

УДК 004.052; 004.942

В. С. ЯКОВИНА, В. А. МАЦЕЛЮХ, О. О. НИТРЕБИЧ

Національний університет «Львівська політехніка», Україна

## АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ РОЗПОДІЛУ ЗУСИЛЬ ТЕСТУВАННЯ ДЛЯ ОЦІНЮВАННЯ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З УРАХУВАННЯМ НЕДОСКОНАЛОГО ВІДЛАГОДЖЕННЯ

Ряд моделей надійності програмного забезпечення у врахуванням недосконалого відлагодження базуються на функції розподілу зусиль тестування. В цій статті здійснено огляд найвідоміших математичних моделей розподілу зусиль тестування та проведено їх порівняльний аналіз з використанням двох емпіричних наборів даних. Показано, що математична модель, що описує зусилля тестування у вигляді логарифм логістичної функції розподілу не поступається своїм аналогам в точності опису емпіричних даних тестування і краще описує програмне забезпечення на етапі підтримки та супроводу.

**Ключові слова:** програмне забезпечення, модель надійності, недосконале відлагодження, розподіл зусиль тестування.

### Вступ

Більшість сучасних електронних систем працюють під управлінням програмного забезпечення (ПЗ) в тому чи іншому вигляді. Використання таких систем для критичних застосувань підвищує вимоги до достовірності оцінювання показників їх надійності. Оцінювання показників надійності програмно-апаратних систем неможливе без врахування показників надійності ПЗ, оцінювання яких здійснюється на основі відповідних моделей надійності. Водночас сучасні програмні засоби стають все складнішими, удосконалюються і стають складнішими технології і методології їх розробки, що призводить до зменшення ступеня адекватності існуючих моделей надійності ПЗ. Саме цим і зумовлена актуальність підвищення ступеня адекватності моделей надійності ПЗ з метою підвищення достовірності оцінювання показників його надійності.

### Моделі надійності ПЗ з врахуванням недосконалого відлагодження

Протягом останніх десятиліть для оцінки якості та надійності ПЗ було запропоновано та досліджено значну кількість аналітичних моделей надійності ПЗ (МНПЗ). Кожна модель базується на певних припущеннях про процес розробки та середовище функціонування ПЗ. При цьому на різних етапах життєвого циклу можуть використовуватись різні моделі надійності ПЗ, оскільки поведінка показників надійності суттєво відрізняється протягом життєвого циклу ПЗ. Так, під час етапу відлагодження ПЗ кількість помилок, що залишилися в коді програми, змінюється. Переважна більшість моделей припускає ідеальне відлагодження – імовірність усунення помилки становить 100%, під час усунення помилки не

вводиться нових, таким чином кількість помилок, що залишилась в програмі є незростаючою функцією тривалості відлагодження [1, 2]. В моделях, що припускають неідеальне відлагодження (помилки не усуваються, вводяться, чи не змінюються під час кожного відлагодження), кількість помилок, що залишилися, може зростати чи зменшуватись. Такий процес можна представити біноміальною моделлю, моделлю Пуассона, складним пуассоновим процесом, чи марковським процесом [3–6].

Порівняно з ідеальним відлагодженням, недосконале може описувати процес тестування більш адекватно і достовірно. В останні роки процес недосконалого відлагодження привертає до себе все більше уваги. На сьогодні низка авторів і наукових шкіл (Н. Pham [7], Р. К. Kapur та S. Younes [8], Е. Slud [9], В. С. Харченко, О. М. Одарущенко, О. А. Руденко [10], Д. А. Маєвський [11] та ін.) працюють над питанням розроблення та вдосконалення МНПЗ з врахуванням недосконалого відлагодження.

У роботі [8] запропоновано наступну функцію інтенсивності відмов ПЗ, яка враховує недосконале відлагодження:

$$\lambda(t) = \frac{\delta mf(t)}{\delta t} = \frac{Nb^2 e^{-bt}}{(b-c) + ce^{-bt}}, \quad (1)$$

де  $N$  – початкова кількість помилок;

$b$  – коефіцієнт видалення помилок;

$c$  – початковий коефіцієнт недосконалого відлагодження.

