



ЗАСОБИ РОЗРАХУНКУ ПОКАЗНИКІВ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ПІДСТАВІ МОДЕЛЕЙ З УРАХУВАННЯМ НЕДОСКОНАЛОГО ВІДЛАГОДЖЕННЯ

Проаналізовано моделі надійності програмного забезпечення (ПЗ) з урахуванням процесу недосконалого відлагодження, внаслідок чого визначено, що вони не враховують показників, які характеризують програмне забезпечення або ж процес його тестування. Відповідно, вдосконалення наявних і побудова нових математичних моделей надійності ПЗ, які враховували б його складність, архітектуру та етапи життєвого циклу, а також розроблення відповідних методів і засобів для аналізу надійності функціонування ПЗ, є актуальною проблемою, результати реалізації якої дадуть змогу підвищити достовірність оцінювання показників надійності сучасного ПЗ. Для модифікації обрано модель Гоеля-Окумото та S-подібну модель зі затримкою, для яких введено коефіцієнт покриття коду тестами, що уможливило враховувати кількість протестованих рядків коду під час прогнозування показників надійності ПЗ. На підставі удосконалених моделей розроблено алгоритм розрахунку показників надійності та програмний засіб для автоматизованого розрахунку показників надійності ПЗ, під час розроблення якого було використано веб-фреймворк Angular 12, технології HTML, CSS та SCSS. Для виконання математичних обчислень використано web-worker, який є вбудованим у фреймворк Angular. Розроблений програмний засіб реалізовано як веб-застосунок, який може виконуватися на будь-якому пристрої у браузері. Для роботи програми необхідним є використання платформи Node.js на стороні сервера. Створений програмний продукт дає змогу виконувати розрахунок показників надійності ПЗ та прогнозувати кількість помилок, які залишилися в розроблюваному ПЗ, на підставі зібраних статистичних даних про знайдені програмні дефекти та коефіцієнт покриття коду тестами, що дасть змогу підвищити рівень адекватності розрахунку та прогнозування вищезазначених показників, а також оптимальніше розподіляти ресурси проєкту.

Ключові слова: інженерія ПЗ; моделі надійності ПЗ; методи розрахунку показників надійності ПЗ; модель Гоеля-Окумото; S-подібна модель з затримкою.

Вступ / Introduction

Важливою проблемою сучасності є питання забезпечення надійності технологій. Розвиток сучасних інформаційних технологій та використання програмно-апаратних комплексів і систем у всіх сферах людського життя призводить до значного збільшення ризику виникнення збоїв у роботі такого обладнання. У цьому випадку через специфіку характеру програмного забезпечення (ПЗ) та особливості його поведінки з точки зору надійності, питання аналізу та забезпечення його надійності все ще залишається широким полем для досліджень. Разом із ускладненням технічного завдання пропорційно зростає складність ПЗ і, як наслідок – процес точного і правдивого оцінювання показників надійності пропорційно ускладнюється.

Для кількісного оцінювання та прогнозування надійності ПЗ використовуються відповідні моделі надійності, призначені для передбачення кількості програмних дефектів у ПЗ. Існує багато моделей надійності ПЗ, які переважно ділять на два класи: статичні та динамічні [1, 9]. Поширеніші з них динамічні моделі припускають, що в програмі кількість помилок є величи-

ною дискретною, а враховуючи, що кумулятивне значення величини помилок – це ціле число, динамічні моделі розраховують величину початкових дефектів на етапі відлагодження ПЗ і кількість дефектів, які ми отримуємо на кінець етапу відлагодження. Ці моделі використовують показник інтенсивності відмов як основну характеристику появи відмови. Звичайно, що динамічні моделі припускають, що інтенсивність відмов є сталим числом або функцією часу відлагодження або ж випадковим числом із заданим законом розподілу [6]. Багато моделей припускають ідеальне відлагодження – імовірність усунення помилки становить 100 %, під час усунення помилки не вводиться нових, у такий спосіб кількість помилок, що залишилась в програмі, є незростаючою функцією тривалості відлагодження. У моделях, що припускають не ідеальне відлагодження (помилки не усуваються, вводяться, чи не змінюються під час кожного відлагодження), кількість помилок, що залишились, може зростати чи зменшуватись [9]. Але на сьогодні не існує універсальної моделі надійності ПЗ, яка б повністю адекватно описувала процес прогнозування надійності та була б застосовною для всіх видів і типів ПЗ, а також враховувала всі метрики, характерис-

