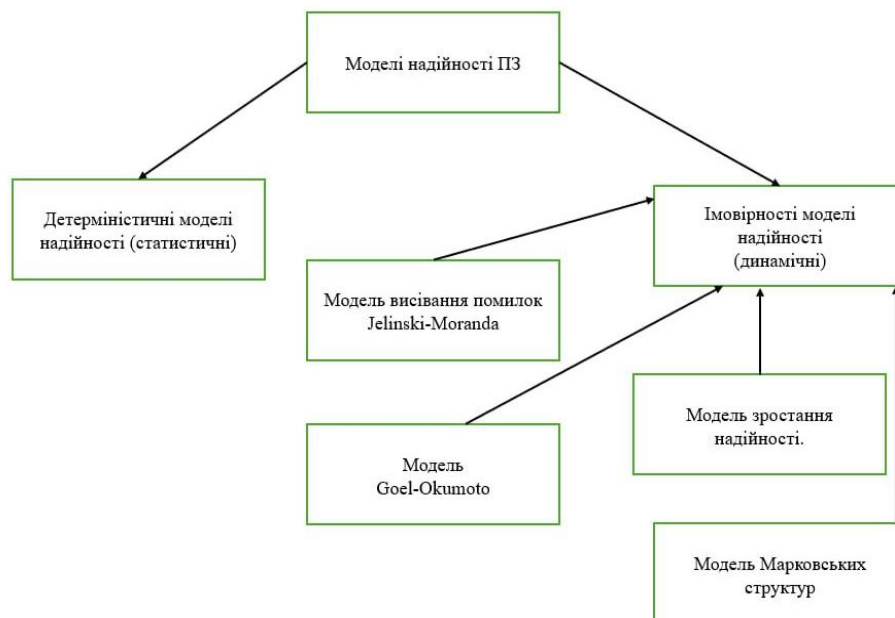


Рис. 2. Моделі, що використовуються

1. Модель висівання помилок Jelinski-Moranda [1, с. 226];

Ця модель дозволяє прогнозувати кількість помилок, що можуть з'явитися на платформі з часом. Вона корисна для визначення, скільки помилок залишилось у системі після тестування, що дозволяє налаштувати процес усунення дефектів. Ця модель припускає, що на кожній ітерації тестування усувається одна помилка, і загальна кількість помилок у системі є сталою. Формула для цієї моделі виглядає:

$$\lambda(t_i) = \frac{N-i+1}{N} \cdot \lambda_0,$$

де $\lambda(t_i)$ – інтенсивність помилок на i -му етапі тестування, N – загальна кількість помилок у системі, i – номер етапу (ітерації) тестування, λ_0 – початкова інтенсивність помилок (до тестування).

Схематично, модель Jelinski-Moranda можна представити як діаграму, де кожен етап тестування зменшує загальну кількість помилок. Це поступове «висівання» помилок:

Етап тестування 1 -> Виявлено помилки -> Усувається одна помилка

Етап тестування 2 -> Виявлено помилки -> Усувається ще одна помилка

...

Етап тестування N -> Всі помилки вилучено

2. Модель зростання надійності [14, с.209];

Ця модель описує, як система стає надійнішою з часом у міру усунення виявлених помилок. Для web-платформи, яка регулярно оновлюється та покращується, ця модель дозволяє оцінити, наскільки стабільною стає система після кожного релізу. Модель зростання надійності передбачає, що надійність системи збільшується зі збільшенням часу тестування через поступове усунення помилок. Надійність $R(t)$ як функція часу може бути виражена через експоненційну функцію:

$$R(t) = e^{-\lambda t},$$

де $R(t)$ – ймовірність безвідмовної роботи системи на момент часу t , λ – середня інтенсивність відмов, t – час роботи системи.

Цю модель можна зобразити графічно, де крива експоненційного зростання надійності демонструє зменшення частоти помилок з часом.

3. Модель Goel-Okumoto [1, с. 226];

Вона передбачає, що відмови відбуваються через непередбачені помилки у системі і дозволяє оцінити ймовірність виникнення помилок. Для платформи 3D-моделей це особливо корисно для прогнозування відмов при складних взаємодіях між користувачами та великою кількістю завантажень моделей. Ця модель передбачає, що кількість помилок у системі зменшується зі збільшенням часу тестування, але не всі помилки можуть бути виявлені. Кількість виявлених помилок $m(t)$ на момент часу t описується формулою:

$$m(t) = N(1 - e^{-\lambda t}),$$

де $m(t)$ – кількість виявлених помилок на момент часу t , N – загальна кількість помилок у системі, λ – параметр інтенсивності помилок, t – час тестування.