В [10] описано ряд моделей, що враховують недосконале відлагодження. Авторами запропоновано модифікацію існуючих моделей, які враховують фактор внесення вторинних дефектів. Зокрема модифікацію функції ризику моделі Jelinski-Moranda здійснено шляхом додавання доданку  $n^{BH}$ , який

визначає інтенсивність прояву дефектів:

$$\lambda(t_i) = K(B - (i-1) + n^{BH}), \quad (2)$$

де  $B$  – початкова кількість дефектів;

$K$  – коефіцієнт пропорційності;

$t_i$  – довільна точка часу між виявленнями  $i-1$  та  $i$ -го дефектів.

Іншим підходом до урахування процесу недосконалого відлагодження в МНПЗ, який активно розвивається протягом останніх років, є використання функції зусиль тестування [12]. Функції, що описують як розподілені зусилля тестування в визначений період розробки ПЗ називають функціями зусиль тестування [13]. Зусилля тестування визначаються як людино-години, кількість тестових випадків, час роботи процесора, які були затрачені під час фази тестування.

Для побудови МНПЗ з урахуванням недосконалого відлагодження на основі функції зусиль тестування, у статті [14] комбінують експоненційну функцію зусиль тестування Вейбулла з  $S$ -подібною МНПЗ. N. Ahmad [15] запропонував  $S$ -подібну МНПЗ, яка враховує недосконале відлагодження та використовує логарифм-логістичну функцію зусиль тестування. Р. К. Каруг [16] побудував уніфіковану МЗНПЗ, що розглядає зусилля тестування та недосконале відлагодження, згідної якої реальний процес тестування поділяється на етапи виявлення та виправлення відмов [12].

Зазвичай припускають, що кількість виявлених відмов за інтервал часу  $[t, t + \Delta t]$  при поточних зусиллях тестування є пропорційною до кількості помилок, які залишились в системі на час  $t$  [17]. Звідси функція інтенсивності відмов може бути записана як:

$$\lambda_d(t) = \frac{dm_d(t)}{dt} = b(t)w(t)(a(t) - m_d(t)), \quad (3)$$

де  $b(t)$  – коефіцієнт виявлення відмов;

$w(t)$  – зусилля тестування за час  $t$ ;

$a(t)$  – функція кількості помилок;

$m_d(t)$  – кількість відмов виявлених за час  $t$ .

Врахування зусиль тестування під час розроблення МНПЗ підвищує ступінь адекватності таких моделей та дає можливість краще пояснити процес споживання ресурсів під час цієї фази розробки ПЗ [18].

### Постановка задачі дослідження

Чим достовірніше функція зусиль тестування описуватиме реальний процес тестування, тим краще модель надійності ПЗ, яка враховує зусилля тестування, буде описувати інтенсивність відмов ПЗ. Тому побудова функції зусиль тестування, яка адекватно описує процес тестування, є актуальним за-

вданням, вирішення якого дасть можливість підвищити достовірність оцінювання показників надійності ПЗ.

Метою цієї роботи є дослідження моделей, що описують розподіл зусиль тестування ПЗ та пошук тієї, яка б більш достовірно описувала процес тестування ПЗ для її використання в моделях надійності ПЗ з урахуванням недосконалого відлагодження.

### Огляд найбільш поширених функцій зусиль тестування

Найчастіше в МНПЗ використовуються такі функції розподілу зусиль тестування: постійна, експоненційний та логістичний розподіли, розподіли Релея та Вейбулла. Експоненційний та Релеїв розподіли можуть розглядатись як часткові випадки розподілу Вейбулла [13], [17],[19].

**Постійна функція.** Дана функція розглядає зусилля тестування як сталу величину. У класичних моделях зростання надійності ПЗ дослідники вважають, що зусилля тестування програмного забезпечення є константою:

$$w(t) = w_0. \quad (4)$$

Кумулятивні зусилля тестування можна отримати з наступного виразу:

$$W(t) = wt. \quad (5)$$

Як видно з (5), загальні зусилля тестування прямують до безмежності. У випадку, якщо функція зусиль тестування не розглядається, можна вважати, що використовується постійна функція з одиничним значенням:

$$w(t) = 1. \quad (6)$$

Ця функція погано описує розподіл зусиль тестування, оскільки зазвичай такі зусилля не розподілені рівномірно під час фази тестування.

