

«досконалих» симетрій гармонізованого простору-часу, відкриває широкі можливості для теоретичних та прикладних досліджень у галузі інформаційних технологій та комп'ютерної техніки. Ідея оптимальних структурних пропорцій є, по суті, новою інтерпретацією унікальних геометричних властивостей реального простору-часу, які відкривають широкі перспективи для розвитку фундаментальних та прикладних досліджень при створенні новітніх обчислювальних систем, мереж та апаратних засобів інформаційної техніки. Структурна досконалість одно- і багатовимірних ІКВ дає змогу знаходити нові практичні застосування не лише в галузі інформаційних технологій, але й в суспільних науках та навчально-освітній сфері на засадах законів гармонії “природи і виховання” [4].

1. *Енциклопедія кібернетики*. – К.: Головна редакція УРЕ, 1973. 2. *Riznyk V.V. Multi-dimensional Systems Based on Perfect Combinatorial Models, IEE, Multidimensional Systems: Problems and Solutions, 1998, pp. 5/1-5/4*. 3. *Семенюк Е.П., Мельник В.П. Філософія сучасної науки і техніки*. – Львів: Світ, 2006. – 152 с. 4. *Вейль Г. Симметрия*. – М.: Наука, 1968. – 192 с.

УДК 004.052; 004.413.5

В. Яковина, М. Сенів, Я. Чабанюк, Д. Федасюк, У. Хімка
Національний університет “Львівська політехніка”,
кафедра програмного забезпечення,
кафедра обчислювальної математики і програмування

КРИТЕРІЙ ДОСТАТНОСТІ ПРОЦЕСУ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

© Яковина В., Сенів М., Чабанюк Я., Федасюк Д., Хімка У., 2010

Побудовано нову математичну модель надійності програмного забезпечення з динамічним показником величини програмного проекту. Проведено порівняльний аналіз існуючих та розробленої моделей на реальних тестових прикладах. Здійснено формалізацію критерію достатності процесу тестування програмного забезпечення на основі запропонованої моделі.

Ключові слова: модель надійності, динамічним показником величини програмного проекту, тестування програмного забезпечення.

The new mathematical model for software reliability with dynamic index of software project size has been developed. The comparative analysis of existing and proposed models using real test examples has been carried out. The criterion of software testing process sufficiency has been formulated on the basis of developed model.

Keywords: model for software reliability, dynamic index of software project size, software testing process.

Вступ

Розвиток інформаційних технологій та комп'ютерної техніки та всебічне проникнення її в усі сфери життєдіяльності людини передбачає постановку нових задач для розробників програмного забезпечення. Програмні продукти стають дедалі складнішими, багатокomпонентними і вимагають спеціалізованого підходу.

За умови досягнення високої надійності сучасна техніка стає ефективною та конкурентоспроможною. Саме від показника надійності похідними будуть інші не менш важливі показники – якість, живучість, безпека, готовність.

У багатьох дослідженнях поняття надійності програмного забезпечення (ПЗ) виділяють окремо [1], тому що при застосуванні понять надійності до програмних засобів варто враховувати особливості і відмінності цих об'єктів від традиційних технічних систем, для яких спочатку розроблялася теорія надійності. Принципова відмінність програм від техніки, та технічних систем зокрема, полягає в тому, що програма не зношується з плином часу, а навпаки, виявляються помилки, яких не було знайдено раніше, ПЗ з часом вдосконалюється і покращується.

Водночас підвищуються і вимоги до надійності та витривалості програм, виникає потреба у скороченні затрат на тестування та, відповідно, у прогнозуванні надійності розроблюваного програмного забезпечення.

Для розв'язання таких задач оцінювання та прогнозування надійності сьогодні використовують моделі надійності ПЗ [1–3]. Однак усі перелічені моделі містять цілий ряд спрощень і припущень, що зменшує клас задач та область застосування їх для реального об'єкта.

Тому сьогодні актуальною задачею програмної інженерії є розроблення моделей надійності ПЗ з підвищеним ступенем адекватності реальним об'єктам.

Аналіз існуючих моделей надійності ПЗ

Загальне визначення надійності ПЗ наведено в [4]. Нехай F – довільно визначений клас помилок, а T – відносна одиниця часу, одиниці вимірювання якої визначаються конкретною програмою, надійність якої оцінюється. Тоді надійністю програмного пакета відносно класу помилок F і відносно метрики часу T є імовірність того, що жодної помилки цього класу не виникне протягом виконання програми за наперед заданий відносний період часу.

Сучасні підходи до оцінювання надійності ПЗ засновані на паралелях з надійністю апаратного забезпечення з урахуванням принципів відмінностей між програмним та апаратним забезпеченням [4]. Наприклад, апаратне забезпечення характеризується як спаданням, так і зростанням інтенсивності виявлення помилок. Спадання інтенсивності виявлення помилок пов'язано з етапом прироблення (виявлення і усунення технологічного браку) і, крім того, з виявленням і усуненням помилок, внесених на етапі проектування. Зростання інтенсивності виявлення помилок насамперед пов'язано зі зношуванням та старінням компонентів апаратного забезпечення. Натомість у випадку програмного забезпечення відсутня така характеристика, як старіння. ПЗ може морально застарівати внаслідок змін вимог користувача або комп'ютерного оточення (апаратного забезпечення, операційної системи тощо), однак відколи програма модифікується для того, щоб задовольняти ці зміни, це вже буде не те саме ПЗ, а модифікована або покращена версія. Так само, як і апаратне забезпечення, ПЗ характеризується спаданням інтенсивності виявлення помилок (покращанням якості) у міру того, як зростає час використання системи, і помилки, наприклад дизайну чи коду, виправляються. Слід зазначити, що визначене значення міри надійності ПЗ завжди є відносним до заданого середовища використання. Так, наприклад, різні користувачі, що використовують різні гілки виконання того самого програмного продукту, найімовірніше, отримають різні значення надійності програми.

За останні 30 років було запропоновано багато аналітичних моделей для вирішення проблеми вимірювання надійності ПЗ. Ці підходи ґрунтуються переважно на історії спостереження помилок ПЗ і можуть бути класифіковані згідно з процесом дослідження помилок так [4]:

- *Моделі на основі часу між помилками.* У цьому класі моделей предметом дослідження є час між виявленнями помилок. Найпоширенішим припущенням є те, що час між $(i-1)$ -ю та i -ю помилками підлягає певному виду розподілу, параметри якого залежать від кількості помилок, що залишаються в програмі протягом цього інтервалу. Оцінки параметрів отримуються на основі спостережених часових інтервалів між помилками, а оцінки надійності ПЗ (середній час до наступної помилки тощо) потім отримують на основі моделі з визначеними параметрами. Інший підхід трактує час між помилками як реалізацію стохастичного процесу і використовує відповідні моделі часових рядів для опису процесу виявлення помилок, що покладено в основі цього процесу.