Інформація про авторів:

Сенів Максим Михайлович, канд. техн. наук, доцент, кафедра програмного забезпечення. Email: max1sudden@gmail.com;

<https://orcid.org/0000-0003-1044-4628>

Роїк Остап Орестович, студент, кафедра програмного забезпечення. Email: ostap2308@gmail.com

Цитування за ДСТУ: Сенів М. М., Роїк О. О. Засоби розрахунку показників надійності програмного забезпечення на підставі моделей з урахуванням недосконалого відлагодження. Науковий вісник НЛТУ України. 2021, т. 31, № 6. С. 87–91.

Citation APA: Seniv, M. M., & Roik, O. O. (2021). Means of calculating the reliability of software based on models, taking into account imperfect debugging. *Scientific Bulletin of UNFU*, 31(6), 87–91. <https://doi.org/10.36930/40310613>

тики та особливості програмних продуктів. Відповідно, актуальною науковою задачею є створення нових й удосконалення наявних моделей надійності ПЗ, які б точніше здійснювали прогнозування показників надійності ПЗ та враховували різноманітні характеристики проєктів.

Об'єкт дослідження – розрахунок надійності ПЗ.

Предмет дослідження – моделі та методи оцінювання та прогнозування показників надійності ПЗ з урахуванням недосконалого відлагодження, що дасть змогу підвищити достовірність розрахунку та прогнозування даних показників.

Мета роботи – удосконалити моделі надійності ПЗ, які давали б змогу враховувати коефіцієнт покриття коду тестами.

Для досягнення зазначеної мети визначено такі основні завдання дослідження: удосконалити моделі надійності ПЗ з урахуванням коефіцієнта покриття коду тестами; розробити алгоритм розрахунку надійності ПЗ з використанням удосконалених моделей; реалізувати розроблений алгоритм та верифікувати розроблені засоби на тестових даних реальних ІТ-проєктів.

Наукова новизна отриманих результатів дослідження – вперше удосконалено модель Гоеля-Окумото та S-подібну модель із затримкою, шляхом врахування

коефіцієнта покриття коду тестами, що дасть змогу підвищити рівень адекватності розрахунку та прогнозування показників надійності ПЗ.

Практична значущість результатів дослідження – можливість використання розроблених засобів для прогнозування кумулятивної кількості помилок у розроблюваному програмному продукті, що водночас дасть змогу оптимальніше розподілити ресурси проєкту.

Аналіз останніх досліджень та публікацій. Вартість виправлення дефекту на етапі експлуатації ПЗ є в рази вищою, аніж вартість виправлення дефекту на етапі розроблення чи тестування. Звичайно, що такі етапи життєвого циклу програмного забезпечення, як тестування чи від лагодження, можуть зменшити кількість дефектів у програмі, але це водночас зумовлює збільшення вартості розроблення цього ПЗ [3, 4, 7, 8]. Отже, постає питання у правильному балансуванні витрат на тестування розроблення та проєктування програмного забезпечення. Під час стадії тестування показники надійності є важливими чинниками, які впливають на рішення чи програма буде випущена в експлуатацію. Тому виникає потреба у використанні різних моделей надійності ПЗ, які давали б змогу розраховувати кількісні показники надійності ПЗ.