**Розподіл Вейбулла.** Крива Вейбулла є дуже гнучкою і вона може значно покращити опис емпіричних даних, які часто використовуються при дослідженні МНПЗ [19]. Там, де важко описати зусилля тестування тільки експоненційною кривою чи кривою Релея, використовують криву Вейбулла, яка має вигляд:

$$w(t) = N\beta m t^{m-1} e^{-\beta t^m}, \quad (7)$$

де  $N$  – очікувана загальна кількість зусиль тестування необхідна для тестування ПЗ;

$\beta$ ,  $m$  – параметри масштабу та форми відповідно.

Значення  $m > 3$  не використовуються при описі розподілу зусиль тестування ПЗ, оскільки в цьому випадку форма кривої не відповідає реальним процесам розроблення ПЗ.

Кумулятивні зусилля тестування записують як:

$$W(t)=N(1-e^{-\beta t^m}). \quad (8)$$

Варто зазначити, що кумулятивні зусилля тестування скінченні і прямують до  $N$ .

**Експоненційний розподіл.** Така крива є частковим випадком кривої Вейбулла при  $m=1$ . Експоненційна крива є зручною для опису середовища тестування, яке монотонно знижує показник зусиль тестування [18].

Експоненційна функція зусиль тестування описується формулою:

$$w(t)=N\beta e^{-\beta t}, \quad (9)$$

а кумулятивна функція зусиль тестування:

$$W(t)=N(1-e^{-\beta t}), \quad (10)$$

де  $N$  – очікувана загальна кількість зусиль тестування необхідна для тестування ПЗ;

$\beta$  – параметр масштабу.

**Розподіл Релея.** Крива Релея є частковим випадком функції розподілу зусиль тестування Вейбулла при  $m=2$ . Функція зусиль тестування Релея спочатку збільшується, досягаючи свого піку, і потім спадає, зменшуючи зусилля тестування асимптотично до нуля [19]:

$$w(t)=N\beta t e^{-\frac{\beta}{2}t^2}, \quad (11)$$

кумулятивні зусилля тестування:

$$W(t)=N(1-e^{-\frac{\beta}{2}t^2}), \quad (12)$$

де  $N$  – очікувана загальна кількість зусиль тестування необхідна для тестування ПЗ;

$\beta$  – параметр масштабу.

**Логістичний розподіл.** Логістична крива вперше запропонована в [20] як альтернатива кривій Релея. Вони мають подібну поведінку окрім початкової стадії процесу тестування. Логістичний розподіл зусиль тестування використовувався замість кривої Вейбулла в деяких роботах [20, 21] і показав високу точність опису розподілу зусиль тестування.

Логістична крива зусиль тестування записується як:

$$w(t)=\frac{NA\eta}{(e^{\frac{\eta t}{2}} + Ae^{-\frac{\eta t}{2}})^2}, \quad (13)$$

де  $A$  – параметр-константа;

$\eta$  – міра споживання затрат на тестування;

$N$  – кумулятивні зусилля тестування.

Кумулятивні зусилля тестування у випадку логістичного розподілу мають вигляд:

$$W(t)=\frac{N}{1+Ae^{-\eta t}}. \quad (14)$$

Подібно до кривої Вейбулла, кумулятивні зусилля тестування є скінченними і прямують до  $N$ .

**Логарифм логістична функція зусиль тестування.** Враховуючи неоднорідність шаблонів розподілу даних тестування, доцільно використовувати функції зусиль тестування, які найкраще описують емпіричну криву розподілу зусиль тестування.

Бохарі та Ахмад запропонували логарифм логістичну функцію зусиль тестування для прогнозування поведінки помилок та відмов в програмному забезпеченні. Вони показали, що логарифм логістична функція зусиль тестування добре підходить і є гнучкою для оцінки надійності програмних продуктів [20].