- *Моделі на основі кількості помилок.* Предметом дослідження цього класу моделей є кількість помилок у визначеному часовому інтервалі, а не час між помилками. Вважають, що кількість помилок відповідає відомому стохастичному процесу з дискретною або неперервною інтенсивністю виявлення

помилки, що залежить від часу. Параметри функції інтенсивності виявлення помилок можна оцінити на основі спостережених значень кількості помилок чи з часів між помилками. Оцінку надійності ПЗ, як і в попередньому випадку, можна отримати з відповідних рівнянь.

- *Моделі на основі висівання помилок.* Основним підходом цього класу моделей є "висівання" відомої кількості помилок в програму, яка, вважається, має невідому кількість власних помилок. Після цього програма тестується і підраховується виявлена кількість висіяних та власних помилок. На основі цього отримують оцінку кількості помилок в програмі до висівання, яка використовується для отримання оцінок надійності ПЗ та інших відповідних характеристик.

- *Моделі на основі області вхідних даних.* Основним підходом є генерування набору тестових прикладів з вхідного розподілу, який, вважається, репрезентує цільове використання програми. Через складність отримання такого розподілу, область вхідних даних розділяють на набір класів еквівалентності, кожен з яких зазвичай пов'язують з одним зі шляхів виконання програми. Оцінку надійності ПЗ отримують на основі помилок, виявлених шляхом фізичного чи символічного виконання тестових прикладів, взятих з області вхідних даних.

Широко використовуються моделі на основі кількості помилок [2, 4, 5]. Цей клас моделей пов'язаний з кількістю виявлених помилок на заданому інтервалі тестування. У міру усунення дефектів з системи очікується, що виявлена кількість помилок в одиницю часу спадатиме. В такому випадку залежність кумулятивної кількості помилок від часу виходитиме на насичення. Слід зазначити, що під часом в цьому випадку розуміють календарний час, процесорний час, кількість виконаних тестових прикладів чи деяку іншу відповідну метрику.

Було запропоновано багато моделей, що описують такі явища. В основу більшості таких моделей покладено розподіл Пуассона, параметри якого мають різний вигляд для різних моделей, оскільки використання такого розподілу випадкових величин добре зарекомендувало себе в багатьох областях, де основна зацікавленість полягає в кількості подій [4].

До таких моделей належать моделі: Shooman [6], Jelinski–Moranda [7], Schick–Wolverton [8], Musa [9], Schneidewind [10], Goel–Okumoto [11], S-подібна модель зростання надійності [12] тощо. Крім того нещодавно було представлено "узагальнену модель негомогенного пуассонівського процесу" [13]. Для підвищення точності цієї моделі автори [13] пропонують використовувати таку форму кривої інтенсивності виявлення несправностей, у якій введено додатковий параметр n для оцінювання величини проекту, де вибір параметра n залежить від процесу проведення тестування з такими рекомендованими значеннями:

$n = 0$ – для невеликого проекту, в якому розробник є одночасно і тестером (моделі Musa, Goel–Okumoto і Schneidewind);

$n = 1$ – для середнього проекту, в якому тестування і проектування ПЗ виробляється різними людьми з однієї робочої групи (S-подібна модель);

$n = 2$ – для великого проекту, в якому групи тестування і розроблення ПЗ працюють над проектом паралельно;

$n = 3$ – для дуже великого проекту, в якому відділи тестування і розроблення незалежні.

Усі ці моделі ґрунтуються на таких основних припущеннях [5–13]:

- Кількість помилок на момент t , $M(t)$ відповідає пуассоновому процесу з функцією математичного сподівання $\mu(t)$. При цьому очікувана кількість помилок, які буде виявлено за проміжок часу $t + \Delta t$ пропорційна кількості несправностей, що залишилися у системі на момент t . Також вважається, що $\mu(t)$ – обмежена, не спадаюча функція часу: $\lim_{t \rightarrow \infty} \mu(t) = N < \infty$. Отже, ця модель належить до категорії скінчених функцій.

- Кількості помилок (f_1, f_2, \dots, f_n) , виявлених на відповідних часових інтервалах $[(t_0 = 0, t_1), (t_1, t_2), \dots, (t_{n-1}, t_n)]$, незалежні для будь-якої скінченної кількості часових проміжків $t_1 < t_2 < \dots < t_n$.

- Імовірності виявлення будь-якої помилки на будь-якій ітерації є однаковими.

- При виявленні помилки з програмного коду видаляється тільки один дефект без уведення нових.

Аналітичний вигляд функцій інтенсивності виявлення несправностей ($\lambda(t)$) та функції кумулятивної кількості несправностей ($\mu(t)$) основних моделей наведено в табл. 1. (Зауважимо, що

$$\lambda(t) = \frac{d\mu(t)}{dt} \quad [1]).$$

Таблиця 1

Функції інтенсивності виявлення несправностей основних моделей надійності ПЗ

Модель	Функція інтенсивності виявлення несправностей	Функція кумулятивної кількості несправностей
Goel–Okumoto	$\lambda(t) = Nb \exp(-bt)$	$\mu(t) = N(1 - e^{-bt})$
Schneidewind	$\lambda(t) = \alpha_0 \exp(-\beta t)$	$\mu(t) = \frac{\alpha_0}{\beta} (1 - e^{-\beta t})$
Базова модель Musa	$\lambda(t) = \beta_0 \beta_1 \exp(-\beta_1 t)$	$\mu(t) = \beta_0 (1 - e^{-\beta_1 t})$
S-подібна модель	$\lambda(t) = \alpha \beta^2 t \exp(-\beta t)$	$\mu(t) = \alpha (1 - (1 + \beta t) e^{-\beta t})$
Узагальнена модель негомогенного пуассонівського процесу	$\lambda(t) = \alpha \beta^{n+1} t^n \exp(-\beta t)$	$\mu(t) = \alpha \left(n! - \sum_{i=0}^n \frac{n! \beta^{n-i}}{(n-i)!} t^{n-i} e^{-\beta t} \right)$

Неважко помітити, що параметри моделей, наведених в табл. 1, пов'язані такими співвідношеннями:

$$N = \alpha, \quad \beta_0 = \alpha, \quad b = \beta, \quad \beta_1 = \beta, \quad \alpha_0 = \alpha \beta,$$

де α – загальна кількість несправностей під час спостережень, β – деякий коефіцієнт.