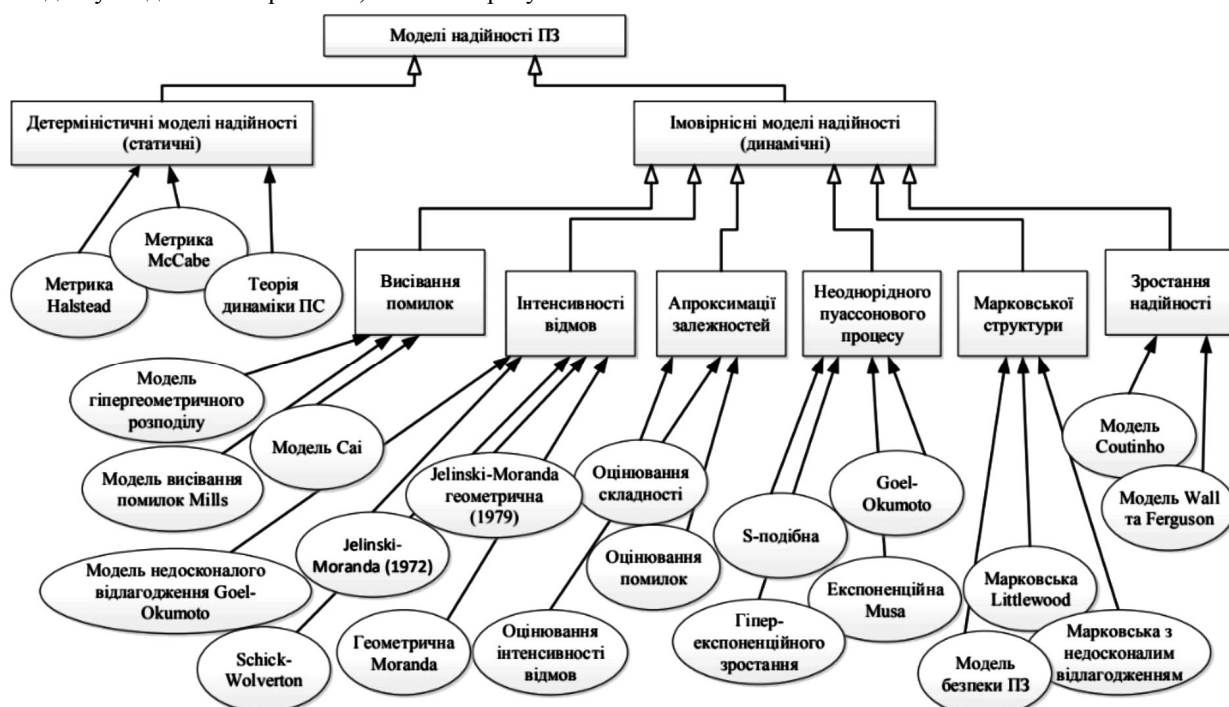


Рис. 1. Схема класифікації моделей надійності ПЗ / Scheme of classification of software reliability models [8]

Моделі, які використовуються для визначення надійності ПЗ, поділяють на два класи: ймовірнісні та детерміністичні [3, 4, 7, 8]. Звичайно, що класифікація тих чи інших моделей може відрізнятися від праці до праці. Можна виділити такі класи моделей: моделі, які засновані на висіванні помилок, апроксимації залежностей, інтенсивності відмов, на підставі неоднорідного пуассонового процесу, на підставі марковської структури та на підставі зростання надійності. Також варто зазначити, що у певних джерелах загальний розподіл моделей зберігається, натомість додаються моделі, які враховують архітектурний підхід, уніфіковані моделі, також моделі з урахуванням джерела походження вхідних даних і т.ін. [8]. До того ж моделі з урахуванням архітек-

тури програмного забезпечення нерідко виділяють в окрему групу, які водночас поділяють на моделі, що засновані на використанні певних архітектурних підходів. Найчастіше згадувані моделі надійності ПЗ проілюстровано на рис. 1 [8].