Вигляд логарифм логістичної функції розподілу зусиль тестування є таким:

$$w(t)=\frac{N(\frac{t}{\lambda})^{-\beta}}{(1+(\frac{t}{\lambda})^{-\beta})^2 t}, \quad (15)$$

де  $N$  – кумулятивні зусилля тестування;

$\lambda > 0$  – параметр масштабу;

$\beta > 0$  – параметр форми.

Функція кумулятивних зусиль тестування в такому випадку має вигляд:

$$W(t)=\frac{N}{1+(\frac{t}{\lambda})^{-\beta}}. \quad (16)$$

Логарифмічно логістичний розподіл випадкової величини, такої як кількість зусиль тестування, говорить про більшу імовірність для відхилення значень величини вгору, оскільки крива такого розподілу має правобічну скошеність. Тому, на відміну від аналогів, функція (15) добре описує розподіл зусиль тестування для проектів, де ведеться активна фаза підтримки проекту разом з додаванням нового функціоналу. У таких проектах зусилля тестування не будуть різко спадати, а поступово нарощуються до певної межі. Функції Вейбулла (8) та логістична (13) відображають доволі різке призупинення нарощування зусиль тестування порівняно із логарифм логістичною функцією (15).

## Результати експерименту та їх аналіз

Для дослідження ефективності використання різних функцій розподілу зусиль тестування та їх порівняльного аналізу, було використано два набори даних. Перший набір – це дані про тестування системи Т1 Римського повітряного центру розробки [21]. Незважаючи на те, що цей набір досить старий, він широко використовується та містить дані про зусилля тестування, які представлено використаним процесорним часом за тиждень. Другий набір – це результати тестування програмного пакету введення даних онлайн [22]. Результати тестування зазначе-

них програмних продуктів наведені в табл. 1 та 2 відповідно.

Таблиця 1

Розподіл зусиль тестування для тестового набору № 1 [21]

t, тижні	Зусилля тестування w(t <sub>i</sub> ) (проц. год.)	Кумулятивні зусилля w(t <sub>i</sub> ) (проц. год.)
1	4	4
2	4,3	8,3
3	2	10,3
4	0,6	10,9
5	2,3	13,2
6	1,6	14,8
7	1,8	16,6
8	14,7	31,3
9	25,1	56,4
10	4,5	60,9
11	9,5	70,4
12	8,5	78,9
13	29,5	108,4
14	22	130,4
15	39,5	169,9
16	26	195,9
17	25	220,9
18	31,4	252,3
19	30	282,3
20	12,8	295,1
21	5	300,1

Таблиця 2

Розподіл зусиль тестування для тестового набору № 2 [22]

t, тижні	Зусилля тестування w(t <sub>i</sub> ) (проц. год.)	Кумулятивні зусилля w(t <sub>i</sub> ) (проц. год.)
1,00	2,45	2,45
2,00	2,45	4,90
3,00	1,96	6,86
4,00	0,98	7,84
5,00	1,68	9,52
6,00	3,37	12,89
7,00	4,21	17,10
8,00	3,37	20,47
9,00	0,96	21,43
10,00	1,92	23,35
11,00	2,88	26,23
12,00	1,44	27,67
13,00	3,26	30,93
14,00	3,84	34,77
15,00	3,84	38,61
16,00	2,30	40,91
17,00	1,76	42,67
18,00	1,99	44,66
19,00	2,99	47,65

На основі емпіричних даних представлених у табл. 1, 2 з використанням методу найменших квад-

ратів, обчислено значення параметрів постійної, логістичної, логарифм логістичної та вейбулівської функцій зусиль тестування. Значення параметрів цих функцій для обох наборів даних подано у табл. 3.