Основним недоліком моделей [6–12] є те, що внаслідок припущень і спрощень вони не достатньо повно відображають процес тестування, а їх результати не завжди збігаються з практичними [5, 13]. Основними недоліками моделі [13], на нашу думку, є відсутність формалізації параметра величини проекту n ; його апріорне встановлення, незалежно від реальних експериментальних даних тестування ПЗ; значення параметра n з діапазону цілих чисел, що не дасть змоги описувати поведінку реального ПЗ, яке за своєю величиною і складністю має скоріше неперервний ніж дискретний характер. Крім того, оскільки значення n не є параметром розподілу, модель [13] фактично не є однією взаємопов'язаною моделлю, а набором з чотирьох моделей, що відрізняються значенням n .

Основним завданням роботи є формалізація критерію достатності процесу тестування ПЗ з погляду його надійності та наявності дефектів. Такий критерій повинен слугувати кількісною мірою при прийнятті рішення про достатність етапу тестування при розробленні програмних продуктів. Цей критерій має ґрунтуватися на моделі надійності, побудова якої для неоднорідного пуассонового процесу з підвищеним ступенем адекватності реальним об'єктам шляхом уведення і формалізації неперервного індексу складності проекту є складовою частиною роботи. Крім того, для визначення ступеня точності відображення властивостей реального ПЗ розробленою моделлю буде описано поведінку надійності ПЗ на основі експериментальних даних та порівняно його з існуючими моделями.

Побудова моделі оцінювання та прогнозування надійності ПЗ з індексом величини проекту

Припустимо, що час виявлення помилок у моделі оцінювання та прогнозування надійності ПЗ розподілений за законом Пуассона. Крім того, вважаємо, що індекс величини проекту є параметром моделі визначається на основі експериментальних даних і набуває значення з реального діапазону і завжди більший від нуля.

Пропонується такий вигляд функції інтенсивності виявлення несправностей:

$$\lambda(t) = \alpha \beta^{s+1} t^s \exp(-\beta t), \quad (1)$$

де α – коефіцієнт, що визначає загальну кількість помилок в ПЗ, β – коефіцієнт, що характеризує загальну тривалість процесу виявлення помилок, s – індекс величини проекту, що узагальнює S-подібну модель.

Для інтенсивності (1) функція кумулятивної кількості несправностей має вигляд:

$$\mu(t) = \int_0^t \lambda(\tau) d\tau = \alpha \left[-\beta^s t^s e^{-\beta t} + s \Gamma_{\beta^s}(s) \right], \quad (2)$$

де $\Gamma_z(p) = \int_0^z t^{p-1} e^{-t} dt$, ($\text{Re } p > 0$), – неповна гамма-функція. Зауважимо, що при $s=1$ функція інтенсивності виявлення несправностей (1) та кумулятивна функція (2) збігаються з виглядом відповідних функцій S-подібної моделі.

Загальна кількість помилок в ПЗ визначається кумулятивною функцією при $t \rightarrow \infty$, тобто:

$$\mu(\infty) = \alpha s \Gamma(s), \quad (3)$$

де $\Gamma(s)$ – гамма-функція.

Отже, аналітичний вигляд побудованої моделі дає змогу узагальнити вираз для загальної кількості помилок в системі (3), яка залежить від величини та складності проекту і визначається параметрами моделі. Крім того, частковий випадок при $s=1$ (S-подібна модель) з урахуванням того, що $\Gamma(1)=1$, дає значення $\mu(\infty) = \alpha$, що відповідає S-подібній моделі.

Рівняння (1) та (2) будемо називати моделлю з індексом величини проекту.

Під величиною проекту розуміємо комплексний показник, який корелює з метриками складності коду програмного продукту [14]. Встановлення залежності між індексом величини проекту та метриками складності коду є предметом подальших досліджень.

Проведені попередні дослідження дають змогу встановити такі інтервали для індексу величини проекту:

- при $s \in (0; 0,7)$ проект можна вважати невеликим,
- при $s \in [0,7; 1,5)$ проект можна вважати середньої величини,
- при $s \in [1,5; 2,2)$ проект можна вважати великим,
- при $s \in [2,2; e]$ – дуже великим.

Однак остаточне вирішення цього питання, так само як і отримання аналітичного виразу для індексу величини проекту зі статистичною достовірністю, потребує подальших досліджень із залученням множини експериментальних даних для різних типів і класів ПЗ.

Для визначення параметрів α , β , s побудованої моделі застосуємо метод максимальної правдоподібності.

Нехай на проміжках $(t_i, t_{i+1}]$, $i = \overline{0, n}$ виявлено m_i помилок ($\sum_{i=1}^n m_i = k$). Припустивши, що час виявлення помилок розподілений за Пуассоном, побудуємо функцію $L(\alpha, \beta, s)$ максимальної правдоподібності:

$$L(\alpha, \beta, s) = \prod_{i=1}^n \frac{[\mu(t_i) - \mu(t_{i-1})]^{m_i}}{m_i!} \exp(\mu(t_{i-1}) - \mu(t_i))$$

Враховуючи вигляд $\mu(t)$ з рівняння (2), маємо:

$$L(\alpha, \beta, s) = \prod_{i=1}^n \frac{\alpha^{m_i}}{m_i!} \left[s (\Gamma_{\beta^s}(s) - \Gamma_{\beta^{s-1}}(s)) + \beta^s (t_{i-1}^s \exp(-\beta t_{i-1}) - t_i^s \exp(-\beta t_i)) \right]^{m_i} \exp(\mu(t_{i-1}) - \mu(t_i)).$$

Функція правдоподібності L задовольняє умови:

1. Диференційована при довільних значеннях вибірки;
2. Досягає максимуму в інтервалі можливих значень.