Моделі на підставі неоднорідного пуассонового процесу розглядають стадію розроблення та відлагодження як процес підрахунку кількості знайдених дефектів, що визначається функцією середнього значення. У разі використання однієї з моделей, які засновані на неоднорідному пуассоновому процесі, визначення показників надійності ПЗ можливе після обчислення функції математичного сподівання відмов. Обчислення параметрів, які використовуються у вибраних моделях, зазвичай

проводиться за використання методу Ньютона, методу максимальної правдоподібності або регресійним методом [8].

Матеріали та методи дослідження. Для верифікації удосконалених моделей надійності ПЗ використано два набори даних, які отримано з архівних джерел компанії – Exoft (<https://exoft.net/>). Перший набір даних характеризує систему, яка використовується для керування даними студентів-практикантів вищих навчальних закладів. Цю систему розробляли у період з 2019 по 2020 роки. Система складається з двох частин: серверної та клієнтської; на стороні сервера використовується трирівнева архітектура, яка дає змогу з легкістю створювати Unit-тести для тестування функціональних частин програми, на стороні клієнта використовується модульна архітектура застосунку, що теж дає змогу проводити тестування незалежних функціональних фрагментів програми. Сумарна кількість рядків коду має становити 62454, в т.ч. як серверну частину (бекенд, 17890 рядків коду), так і частину інтерфейсу користувача (фронтенд, 44565 рядків коду). Кількість протестованих рядків коду становить 57600, або близько 92 % всього коду. Загалом за час тестування, а це 62 робочі тижні, виявлено 831 програмних дефектів. Другий набір даних описує систему генерації маршруту користувача між двома заданими точками на карті, враховуючи наперед визначенні місця, які користувач повинен відвідати. Цю програму було розроблено в період між 2020 та 2021 роками.

Результати дослідження та їх обговорення / Research results and their discussion

Було обрано моделі надійності програмного забезпечення Гоеля-Окумото та S-подібну модель з затримкою [8]. Для підвищення точності роботи моделі запропоновано внести додаткові параметри для моделей, які характеризують вибраний програмний продукт, а саме для підвищення точності оцінки можна використати коефіцієнт покриття коду тестами. Цей коефіцієнт можна обчислювати в такий спосіб: $TLOC/LOC$, де $TLOC$ – кількість відтестованих рядків коду, а LOC – сумарна кількість рядків коду. Враховуючи наведену вище формулу вигляд функції кількості програмних дефектів для моделі Гоеля-Окумото набуде такого вигляду:

$$a = \frac{n}{1 - e^{-bs_n}} \left(2 - \frac{TLOC}{LOC} \right). \quad (1)$$

Припускаючи, що у кодї, який матиме коефіцієнт покриття коду тестами менше за 1, буде виявлятися більше помилок. Для цього у вихідну формулу вносяться зміни, які коригують результуючу функцію, цим самим забезпечуючи приріст кількості помилок на кожну одиницю часу. Ця зміна торкнеться тільки функції, яка визначає кількість помилок у певний момент часу у програмі. Функція, яка відповідає за визначення інтенсивності знаходження програмних дефектів, залишиться не зміненою. Звичайно, на формулу (1) накладені певні обмеження, а саме: сумарна кількість рядків коду повинна бути більша від 0. У разі, коли LOC дорівнює 0, функція не буде мати розв'язків. Але це очікувано, оскільки жодна програма не може бути написана без використання хоча б одного рядка коду. Насправді цю модифікацію можна застосовувати до різних моделей надійності програмного забезпечення, оскільки майже

всі моделі беруть за основу статистичні дані про кількість програмних дефектів, виявлених упродовж часу спостережень [9]. Також варто зазначити, що внесені коефіцієнти не впливають на алгоритмічну складність алгоритму загалом. Оскільки коефіцієнт покриття коду тестами може бути обчислений тільки раз та опісля буде використовуватися як значення-константа при обчисленнях. Якщо внести коефіцієнт покриття коду тестами у S-подібну модель зі затримкою, то отримаємо такий вигляд функції:

$$a = \frac{y_n \left(2 - \frac{TLOC}{LOC} \right)}{1 - (1 + bt_n) e^{-bt}}. \quad (2)$$

Для розрахунку надійності ПЗ складено такий алгоритм роботи, який використовує наведені вище формули (рис. 2).