Для оцінювання якості апроксимації зусиль тестування (табл. 1, 2) моделями розподілу зусиль тестування згідно (17) – (20) використано такі критерії [21]: RMSE (середньоквадратична похибка апроксимації):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (w(t_i) - w_i)^2}, \quad (17)$$

Bias (сума різниці відхилень):

$$Bias = \frac{1}{n} \sum_{i=1}^n (w(t_i) - w_i), \quad (18)$$

Variance (дисперсія):

$$Variance = \sqrt{\frac{1}{n} \sum_{i=1}^n (w(t_i) - w_i - Bias)^2}, \quad (19)$$

та RMSPE (середньоквадратична похибка прогнозування):

$$RMSPE = \sqrt{Variance + Bias^2}. \quad (20)$$

Значення цих критеріїв для різних моделей розподілу зусиль тестування також наведено в табл. 3.

Таблиця 3

Порівняння якості апроксимації зусиль тестування різними функціями

	Постійна	Вейбулла	Логістична	Логарифм логістична
<b>Тестовий набір № 1</b>				
Оцінки параметрів	W=11,5	N=375,8 β=8.4E-5 μ=3,27	N=366,2 η=0,314 A=134,6	N=549,1 λ=19,2 β=3,35
RMSE	45,6	7,93	6,04	8,48
Bias	16,0	-2,18	-0,483	-2,09
Variance	42,7	7,63	6,02	8,22
RMSPE	17,3	3,52	2,50	3,55
<b>Тестовий набір № 2</b>				
Оцінки параметрів	W=11,5	N=375,8 β=8.4E-5 μ=3,27	N=366,2 η=0,314 A=134,6	N=549,1 λ=19,2 β=3,35
RMSE	1,27	0,93	1,28	0,93
Bias	0,34	-0,04	0,10	-0,04
Variance	1,22	0,93	1,27	0,93
RMSPE	1,16	0,96	1,13	0,96

Графіки функцій зусиль тестування (табл. 3) для емпіричних даних обох експериментів (табл. 1 і 2) наведено на рис. 1 і 2, відповідно.

З представлених рисунків видно, що логарифм логістична функція є дуже гнучкою. Представлені набори тестових даних досить відрізняються розподілом зусиль тестування на кожен тиждень. Лога-

рифм логістична функція досить точно описує розподіл зусиль тестування на кожному з наборів даних. Це також видно з обчислених коефіцієнтів RMSPE. Такі ж хороші показники у функції Вейбулла. Різниця між функціями полягає у представленні розподілу зусиль тестування на пізніх етапах експлуатації ПЗ.