Для отримання оцінок α, β, s розв'язуємо систему рівнянь:

$$\begin{cases} \frac{\partial L(\alpha, \beta, s)}{\partial \alpha} = 0, \\ \frac{\partial L(\alpha, \beta, s)}{\partial \beta} = 0, \\ \frac{\partial L(\alpha, \beta, s)}{\partial s} = 0. \end{cases} \quad (4)$$

Без зменшення загальності замість функції L у системі (4) розглядаємо функцію $\ln L$ і після деяких елементарних перетворень отримаємо систему рівнянь:

$$\begin{cases} \alpha = \frac{k}{s\Gamma_{\beta t_n}(s) - \beta^s t_n^s e^{-\beta t_n}}; \\ \sum_{i=1}^n \frac{m_i (t_i^{s+1} e^{-\beta t_i} - t_{i-1}^{s+1} e^{-\beta t_{i-1}})}{s\Phi_{\beta t_{i-1}}(s) + \beta^s (t_{i-1}^s e^{-\beta t_{i-1}} - t_i^s e^{-\beta t_i})} - \frac{k t_n^{s+1} e^{-\beta t_n}}{s\Gamma_{\beta t_n}(s) - \beta^s t_n^s e^{-\beta t_n}} = 0; \\ \sum_{i=1}^n \frac{m_i}{s\Phi_{\beta t_{i-1}}(s) + \beta^s (t_{i-1}^s e^{-\beta t_{i-1}} - t_i^s e^{-\beta t_i})} \left[\Phi_{\beta t_{i-1}}(s) + sF_{\beta t_{i-1}}(s) + \beta^s (t_{i-1}^s e^{-\beta t_{i-1}} \ln(\beta t_{i-1}) - t_i^s e^{-\beta t_i} \ln(\beta t_i)) \right] + \\ + \frac{k}{s\Gamma_{\beta t_n}(s) - \beta^s t_n^s e^{-\beta t_n}} \left[\beta^s t_n^s e^{-\beta t_n} \ln(\beta t_n) - \Gamma_{\beta t_n}(s) - s \int_0^{\beta t_n} \tau^{s-1} \ln \tau e^{-\tau} d\tau \right] = 0; \end{cases} \quad (5)$$

де $\Phi_{\beta t_{i-1}}(s) = \int_{\beta t_{i-1}}^{\beta t_i} \tau^{s-1} \exp(-\tau) d\tau$, $F_{\beta t_{i-1}}(s) = \int_{\beta t_{i-1}}^{\beta t_i} \tau^{s-1} \ln \tau \exp(-\tau) d\tau$.

Систему трансцендентних рівнянь (5) можна розв'язати чисельно (наприклад, модифікованим методом Ньютона), а отримані наближені значення $\hat{\alpha}, \hat{\beta}, \hat{s}$ можна використати для аналізу достатності тестування, величини проекту, надійності програмного продукту.

Зауваження 1. Першим етапом алгоритму розв'язку задачі побудови функції правдоподібності L є встановлення інтервалів $(t_i, t_{i+1}]$, $i=0, n$, на яких кількості помилок m_i мають пуассонів розподіл [5].

Зауваження 2. Встановлення властивостей точкових оцінок $\hat{\alpha}, \hat{\beta}, \hat{s}$, таких як правдивість, ефективність, незміщеність і конзистентність є предметом подальших досліджень.

Дослідження моделі на результатах експериментальних даних тестування ПЗ

У роботі [5] було проведено ряд програмних експериментів з метою перевірки належності поведінки зростання надійності ПЗ до неоднорідного пуассонового процесу. Як емпіричні спостереження, так і тестування статистичних гіпотез показали, що поведінка надійності ПЗ не відповідає неоднорідному пуассоновому процесу загалом і не описується моделлю Goel–Okumoto зокрема [5].

У цій частині статті представлені результати опису поведінки надійності ПЗ побудованою моделлю, моделлю Goel–Okumoto та S-подібною моделлю.

Для коректного порівняння результатів застосування побудованої моделі з відомими моделями надійності ПЗ було взято емпіричні дані з першого експерименту роботи [5].

Дослідження проводили так, щоб імітувати використання моделі надійності на етапі тестування проектів з розроблення програмного забезпечення:

- етап тестування з 1200 ітерацій розбивали на фази, у нашому випадку по 50 ітерацій;
- після кожних 50 ітерацій тестування методом Ньютона з використанням пакета Mathcad 14.0 розв'язували систему рівнянь (5) на інтервалі $(0, t_i]$ і отримували точкові оцінки параметрів $\hat{\alpha}$, $\hat{\beta}$ та \hat{s} . Отримані наближені значення підставляли у вираз (2) і будували залежність $\mu(t)$ на інтервалі $(t_{i-1}, t_i]$;

- після побудови залежності $\mu(t)$ на усьому проміжку експериментальних даних розраховували коефіцієнт кореляції та середньоквадратичну похибку для експериментальних значень і значень, отриманих з моделі.

Такі дослідження проводились для моделі (2), моделі Goel–Okumoto та S-подібної моделі (див. табл. 1).

Квадрат коефіцієнта кореляції між експериментальною та обчисленою кумулятивною кількістю помилок (R^2) та середня квадратична похибка ($\Delta^2 = \sum_i (\mu_{\text{exp}}(t_i) - \mu_{\text{calc}}(t_i))^2 / n$, тут $\mu_{\text{exp}}(t_i)$ – експериментальне значення кумулятивної кількості помилок в момент часу t_i , $\mu_{\text{calc}}(t_i)$ – розраховане з параметрів відповідної моделі значення кумулятивної кількості помилок в момент часу t_i) наведено в табл. 2.

Таблиця 2

Параметри опису експериментальних даних дослідженими моделями

Модель	R^2	Δ^2
Модель Goel–Okumoto	0,993	0,12
S-подібна модель	0,995	0,18
Модель з індексом складності проекту (2)	0,999	0,05

Дані табл. 2 показують, що запропонована модель з індексом складності проекту значно краще описує реальні експериментальні результати: так, квадрат коефіцієнта кореляції між експериментальними даними і моделлю з індексом проекту становить 0,999, тоді як для моделі Goel–Okumoto – 0,993, а для S-подібної моделі – 0,995; середнє значення квадрата відхилень між обчисленими та експериментальними значеннями становить 0,05 для моделі (2), що майже на порядок краще, ніж двох інших моделей – 0,12 для моделі Goel–Okumoto та 0,18 для S-подібної моделі. Крім того, отримані результати показують, що загалом S-подібна модель краще описує експериментальні дані, ніж модель Goel–Okumoto.

Розрахована за рівнянням (3) загальна кількість помилок в програмному продукті з використанням отриманих для усього етапу тестування параметрів моделі (2) становить 35,1, тоді як за даними [5] в цей програмний продукт було впроваджено 38 помилок. Тобто модель надійності ПЗ з індексом складності проекту придатна до використання на підприємстві для оцінювання загальної кількості помилок та прогнозування частоти їх появи. Отримана оцінка індексу величини проекту s для усього етапу тестування становить 0,085, що дає змогу вважати програму невеликою, що підтверджується даними роботи [5] – програма складалась з 9564 рядків коду мовою C, з яких 6218 виконуваних.