Рис. 2. Блок-схема алгоритму / Block diagram of the algorithm

Табл. 1. Порівняльні результати роботи програми (набір даних 1) / Comparative results of the program (data set 1)

№ тижня розробки	Знайдена кількість дефектів (дані тестування)	Кількість дефектів, розрахована за моделлю Гоеля-Окумото	Кількість дефектів, розрахована за вдосконаленою моделлю Гоеля-Окумото
55	778	742	800
56	784	755	813
57	790	768	827
58	795	780	841
59	799	793	855
60	800	806	868
61	813	818	882
62	831	831	896

Для автоматизації розрахунку показників надійності розроблено програмне забезпечення, яке дає змогу визначити та спрогнозувати кумулятивну кількість помилок в ПЗ. Під час його розроблення було використано веб-фреймворк Angular 12 (зокрема і 4.2.4), технології HTML, CSS із застосуванням інтерпретатора SCSS. Для проведення математичних обчислень використано web-worker, який є вбудований у фреймворк Angular. Розроблена програма складається з двох окремих застосунків – із серверного застосунка та клієнтського. Розроблення цих застосунків проводилось за використання технології розроблення TDD (test-driven development). Використання цієї технології забезпечило покриття ко-

ду тестами на 100 %. Під час розроблення виявлено 100 програмних дефектів. Сумарна кількість рядків коду становить 16570. Кількість відтестованих рядків становить 16570. Результати використання моделі Гоеля-Окумото та вдосконаленої моделі Гоеля-Окумото наведено в табл. 1. Аналізуючи отримані результати, можна прийти до висновку, що модель Гоеля-Окумото дає незначну похибку приблизно 0,6 %, якщо порівнювати з реальними даними.

Аналізуючи результат використання удосконаленої моделі Гоеля-Окумото, видно, що відхилення від даних,

отриманих шляхом спостережень, дорівнює 8,4 %. За аналогією з попереднім експериментом було проведено верифікацію застосування оригінальної S-подібної моделі зі затримкою (рис. 3,а – (набір даних 1), рис. 3,б – (набір даних 2)). З рисунку видно, що на початкових етапах тестування розрахована кількість помилок дещо нижча за отримані статистичні дані, проте ближче до остаточних результатів бачимо, що розрахована кількість дефектів зростає на певних ділянках відповідно до статистичних даних.