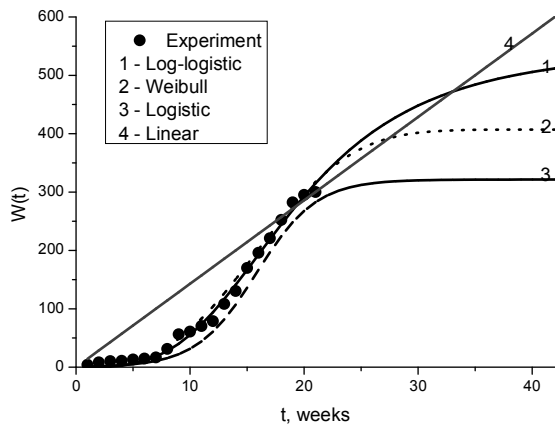


Рис. 1. Опис зусиль тестування різними функціями для тестового набору даних № 1

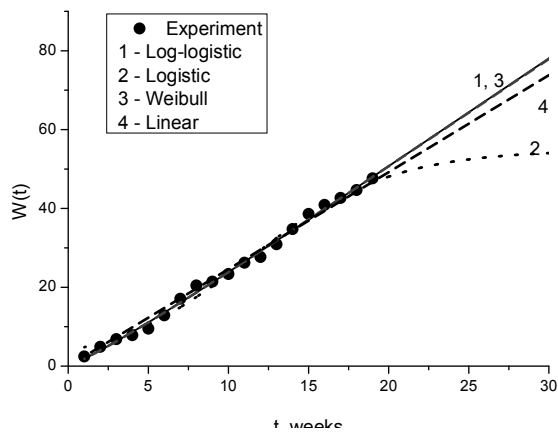


Рис. 2. Опис зусиль тестування різними функціями для тестового набору даних № 2

З наведених графіків видно, що досліджувані набори тестових даних мають достатньо різний розподіл зусиль тестування. Так, у випадку набору № 1 (рис. 1) розподіл зусиль тестування носить яскраво виражений нелінійний характер з тенденцією до виходу на насичення. В той же час наявні емпіричні дані щодо розподілу зусиль тестування другого програмного продукту не дають можливості зробити висновок про їх насичення і є достатньою мірою лінійними. Це підтверджують і дані табл. 3, де лінійна функція розподілу зусиль тестування доволі непогано описує другий набір даних. З табл. 3 видно, що для першого набору даних розподіл зусиль тестування найкраще описує логістична функція, а для другого набору – логарифм логістична і функція Вейбулла. З рис. 1, натомість, яскраво видно відмінність між досліджуваними функціями розподілу

зусиль тестування, які найяскравіше проявляються на пізніх етапах процесу тестування. Так, логістична функція і функція Вейбулла демонструють насичення кумулятивних зусиль тестування, тобто відображають факт припинення процесу тестування, що доволі обмежено описує реальні процеси індустрії ПЗ. Лінійна функція демонструє нескінченний процес тестування, що не відповідає реаліям процесу розроблення програмних продуктів. Логарифм логістична функція, на відміну від інших, показує сповільнення зростання кумулятивних зусиль тестування, однак без виходу її на насичення, що, як уже зазначалось, достатньо адекватно описує процеси підтримки проекту разом з додаванням нового функціоналу. З рис. 2 таких висновків зробити неможливо, оскільки емпіричні дані є майже лінійними і усі функції (крім логістичної) демонструють схожу поведінку зусиль тестування.

## Висновки

Функції розподілу зусиль тестування використовують в моделях надійності ПЗ, що враховують процес недосконалого відлагодження, під час якого помилка може бути не видалена з системи, або ж можуть бути внесені нові помилки в програмний код. В цій роботі досліджено якість опису розподілу зусиль тестування найбільш поширеними функціями на прикладі емпіричних даних тестування двох програмних систем з різним характером тестування. Якість опису емпіричних даних оцінювали на основі середньоквадратичної похибки апроксимації, суми різниці відхилень, дисперсії та середньоквадратичної помилки прогнозування.

Показано, що логарифм логістична функція зусиль тестування досить точно описує емпіричні дані: критерії якості апроксимації дещо поступаються логістичній кривій і практично не поступаються кривій Вейбулла. На початкових стадіях розробки ПЗ, коли зусилля тестування нарастають спочатку повільніше, а потім стрімко збільшуються, логарифм логістична функція описує їх подібно до функцій Вейбулла та логістичної. Водночас на завершальних етапах життєвого циклу ПЗ, коли логістична і вейбулівська функції демонструють зупинку приросту зусиль тестування, логарифм логістична функція зусиль тестування описує повільно зростаючу криву, що добре відображає фазу супроводу та розробки нового функціоналу ПЗ. Таким чином, функція (15), (16) більш адекватно і достовірно описує розподіл зусиль тестування в проектах з активною фазою супроводу.

Подальші дослідження будуть спрямовані на аналіз застосування логарифм логістичної функції зусиль тестування стосовно різних методологій розробки ПЗ, а також побудову та верифікацію мо-

делі надійності ПЗ з урахуванням недосконалого відлагодження з використанням функції розподілу зусиль тестування.

### Література

1. Goel, A. L. *Software Reliability Models: Assumptions, Limitations, and Applicability [Text]* / A. L. Goel IEEE // *Trans. Software Eng.* – Vol. 11. – 1985. – P. 1411-1423.

2. Pham, H. *Software Reliability Models for Critical Applications [Text]* / H. Pham, M. Pham // EGG—2663 Technical Report. Idaho National Engineering Laboratory. – EG&G Idaho Inc., 1991. – 98 p.