На рис. 1 зображено графік залежності кумулятивної кількості помилок від часу (точки – експериментальні дані з [5], крива 1 – модель з індексом складності проекту (2), крива 2 – модель Goel–Okumoto, крива 3 – S-подібна модель). На вставці зображено у більшому масштабі ті самі залежності на початкових ітераціях тестування.

Як видно з цього рисунка, на пізніх етапах тестування усі моделі однаково добре описують експериментальні дані, що узгоджується з результатами роботи [5]. Однак на початкових етапах тестування модель Goel–Okumoto, як і S-подібна модель, незадовільно описують дані тестування реального ПЗ, на відміну від запропонованої моделі з індексом складності проекту, що підтверджується даними табл. 2.

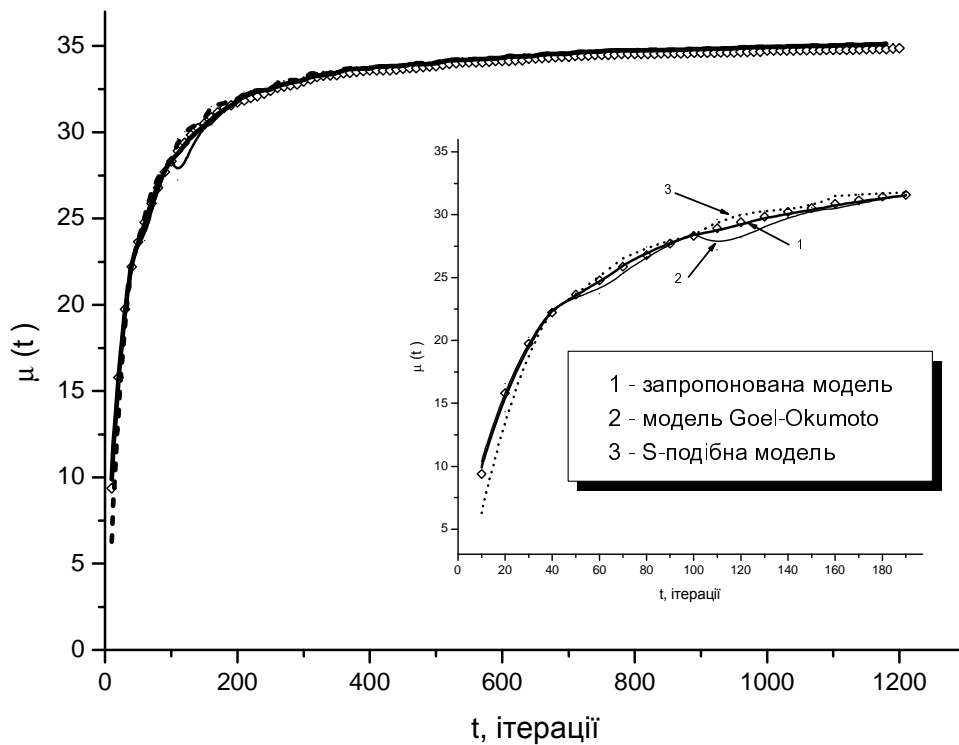


Рис. 1. Залежність експериментальної та розрахованої кумулятивної кількості помилок від часу

Найяскравіше відмінності між цими трьома моделями проявляються на залежностях функції інтенсивності виявлення помилок від часу, які наведені на рис. 2 (крива 1 – модель з індексом складності проекту, крива 2 – модель Goel–Okumoto, крива 3 – S-подібна модель).

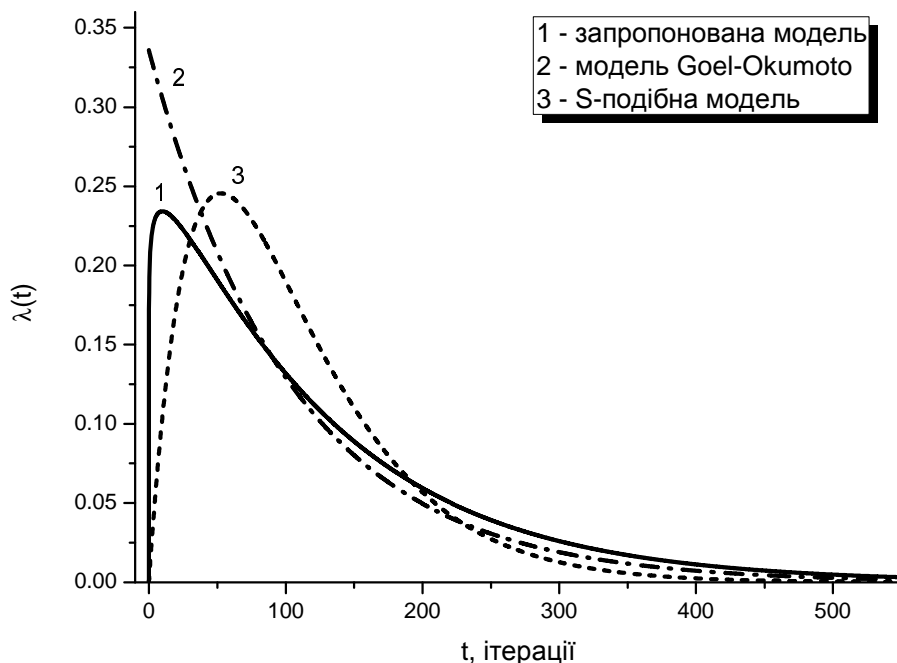


Рис. 2. Залежність інтенсивності виявлення помилок від часу для різних моделей

Як видно з рис. 2, як модель Goel–Okumoto, так і S-подібна модель показують завищені значення інтенсивності виявлення помилок при малих значеннях часу, і навпаки – занижені значення при великих t , вказуючи на більш раннє завершення процесу виявлення помилок, ніж

спостерігається в реальних експериментальних даних. Крім того, модель Goel–Okumoto неадекватно описує початковий етап тестування (рис. 2). Своєю чергою, S-подібна модель найгірше описує середину процесу тестування (у нашому випадку приблизно від 50 до 175 ітерації).