Схематично можна зобразити як криву, яка показує, що з часом кількість виявлених помилок наближається до максимуму N , але не всі помилки можуть бути знайдені.

4. Модель Марковської структури [13, с. 130].

Ця модель базується на ймовірнісних переходах між станами системи, що корисно для визначення можливих відмов під час високого навантаження або складних операцій. Використання цієї моделі дозволяє покращити управління ресурсами та знизити ризик критичних збоїв. Для web-платформи з 3D-моделями можна змоделювати різні стани системи: нормальний стан, перевантаження та збій. Як приклад можна використовувати перехідну матрицю для трьох станів системи (нормальний, перевантаження, збій).

Аналіз проблематики та моделей web-платформи. Розглянувши проблематику web-платформи з обміну 3D-моделями варто підібрати тестові моделі, які найкраще відповідають кожній проблемі та можуть забезпечити ефективне її вирішення [1, с. 226]. Перша проблема це кросбраузерна сумісність. Проблема пов'язана з тим, що різні браузерери можуть по-різному рендерити елементи інтерфейсу, особливо при роботі з WebGL для 3D-графіки.

Модель які краще підходить для вирішення, це модель Марковської структури [14, с. 209]. Допоможе створити ймовірнісну модель переходів системи у різних станах в залежності від браузера чи операційної системи. Модель дозволяє передбачити можливі відмови або некоректну поведінку залежно від конфігурацій, зокрема при високих навантаженнях.

Наступна проблема, яка була розглянута це продуктивність і швидкість завантаження. З огляду на високу вимогливість до продуктивності при роботі з великими 3D-моделями, важливо зменшити час завантаження контенту та оптимізувати використання ресурсів браузера.

Рекомендовані моделі:

- модель зростання надійності [13, с. 130]: допоможе оцінити, як збільшується продуктивність платформи після оптимізації та усунення помилок. Застосування цієї моделі дозволить оцінити загальний ефект від внесених змін і наскільки швидко стабільність системи зростає після кожного випуску;
- модель Goel-Okumoto: корисна для оцінки продуктивності при різних сценаріях використання. Дозволяє передбачити кількість можливих відмов при завантаженні великих моделей і при взаємодії з користувачами, зокрема в умовах високих навантажень.

Далі важливим аспектом є безпека для web-платформи, адже оскільки користувачі можуть завантажувати контент, що створює ризик атак на зразок SQL-ін'єкцій або XSS.

Рекомендована модель це модель висівання помилок Jelinski-Moranda [8, с. 33]. Допомогає передбачити кількість помилок, які ще залишаються в системі, що дозволяє коригувати та вдосконалювати процес усунення дефектів, зокрема в контексті безпеки. Використання цієї моделі дозволяє орієнтуватися на залишкові ризики.

Також описано тестування інтерфейсу користувача і це є трудомістким процесом, особливо коли змінюється дизайн або додаються нові функції, що може викликати проблеми з сумісністю і продуктивністю.

Рекомендована модель це модель Марковської структури [14, с. 209], адже може змоделювати різні сценарії взаємодії користувачів із системою, зокрема, коли інтерфейс змінюється. Це допомагає передбачити, як зміни впливатимуть на взаємодію з системою і чи можливі помилки під час навантаження.

І останнє це регресійне тестування після оновлень. Моделі які краще використовувати в такому випадку, це:

- модель висівання помилок Jelinski-Moranda [8, с. 33]: корисна для визначення залишкових помилок після проведення тестування, що дозволяє оцінити, скільки помилок залишається у системі після оновлення;
- модель зростання надійності: забезпечує оцінку того, наскільки платформа стає надійнішою після чергового оновлення і наскільки зростає стабільність. Модель показує поступове покращення продуктивності системи з кожним оновленням.

Отже, різні тестові моделі можуть бути ефективно застосовані для вирішення основних проблем web-платформи з 3D-моделями. Таким чином, об'єднання різних підходів дозволить комплексно вирішити основні проблеми платформи, підвищуючи її продуктивність, надійність, сумісність і безпеку.