3. Pham, H. *System software reliability [Text]* / H. Pham. – Springer-Verlag London Limited, 2006. – 440 p.

4. Sheakh, Tariq Hussain. *A Study of Analytically Improving the Reliability of Software [Text]* / Tariq Hussain Sheakh, S. M. K. Quadri, VijayPal Singh // *Intern. Journal of Research and Reviews in Computer Science.* – 2012. – Vol. 3, No 1. – P. 1404–1406.

5. Rahmani, Cobra. *Exploitation of Quantitative Approaches to Software Reliability [Text]* / Cobra Rahmani, Azad Azadmanesh. – University of Nebraska at Omaha, 2008. – 32 p.

6. Одарущенко, О. Н. *Метод оцінювання надійності програмних засобів з урахуванням вторинних дефектів [Текст]* / О. Н. Одарущенко, А. А. Руденко, В. С. Харченко // *Радіоелектронні і комп'ютерні системи.* – 2012. – № 7 (59). – С. 294–300.

7. Pham, H. *A Software Cost Model with Imperfect Debugging, Random Life Cycle and Penalty Cost [Text]* / H. Pham // *Int'l J. Syst. Science.* – Vol. 27. – 1996. – P. 455-463.

8. Kapur, P. *Modelling an Imperfect Debugging Phenomenon in Software Reliability [Text]* / P. Kapur, S. Younes // *Microelectronics and Reliability.* – Vol. 36. – 1996. – P. 645- 650.

9. Slud, E. *Testing for Imperfect Debugging in Software Reliability [Text]* / E. Slud // *Scandinavian J. Statistics.* – Vol. 24. – 1997. – P. 555-572.

10. Одарущенко, О. Н. *Учет вторичных дефектов в моделях надежности программных средств [Текст]* / О. Н. Одарущенко, А. А. Руденко, В. С. Харченко // *Математичні машини і системи.* – 2010. – № 1. – С. 205–217.

11. Маевский, Д. А. *Влияние вторичных дефектов на надежность динамических информационных систем [Текст]* / Д. А. Маевский // *Вісник НТУ «ХПИ».* – 2012. – № 50 (956). – С. 54–58.

12. *A Unified and Flexible Framework of Imperfect Debugging Dependent SRGMs with Testing Effort [Text]* / Ce Zang, Gang Cui, Hongwei Liu, Fanchao Meng, Shixiong Wu // *Journal of Multimedia.* – February 2014. – Vol. 9, No. 2. – P. 310-317.

13. Huang, Chin-Yu. *An Assessment of Testing-Effort Dependent Software Reliability Growth Models [Text]* / Chin-Yu Huang, Sy-Yen Kuo, Michael R. Lyu // *IEEE Transactions on Reliability.* – June 2007. – Vol. 56, No. 2. – P. 198-211.

14. Bokhari, M. U. *Analysis of a software reliability growth models: the case of log-logistic testeffort function [Text]* / M. U. Bokhari, N. Ahmad // *The 17th IASTED international conference on Modelling and simulation.* Montreal, Canada, 2006. – P. 540- 545.

15. Ahmad, N. *Analysis of an inflection S-shaped software reliability model considering log-logistic testing-effort and imperfect debugging [Text]* / N. Ahmad, M. G. Khan, L. S. Rafi // *International Journal*

*of Computer Science and Network Security.* – 2011. – Vol. 11. – P. 161-171.

16. *Unified framework for developing testing effort dependent software reliability growth models [Text]* / P. K. Kapur, O. Shatnawi, A. G. Aggarwal, R. Kumar // *Wseas Trans. on Systems.* – 2009. – Vol. 4. – P. 521-531.

17. Kuo, Sy-Yen. *Framework for Modeling Software Reliability, Using Various Testing-Efforts and Fault-Detection Rates [Text]* / Sy-Yen Kuo, Chin-Yu Huang, Michael R. Lyu // *IEEE Transactions on Reliability.* – September 2001. – Vol. 50, No. 3. – P. 310-320.

18. Yamada, S. *A testing-effort dependent software reliability model and its application [Text]* / Shigeru Yamada, Hiroshi Ohtera, Hiroyuki Narihisa // *Microelectron. Reliab.* – 1987. – Vol. 27, No. 3. – P. 507-522.