Важливою характеристикою надійності є частота відмов (у випадку програмного забезпечення коректнішим буде вживати термін “частота помилок”, оскільки не усі помилки ПЗ приводять до його відмови), яка за означенням є щільністю імовірності випадкової величини T і визначається як [1]:

$$f(t) = \lambda(t) e^{-\mu(t)}.$$

Графік залежності частоти помилок від часу для трьох розглянутих моделей надійності наведено на рис. 3.

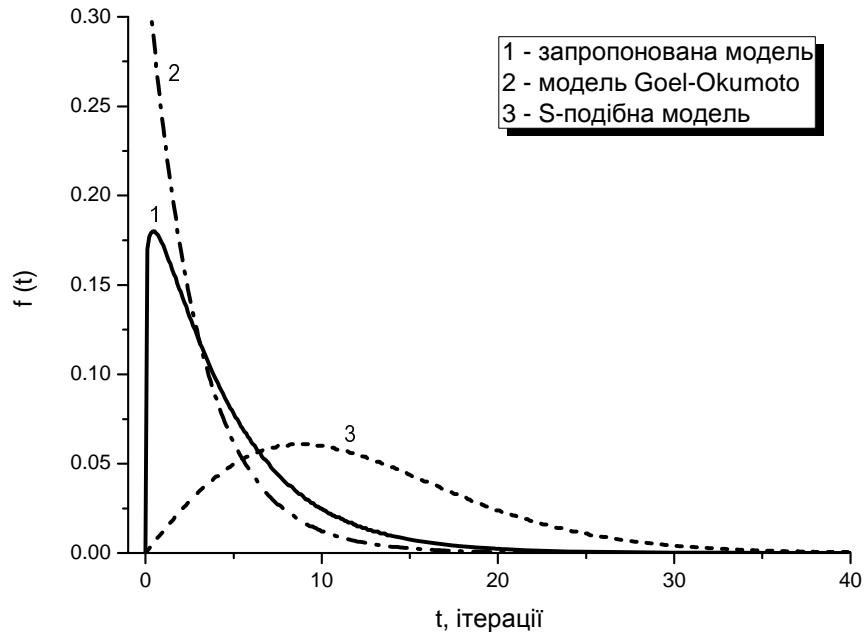


Рис. 3. Залежність частоти помилок від часу для різних моделей

Як і у випадку функції інтенсивності виявлення помилок, залежність частоти помилок (рис. 3) дозволяє зробити аналогічні висновки. Модель Goel–Okumoto (крива 2 рис. 3) показує, що частота помилок є аномально високою в початкові моменти часу і аномально швидко загасає з часом, що не відповідає практичним результатам тестування ПЗ. Натомість S-подібна модель (крива 3 рис. 3) показує занижену максимальну частоту помилок при повільнішому її загасанні з часом.

Як бачимо з рис. 2 та 3, запропонована модель усуває недоліки обох попередніх моделей шляхом уведення реального показника величини проекту і завдяки цьому адекватніше описує залежності характеристик надійності реальних програмних продуктів.

Опис і формалізація критерію достатності процесу тестування ПЗ

Важливим прикладним аспектом моделей надійності ПЗ може стати встановлення кількісного критерію достатності процесу тестування програмного продукту, який би дав змогу керівникам програмних проектів більш обґрунтовано приймати рішення про виділення ресурсів на тестування та про завершення цього етапу розробки ПЗ. Так, сьогодні у переважній більшості ІТ компаній такі показники мають скоріше якісний та неформалізований характер на зразок "задоволення замовника", які жодним чином не можна використовувати, наприклад, розробляючи ПЗ відповідального призначення.

Для визначення критерію достатності процесу тестування було побудовано залежності параметрів трьох досліджуваних моделей надійності залежно від ітерацій, на яких було припинено тестування (рис. 4–6). На рис. 4, 5 криві 1 зображають відповідні параметри моделі з індексом складності проекту, криві 2 – параметри моделі Goel–Okumoto, криві 3 – параметри S-подібної моделі. Побудова таких залежностей відповідає практиці проведення тестування, коли здійснюється певна ітерація (цикл тестування), після якої оцінюється якість програмного продукту.

Пропонується після кожного такого циклу описувати отримані результати моделлю надійності, а параметри моделі зображати графічно чи в таблиці.

Як видно з рис. 4, параметр α для запропонованої та S-подібної моделей чи N для моделі Goel–Okumoto не дає змоги визначити зазначений критерій достатності процесу тестування, оскільки його залежності від ітерацій тестування є практично гладкими функціями без особливостей. Особливість залежності параметра α для запропонованої моделі (рис. 4, крива 1) в околі 550 ітерацій добре корелює з висновками, які можна зробити з аналізу результатів роботи [5] – після 500–600 ітерацій час виявлення помилок реального ПЗ розподілений за Пуассоном, тоді як в області менших значень часу тестування такої гіпотези дає негативний результат. Отже, запропонована модель, крім іншого, дає змогу встановити точку переходу до пуассонового розподілу часу виявлення помилок і так встановити область експериментальних результатів, які можуть з високим ступенем достовірності бути описаними моделлю надійності на основі такого розподілу.

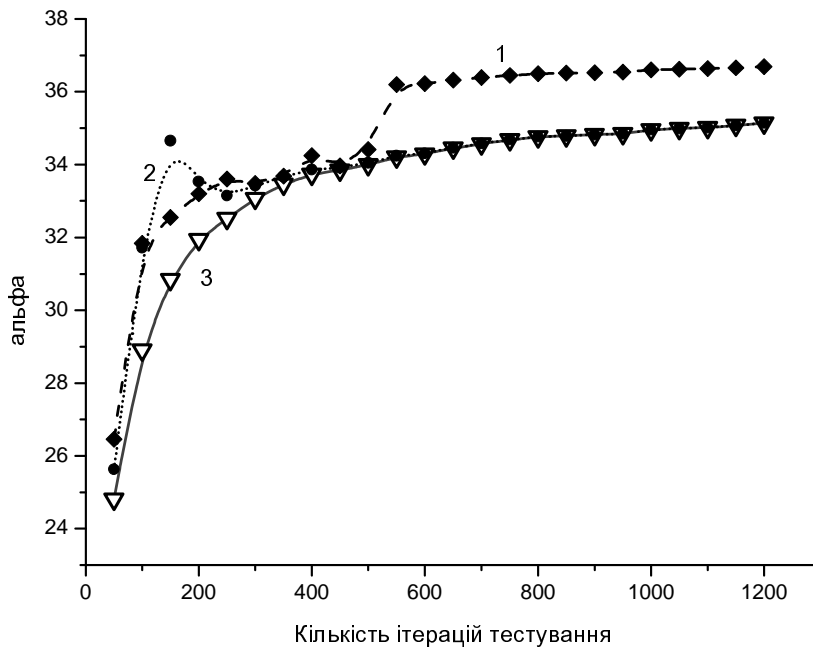


Рис. 4. Залежність параметра, що визначає загальну кількість помилок (α та N відповідно) для моделей надійності від ітерацій, на яких було припинено тестування