Результати та обговорення. За результатами проведеного аналізу було визначено, що різні моделі мають свої переваги та недоліки для вирішення проблем, щодо розробки web-платформи з обміну 3D-моделями. Наприклад, для вирішення проблеми кросбраузерної сумісності найбільш вдалою та ефективною є модель Марковської структури. Ця модель дозволяє прогнозувати помилки, а також ефективна при тестуванні. Щодо проблем продуктивності та швидкості завантаження, найбільш ефективними є моделі зростання надійності та Goel-Okumoto. Адже ці моделі забезпечують точне моделювання продуктивності платформи у різних випадках та при високих навантаженнях. Для підвищення безпеки web-ресурсу, ефективною є модель висівання помилок Jelinski-Moranda. Використання моделей Марковської структури, зростання надійності, Goel-Okumoto та Jelinski-Moranda дозволило якісно та комплексно підходити до вирішення певних та основних проблем web-платформи з

обміну 3D-моделлями. Даний підхід сприяє підвищенню продуктивності, надійності та безпеки системи. Об'єднання таких різних моделей забезпечує гнучке тестування та оптимізацію web-ресурсу, що є ключовим фактором.

Висновки. Розглянувши моделі тестування програмного забезпечення, можна зробити висновок, що для розробки web-платформи для обміну 3D-моделлями важливо використовувати як детерміністичні, так і імовірнісні підходи. Обидва підходи мають свої переваги та недоліки, які необхідно враховувати при впровадженні системи забезпечення якості. Детерміністичні моделі, такі як Halstead і McCabe, пропонують кількісні методи оцінки складності коду, що дозволяє ефективно ідентифікувати проблемні ділянки на ранніх етапах розробки. Це дозволяє підвищити загальну продуктивність і стабільність платформи за рахунок своєчасного виявлення та усунення складних фрагментів коду. Однак ці моделі є статичними і не завжди здатні врахувати динамічні зміни в роботі платформи або специфіку сучасних технологій, таких як 3D-графіка. Імовірнісні моделі, як-от Goel-Okumoto та Jelinski-Moranda, дозволяють враховувати випадкові чинники та передбачити появу помилок у реальних умовах експлуатації платформи. Ці моделі добре підходять для середовищ із великою кількістю невизначеностей, допомагаючи поліпшити стабільність і надійність платформи з кожним циклом тестування. Водночас залежність від точних вхідних даних і певні труднощі в прогнозуванні рідкісних помилок можуть обмежити їхню ефективність у деяких випадках. Отже, для забезпечення високої якості та стабільності web-платформи для обміну 3D-моделлями доцільно використовувати комбінацію обох типів моделей. Детерміністичні моделі забезпечують базову оцінку складності та продуктивності, а імовірнісні – дозволяють враховувати реальні умови роботи платформи, допомагаючи знизити ризик помилок і відмов. Інтеграція різних підходів до тестування допоможе забезпечити надійність та безпеку платформи, водночас підвищуючи її адаптивність до швидкозмінних умов сучасних технологій.

Список використаної літератури:

1. Оптимізація моделі тестування програмного забезпечення з показником величини проекту / М.О. Бубенко, В.Кукурба, Л.Гнатів, Я.М. Чабанюк // Вісник Національного університету «Львівська політехніка». Комп'ютерні науки та інформаційні технології. – 2011. – № 694. – С. 226–232.
2. Гук Н.А. Алгоритм побудови моделі веб-сайту / Н.А. Гук, С.В. Диханов, О.Д. Матющенко // Праці Національного технічного університету України. – 2020.
3. Крепич С.Я. Якість програмного забезпечення та тестування: базовий курс / С.Я. Крепич, І.Я. Снівак. – Київ : Центр навчальної літератури, 2020.
4. Кузьмич О.М. Удосконалений підхід до автоматизованого інтеграційного тестування ПЗ за умов застосування безсерверної архітектури / О.М. Кузьмич, В.Я. Лахай, М.М. Сенів // Науковий вісник НЛТУ України. – 2023. – Т. 33, № 3. – С. 97–101.
5. Liu J. Parameter estimation of Jelinski-Moranda model based on weighted nonlinear least squares and heteroscedasticity / J.Liu, Y.Liu, M.Xu // arXiv: 1503.00094. – 2015.
6. Лісова Т.В. Моделі та методи сучасної теорії тестів / Т.В. Лісова. – Ніжин : Видавець Лисенко М.М., 2012.
7. Мілютіна К.Л. Інновації в інформаційних технологіях / К.Л. Мілютіна, А.Ю. Трофімов. – Харків : Щедра Садиба Плюс, 2020.
8. Паламарчук К.А. Тестування. Моделі розробки програмного забезпечення / К.А. Паламарчук // Журнал досліджень тестування програмного забезпечення. – 2020. – Т. 5, № 1. – С. 33–44.
9. Руденко О.А. Аналіз моделей надійності на основі різномірного процесу Пуассона з метою врахування неповного налагодження програмного забезпечення : дис. ... доктора техн. наук / О.А. Руденко, З.М. Руденко, Н.М. Ревуцька. – International Scientific Unity, 2024.
10. Метод побудови сценаріїв тестування програмного забезпечення на основі аналізу його змінних / Д.В. Федасюк, В.С. Яковина, П.В. Сердюк, О.О. Нитребич // Вісник Національного університету «Львівська політехніка». – 2014. – № 771. – С. 209–213.
11. Чабанюк Я.М. Критерій достатності процесу тестування програмного забезпечення / Я.М. Чабанюк, В.Кукурба, Л.Гнатів // Вісник Національного університету «Львівська політехніка». Комп'ютерні науки та інформаційні технології. – 2011. – № 672. – С. 346–358.
12. Яковина В.С. Методи та засоби аналізу надійності функціонування програмного забезпечення з урахуванням етапів життєвого циклу / В.С. Яковина. – 4-те вид. – Львів : Нац. ун-т «Львівська політехніка», 2016.
13. Яковина В.С. Огляд і аналіз моделей надійності програмного забезпечення / В.С. Яковина, В.Мацелюх // Вісник Національного університету «Львівська політехніка». Комп'ютерні науки та інформаційні технології. – 2017. – № 864. – С. 130–140.
14. Використання марковських ланцюгів вищого порядку в задачах моделювання надійності програмного забезпечення / В.С. Яковина, П.В. Сердюк, О.О. Нитребич, Д.В. Федасюк // Вісник Національного університету «Львівська політехніка». – 2013. – № 771. – С. 209–213.

References:

1. Bubenko, M.O., Kukurba, V., Hnativ, L. and Chabaniuk, Ya.M. (2011), «Optimizatsiia modeli testuvannia prohramnoho zabezpechennia z pokaznykom velychyny proektu», *Visnyk Natsionalnogo universytetu «Lvivska politekhnika»*. *Kompiuterni nauky ta informatsiini tekhnolohii*, No. 694, pp. 226–232.

2. Huk, N.A., Dykhanov, S.V. and Matiushchenko, O.D. (2020), «Alhorytm pobudovy modeli veb-saitu», *Pratsi Natsionalnoho tekhnichnoho universytetu Ukrainy*.
3. Krepych, S.Ia. and Spivak, I.Ia. (2020), *Yakist prohramnoho zabezpechennia ta testuvannia: bazovyi kurs*, Tsentr navchalnoi literatury, Kyiv.
4. Kuzmich, O.M., Lakhai, V.Ia. and Seniv, M.M. (2023), «Udoskonalenyi pidkhd do avtomatyzovanoho intehratsiinoho testuvannia PZ za umov zastosuvannia bezservernoi arkhitektury», *Naukovyi visnyk NLTU Ukrainy*, Vol. 33, No. 3, pp. 97–101.
5. Liu, J., Liu, Y. and Xu, M. (2015), «Parameter estimation of Jelinski-Moranda model based on weighted nonlinear least squares and heteroscedasticity», *arXiv:1503.00094*.
6. Lisova, T.V. (2012), *Modeli ta metody suchasnoi teorii testiv*, Vydavets Lysenko M.M., Nizhyn.
7. Miliutina, K.L. and Trofimov, A.Yu. «2020», *Innovatsii v informatsiinykh tekhnolohiiakh*, Shchedra Sadyba Plus, Kharkiv.
8. Palamarchuk, K.A. (2020), «Testuvannia. Modeli rozrobky prohramnoho zabezpechennia», *Zhurnal doslidzen testuvannia prohramnoho zabezpechennia*, Vol. 5, No. 1, pp. 33–44.
9. Rudenko, O.A., Rudenko, Z.M. and Revutska, N.M. (2024), «Analiz modelei nadiinosti na osnovi riznoridnoho protsesu Puassona z metoiu vrakhuvannia nepovnoho nalahodzhennia prohramnoho zabezpechennia», D.Sc. Thesis of dissertation, International Scientific Unity.
10. Fedasiuk, D.V., Yakovyna, V.S., Serdiuk, P.V. and Nytrebych, O.O. (2014), «Metod pobudovy stsenariiv testuvannia prohramnoho zabezpechennia na osnovi analizu yoho zminnykh», *Visnyk Natsionalnoho universytetu «Lvivska politekhnika»*, No. 771, pp. 209–213.
11. Chabaniuk, Ya.M., Kukurba, V. and Hnativ, L. (2011), «Kryterii dostatnosti protsesu testuvannia prohramnoho zabezpechennia», *Visnyk Natsionalnoho universytetu «Lvivska politekhnika». Kompiuterni nauky ta informatsiini tekhnolohii*, No. 672, pp. 346–358.
12. Yakovyna, V.S. (2016), *Metody ta zasoby analizu nadiinosti funktsionuvannia prohramnoho zabezpechennia z urakhuvanniam etapiv zhyttievoho tsyклу*, 4th ed., Nats. un-t «Lvivska politekhnika», Lviv.
13. Yakovyna, V.S. and Matseliukh, V. (2017), «Ohliad i analiz modelei nadiinosti prohramnoho zabezpechennia», *Visnyk Natsionalnoho universytetu «Lvivska politekhnika». Kompiuterni nauky ta informatsiini tekhnolohii*, No. 864, pp. 130–140.
14. Yakovyna, V.S., Serdiuk, P.V., Nytrebych, O.O. and Fedasiuk, D.V. (2013), «Vykorystannia markovskyykh lantsiuhiv vyshchoho poriadku v zadachakh modeliuвання nadiinosti prohramnoho zabezpechennia», *Visnyk Natsionalnoho universytetu «Lvivska politekhnika»*, No. 771, pp. 209–213.