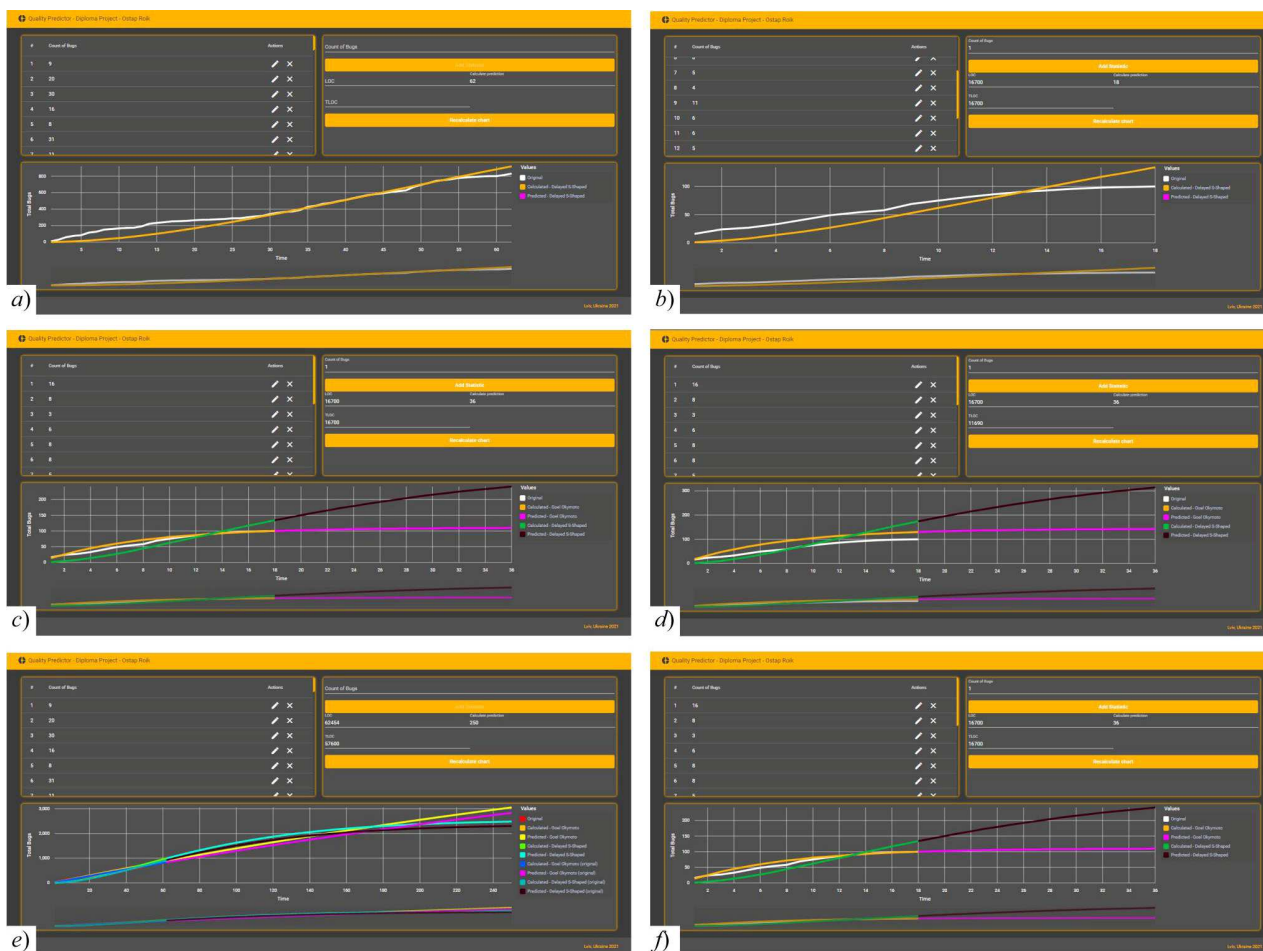


Рис. 3. Результат проведених обчислень за використання S-подібної моделі з затримкою / The result of the calculations using the S-shaped model with a delay: a) набір даних 1 / data set 1; b) набір даних 2 / data set 2. Прогнозована кількість помилок / Estimated number of errors: c) коефіцієнт покриття тестами коду 1 / code test coverage ratio 1; b) коефіцієнт покриття тестами коду 0.7 / code test coverage 0.7. Порівняльний графік прогнозування помилок для різних моделей / Comparative graph of error prediction for different models (e). Результат роботи програми за прогнозування кількості дефектів на проміжок часу, що дорівнює часу, затраченому на розроблення / The result of the program for predicting the number of defects for a period of time equal to the time spent on development (f)

Прогнозування кумулятивної кількості помилок на період з 18 по 28 тижні дасть результат, наведений на рис. 3,с. Аналізуючи дані, отримані шляхом використання удосконаленої S-подібної моделі зі затримкою, видно, що на 28-й тиждень розроблення ПЗ сумарна кількість виявлених помилок становить 204, це прогнозує приріст дефектів у 104 %. Для коефіцієнта покриття коду тестами 0.7, тобто кількість відтестованих рядків коду становитиме 11690, результати використання моделей наведено на рис. 3,д. Як видно з графіка, сумарна кількість знайдених програмних дефектів дещо вища за сумарну кількість дефектів, спрогнозованих за використання коефіцієнта покриття коду тестами, який становить 1 (див. рис. 3,е). Порівняння удосконалених і базових моделей наведено на рис. 3,е. Як неважко помітити,