На рис. 5 показано, що параметр β для запропонованої та S-подібної моделей чи b для моделі Goel–Okumoto так само не може бути основою для визначення критерію достатності процесу тестування, оскільки його залежності від ітерацій тестування є практично гладкими функціями без особливостей. У цьому випадку практично відсутня навіть будь-яка особливість залежності параметра β для запропонованої моделі в околі 550 ітерацій на відміну від параметра α (рис. 4). Хоча слід зазначити, що при більших значеннях часу крива є більш гладкою і не виявляє ознак осциляцій, характерних для значень часу, менших за 500 ітерацій (рис. 5, крива 1).

На противагу до параметрів α та β , залежність індексу величини проекту s (який відсутній в усіх інших моделях на основі розподілу Пуассона) виявляє чітку особливість, яку можна покласти в основу критерію достатності процесу тестування (рис. 6). Ця особливість полягає в тому, що при $t \geq 550$ (що знову таки відповідає переходу до пуассонового розподілу часу виявлення помилок) залежність $s(t)$ стає гладкою, а значення s наближається до постійної величини, на відміну від такої залежності при $t < 550$, яка проявляє різко осцилюючий характер.

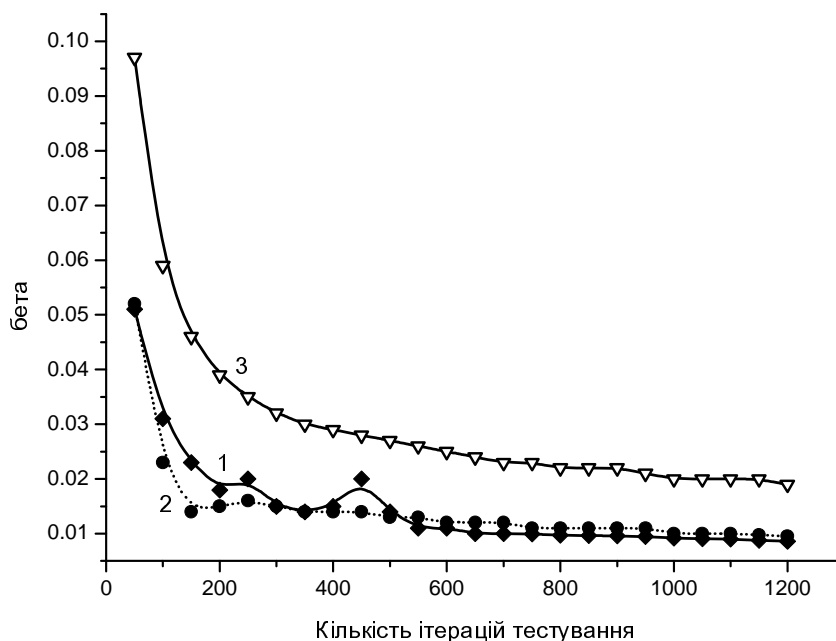


Рис. 5. Залежність параметра, що визначає тривалість процесу виявлення помилок (β та b відповідно) для моделей надійності від ітерацій, на яких було припинено тестування

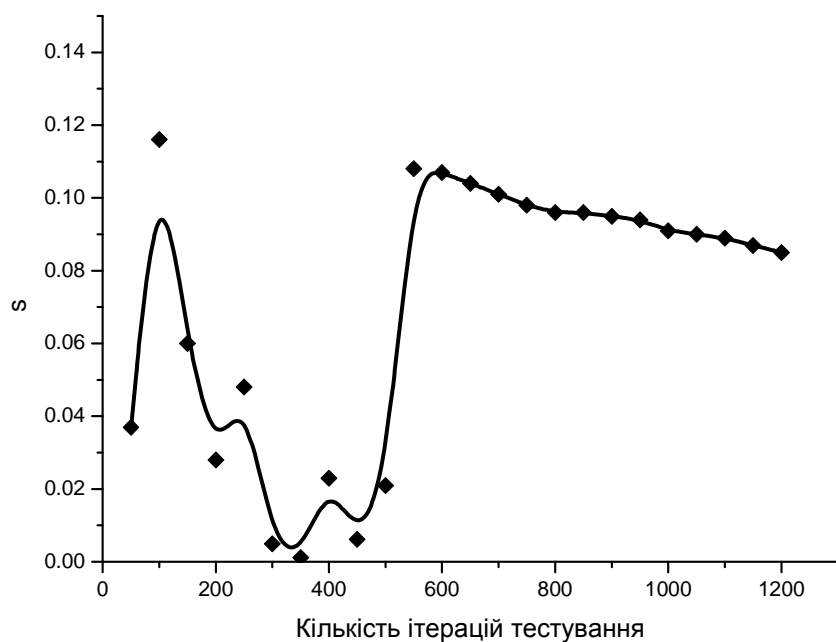


Рис. 6. Залежність індексу величини проекту s запропонованої моделі від ітерацій, на яких було припинено тестування

Таку поведінку залежності $s(t)$ можна зрозуміти, врахувавши, що індекс величини проекту (параметр s моделі) є основним параметром, що визначає форму і функцію розподілу, а відповідно і щільність ймовірності випадкової величини, яка у нашому випадку є часом виявлення помилки. Отже, на пізніх етапах тестування ПЗ, коли корельовані помилки виявлено та усунено, а час виявлення тих помилок, що залишились, відповідає пуассоновому розподілу, якісна характеристика розподілу

(параметр s) вже практично не змінюється, а змінюються переважно кількісні характеристики (параметри α та β), що дає змогу формалізувати критерій достатності процесу тестування ПЗ так:

$$\frac{ds(t)}{dt} \rightarrow 0. \quad (6)$$

Ще раз зауважимо, що усі інші моделі надійності ПЗ на основі розподілу Пуассона не дають змоги отримати такий критерій внаслідок відсутності необхідних параметрів моделей, які б описували якісну зміну форми розподілу. Залежність функції (6), отриману шляхом числового диференціювання даних (рис. 6), наведено на рис. 7. Як бачимо, така залежність наочно ілюструє критерій достатності процесу тестування ПЗ (6) і показує різку зміну форми кривої при переході експериментальних даних до пуассонового розподілу. Отже, з використанням критерію достатності процесу тестування можна визначити загальну кількість помилок у програмному продукті за допомогою рівняння (3) і, порівнявши її з кількістю вже виявлених та виправлених помилок, прийняти обґрунтоване рішення про розподіл ресурсів проекту зі створення програмного продукту.