Макуховська Дар'я Андріївна – бакалавр Державного університету «Житомирська політехніка».

<https://orcid.org/0009-0001-1899-4678>.

Наукові інтереси:

- розробка та проектування вебдодатків;
- UX/UI дизайн, графічний дизайн.

Українець Микола Олександрович – асистент кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<https://orcid.org/0009-0002-1185-491X>.

Наукові інтереси:

- веборієнтовані системи;
- нейронні мережі;
- автономні транспортні засоби;
- комп'ютерний зір.

Лімінович Іван Дмитрович – асистент кафедри інженерії програмного забезпечення, викладач-практик, магістр Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0003-2196-4186>.

Наукові інтереси:

- вебтехнології;
- розробка та оптимізація великих вебдодатків.

Плечистий Дмитро Дмитрович – кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0002-4803-159X>.

Наукові інтереси:

- моделювання та оптимізація;
- аналіз даних;
- тестування;
- алгоритми та структури.

Буданов Валентин Володимирович – експерт у сфері управління інноваційним технологічним стартап-проектом «WholeCall».

<https://orcid.org/0009-0004-8828-0656>.

Наукові інтереси:

- штучний інтелект;
- тестування.

Makuhovska D.A., Ukrainets M.O., Liminovich I.D., Plechysty D.D., Budanov V.V.

Testing models in the development of a web platform for the exchange of three-dimensional models

The article investigates the models of testing support in the development of a web-based platform for the exchange of three-dimensional models. The purpose of this work is to determine certain approaches to testing that will be effective and allow improving the quality and reliability of a product focused on working with three-dimensional models. To achieve this goal, deterministic and probabilistic models of software reliability are analysed. Deterministic models are used to evaluate the structure and complexity of the code and assess test coverage, which helps to detect certain problems in the software code in a timely manner. In turn, probabilistic models are used to take into account random factors, such as unpredictable errors, and allow predicting the reliability of the platform in real-world conditions. The main focus is on analysing these models in order to determine which ones are best suited to web-based platforms for exchanging 3D-models and ensure reliable product performance. The study found that deterministic models are effective in providing a certain degree of structure and timely detection of certain errors, while probabilistic approaches are better suited to modelling real-world conditions of use. Both methods complement each other, creating a comprehensive approach to testing. The results obtained can be used for development aimed at improving the performance of web platforms to meet modern requirements. This provides a significant contribution to the development of a reliable product, a web resource for working with three-dimensional graphics.

Keywords: testing; exchange; web platform; resources; models.

Стаття надійшла до редакції 16.04.2025.