під час прогнозу надійності на більш тривалий період, найменшу кількість помилок прогнозує удосконалений варіант S-подібної моделі зі затримкою, а найвищу – оригінальна модель Гоеля-Окумото.

Водночас, застосувавши прогнозування на 18 тижнів роботи (набір даних 2), отримаємо результат (рис. 3,ф), з якого можна зробити висновок, що для прогнозування надійності ПЗ для невеликого проекту, в якому коефіцієнт покриття тестами кодом становить 1, краще застосовувати удосконалену модель Гоеля-Окумото. Якщо поглянути на результатуючий графік (див. рис. 3,ф), бачимо, що в період з 18 по 27 тижні буде знайдено 7 дефектів у програмі, а розглядаючи часовий інтервал з 27 по 36 тижні, буде знайдено всього тільки 3 дефекти у програмі. Виходячи з цього, компанії-розробнику не-

має потреби виділяти значні ресурси на пошук дефектів у цьому продукті на пізніх етапах тестування.

Обговорення результатів дослідження. Досліджуючи способи удосконалення наявних моделей визначення надійності програмного забезпечення, можна зазначити, що коефіцієнт покриття коду тестами може істотно вплинути на прогнозоване значення кількості дефектів у програмному продукті, а також варто враховувати коефіцієнт складності проекту. Підсумовуючи наведені вище результати, можна зробити висновок, що удосконалені моделі, на відміну від наявних [2, 5, 6], уможливають з вищим ступенем достовірності проводити прогнозування надійності ПЗ, чим самим надаючи можливість оптимально розподіляти бюджет проекту. Для подальшого удосконалення варто додати до моделі коефіцієнт складності проекту та кількість затраченого часу на тестування розроблюваного продукту, що потенційно допоможе прогнозувати кількість дефектів у ПЗ з більшою точністю. Розроблене програмне забезпечення дає змогу прогнозувати кількість помилок, які залишились в розроблюваному ПЗ з вищим ступенем достовірності внаслідок використання удосконалених моделей надійності ПЗ.

Висновки / Conclusions

Проаналізовано моделі надійності програмного забезпечення з урахуванням процесу недосконалого відлагодження. На підставі проведеного аналізу визначено можливість вдосконалення процесу розрахунку показників надійності ПЗ. Для удосконалення було обрано дві моделі надійності ПЗ, зокрема модель Гоеля-Окумото та S-подібну модель зі затримкою, для яких введено коефіцієнт покриття коду тестами, що уможливило враховувати кількість протестованих рядків коду під час прогнозування зазначених вище показників. На підставі удосконалених моделей було розроблено алгоритм та програмний засіб для автоматизованого розрахунку показників надійності ПЗ. Розроблений програмний продукт дає змогу проводити розрахунок інтенсивності виявлення і кумулятивної кількості помилок, а також

прогнозувати кількість помилок, які залишились в розроблюваному ПЗ, на підставі зібраних статистичних даних про знайдені програмні дефекти та коефіцієнт покриття коду тестами, що дасть змогу підвищити рівень адекватності розрахунку та прогнозування показників надійності ПЗ та оптимальніше розподіляти ресурси проекту (позитивний ефект може становити до 5 % бюджету проекту).