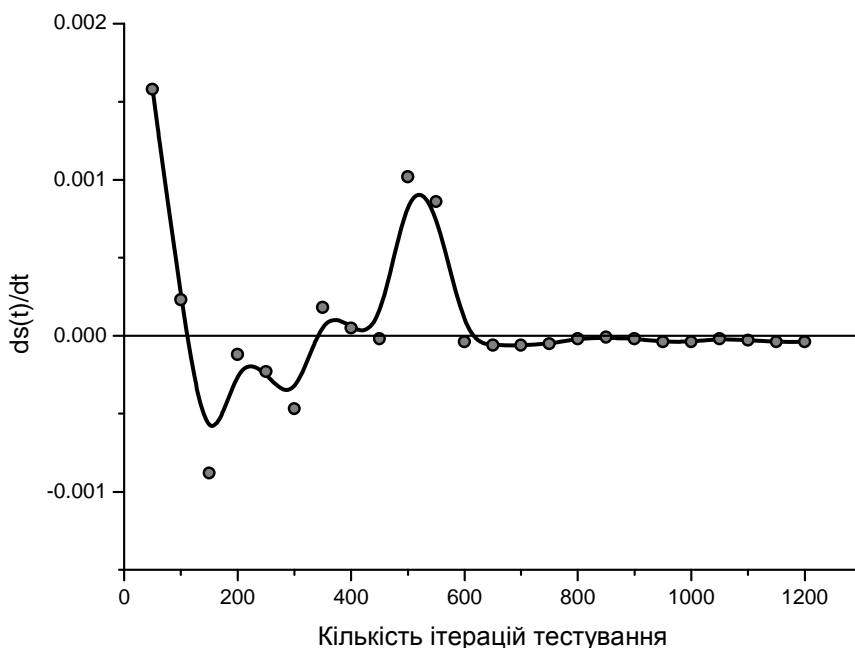


Рис. 7. Залежність критерію достатності процесу тестування ПЗ від ітерацій, на яких було припинено тестування

Висновки

Побудовано нову математичну модель надійності програмного забезпечення з динамічним показником складності програмного проекту. Форми кривих функції кумулятивної кількості несправностей та інтенсивності появи помилок цієї моделі точніше відповідають практичним результатам тестування ПЗ.

Виведено узагальнений вираз для загальної кількості помилок в ПЗ, що може бути використано для прийняття рішень на етапі тестування ПЗ.

Проведено порівняльний аналіз існуючих та розробленої моделей надійності ПЗ на реальних тестових прикладах та показано переваги моделі з індексом складності проекту над S-подібною моделлю та моделлю Goel–Okumoto. Запропонована модель усуває недоліки обох цих моделей шляхом введення дійсного показника величини проекту, і, завдяки цьому, адекватніше описує залежності характеристик надійності реальних програмних продуктів.

Описано та формалізовано критерій достатності процесу тестування ПЗ, який ґрунтується на поведінці залежності індексу величини проекту від циклів тестування програмного продукту. Показано, що усі інші моделі надійності ПЗ на основі розподілу Пуассона не дають змоги отримати

такий критерій внаслідок відсутності необхідних параметрів моделей, які б описували якісну зміну форми розподілу.

Розроблена модель надійності ПЗ придатна до використання на підприємстві для оцінювання загальної кількості помилок в ПЗ та прогнозування частоти їх появи. Індекс складності проекту та критерій достатності процесу тестування дає змогу надати практичні рекомендації керівникам проекту стосовно розподілу виробничих ресурсів між етапами життєвого циклу проекту.

1. Половко А.М., Гуров С.В. *Основы теории надежности*. – СПб.: БХВ-Петербург, 2006. – 704 с.
2. Липаев В.В. *Надежность программных средств*. – М.: СИНТЕГ, 1998. – 232 с.
3. Теїер Т., Липов М., Нельсон Э. *Надежность программного обеспечения*. Пер. с англ. – М.: Мир, 1981. – 323 с.
4. Goel, A.L. *Software reliability models: assumptions, limitations, and applicability*. // *IEEE Transactions on software engineering*. 1985, Vol. SE-11, No 12, pp. 1411-1423.
5. Cai K.-Y., Hu D.-B., Bai Ch.-G., Hu H., Jing T. *Does software reliability growth behavior follow a non-homogeneous Poisson process* // *Information and Software Technology*. – Vol. 50. – 2008. – P. 1232–1247.
6. Shooman M.L. *Probabilistic models for software reliability prediction* // in *Statistical Computer Performance Evaluation*. – W. Freiberger, Ed. – New York: Academic. – 1972. – P. 485–502.
7. Jelinski Z., Moranda P. *Software reliability research* // in *Statistical Computer Performance Evaluation*. – W. Freiberger, Ed. – New York: Academic. – 1972. – P. 465–484.
8. Schick G.J., Wolverton R.W. *Assessment of software reliability* // *Proc. Oper. Res.* – Physica-Verlag. – Wirzberg-Wien. – 1973. – P. 395–422.
9. Musa J.D. *A theory of software reliability and its application* // *IEEE Transactions on Software Engineering*. – SE-1(3). – 1975. – P. 312–327.
10. Schneidewind N.F. *Analysis of Error Process in Computer Software* // *Sigplan Note*. – Vol. 10. – No.6. – 1975. – P.337–346.
11. Goel A.L., Okumoto K. *Time-Dependent Error-Detection Rate Model for Software and other Performance Measures* // *IEEE Transactions on Reliability*. – Vol. R-28. – No. 3. – 1979. – P. 206–211.
12. Yamada S., Ohba M., Osaki S. *S-shaped reliability growth modeling for software error detection* // *IEEE Transactions on Reliability*. – Vol. R-32. – No.5. – 1983. – P. 475–478.
13. Тимошенко Ю.О., Дідковська М.В. *Узагальнена модель негомогенного пуассонівського процесу для оцінювання надійності програмного забезпечення* // *Проблеми програмування*. – № 2–3. – 2004. – С.480–489.
14. McCabe T.J. *A complexity measure* // *IEEE Transactions on Software Engineering*. – Vol. SE-2. – No. 4. – 1976. – P. 308-320.