References

1. Bobalo, Yu. Ya., Volochiy, B. Yu., Lozynsky, O. Yu., Mandy, B. A., Ozirkovsky, L. D., Fedasyuk, D. V., Shcherbovsky, S. V., & Yakovyna, V. S. (2013). Mathematical models and methods of analysis of reliability of radio electronic, electrical and software systems: monograph. Lviv: Lviv Polytechnic Publishing House, 300.
2. Goel, A. L., & Okumoto, K. (1979). Time-Dependent Error-Detection Rate Model for Software and other Performance Measures. *IEEE Transactions on Reliability*. Vol. R-28, no 3, 206–211. <https://doi.org/10.1109/TR.1979.5220566>
3. Kharchenko, V. S., Zhikharev, V. Ya., Ilyushko, V. M., Krasnobayev, V. A., Kulikov, P. M., Lysenko, I. V., Nechiporuk, M. V., & Timonkin, G. M. (2004). Fundamentals of reliability of digital systems: textbook. Kharkiv: Nat. aerospace Kharkiv Aviation Institute, 573. [In Ukrainian].
4. Lipaev, V. V. (1998). Reliability of software. Moscow: SINTEG, 232. [In Russian].
5. Mulyak, O. V., Yakovyna, V. S., & Volochiy, B. Yu. (2015). Influence of software reliability models on software-hardware systems reliability indicators. *Eastern European Journal of Advanced Technologies*. Vol. 4, 9(76), 53–57. <https://doi.org/10.15587/1729-4061.2015.47336>
6. Pham, H. (2006). System software reliability. London: Springer-Verlag London Limited, 440. <https://doi.org/10.1007/1-84628-295-0>
7. Polovko, A. M., & Gurov, S. V. (2006). Fundamentals of the theory of reliability. St. Petersburg: BHV-Petersburg, 704. [In Russian].
8. Yakovyna, V. S., & Seniv, M. M. (2020). Fundamentals of software systems reliability theory: textbook. manual. Lviv: Lviv Polytechnic Publishing House, 248.
9. Yakovyna, V. S., Fedasyuk, D. V., Seniv, M. M., & Nytrebych, O. O. (2015). Models, methods and means of analysis of reliability of software systems: monograph. Nat. Lviv Polytechnic University. Lviv: Lviv Polytechnic Publishing House, 220. [In Ukrainian].

M. M. Seniv, O. O. Roik

Lviv Polytechnic National University, Lviv, Ukraine

MEANS OF CALCULATING THE RELIABILITY OF SOFTWARE BASED ON MODELS, TAKING INTO ACCOUNT IMPERFECT DEBUGGING

The analysis of software reliability models is carried out considering the process of imperfect debugging, which resulted in determining that the reliability models do not take into account the indicators that characterize software or its testing process. Proven that during the debugging process there is a possibility of secondary defects in the program when correcting previous defects, the debugging process is supposed not to be perfect. Two software reliability models were selected for improvement, namely the Goel-Okumoto model and the delayed S-shaped model. These models, like the most similar ones, use only statistics collected by tracking the number of defects in the detection program over a period of time. In order to improve existing models, the authors proposed to include additional parameters in the algorithm for calculating the reliability of the software, namely, the coefficient of coverage of the code by tests. The proposed change allows us to observe the trend of changes in the number of detected errors per unit time and an increase in the number of tested lines of code. On the improvement of the model there is a restriction that in the case when the number of tested lines of code is 0, the calculation can not be performed because the actual testing of the code was not performed. On the basis of improved models, algorithms have also been developed that enables the process of calculating and predicting the reliability of programs. The JavaScript programming language and the Angular 12 web framework were chosen for the software implementation of the algorithms for calculating indicators. The program requires the NodeJs 16 code execution environment. The development environment is JetBrains WebStorm 2021.1. To speed up and simplify the development process, we also used the ngx-charts graphics library, which provides ready-made components for displaying graphs in the program. To speed up the algorithms, Service Worker is used, a special tool that allows running operations in a separate thread of program execution, thus enabled us to avoid blocking the user interface during calculations.

Keywords: software engineering; software reliability models; methods of calculating software reliability indicators; Goel-Okumoto model; delayed S-shaped model.