19. Ahmad, N. *Software Reliability Growth Models with Log-logistic Testing-Effort Function: A Comparative Study [Text]* / N. Ahmad, Md. Zafar Imam // *International Journal of Computer Applications (0975 – 8887).* – August 2013. – Vol. 75, No. 12. – P. 6-10.

20. Parr, F. *An Alternative to the Rayleigh Curve Model for Software Development Effort [Text]* / F. N. Parr // *IEEE Transactions on Software Engineering.* – May 1980. – Vol. 6, No. 3. – P. 291-296.

21. *Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction [Text]* / R. Peng, Y. F. Li, W. J. Zhang, Q. P. Hu // *Reliability Engineering and System Safety.* – 2014. – Vol. 126. – P. 37–43.

22. Ohba, M. *Software reliability analysis models [Text]* / Mitsuru Ohba // *IBM J.RES. DEVELOP.* – July 1984. – Vol. 28, No. 4. – P. 428-443.

### References

1. Goel, A. L. *Software Reliability Models: Assumptions, Limitations, and Applicability.* *Trans. Software Eng.*, vol. 11, 1985, pp. 1411-1423.

2. Pham, H., Pham, M. *Software Reliability Models for Critical Applications.* EGG—2663 Technical Report. Idaho National Engineering Laboratory, EG&G Idaho Inc. Publ., 1991. 98 p.

3. Pham, H. *System software reliability.* Springer-Verlag London Limited Publ., 2006. 440 p.

4. Sheakh, Tariq Hussain, Quadri, S. M. K., Singh, VijayPal. *A Study of Analytically Improving the Reliability of Software.* *Intern. Journal of Research and Reviews in Computer Science*, 2012, vol. 3, no. 1, pp. 1404–1406.

5. Cobra Rahmani. *Exploitation of Quantitative Approaches to Software Reliability.* University of Nebraska at Omaha Publ., 2008. 32 p.

6. Odarushhenko, O. N., Rudenko, A. A., Harchenko, V. S. *Method oцenyvaniya nadezhnosti programnyh sredstv s uchetom vtorychnyyh defektov [Method of estimation of software reliability based on secondary defects].* *Radioelektronni i kompyuterni systemy*, 2012, no. 7 (59), pp. 294–300.

7. Pham, H. *A Software Cost Model with Imperfect Debugging, Random Life Cycle and Penalty Cost.* *Int'l J. Syst. Science*, vol. 27, 1996, pp. 455-463.

8. Kapur, P., Younes, S. *Modelling an Imperfect Debugging Phenomenon in Software Reliability.* *Microelectronics and Reliability*, vol. 36, 1996, pp. 645-650.

9. Slud, E. *Testing for Imperfect Debugging in Software Reliability.* *Scandinavian J. Statistics*, vol. 24, 1997, pp. 555-572.

10. Odarushhenko, O. N., Rudenko, A. A., Harchenko, V., S. *Uchet vtorychnyyh defektov v modeljah*



nadezhnosti programmnykh sredstv [Accounting for secondary defects software reliability models]. *Matem. mashyny i systemy*, 2010, no. 1, pp. 205–217.

11. Maevskiy, D. A. Vlyaniye vtorykhnykh defektov na nadezhnost dynamichnykh ynforymatsionnykh system [Influence of secondary defects on the reliability of the dynamic information systems]. *Visnyk NTU «HPI»*, 2012, no. 50(956), pp. 54–58.

12. Zang, Ce, Cui, Gang, Liu, Hongwei, Meng, Fanchao, Wu, Shixiong. A Unified and Flexible Framework of Imperfect Debugging Dependent SRGMs with Testing Effort. *Journal of Multimedia*, vol. 9, no. 2, February 2014, pp. 310-317.

13. Huang, Chin-Yu, Kuo, Sy-Yen, Lyu, Michael R. An Assessment of Testing-Effort Dependent Software Reliability Growth Models. *IEEE Transactions on Reliability*, vol. 56, no. 2, June 2007, pp. 198-211.

14. Bokhari, M., Ahmad, N. Analysis of a software reliability growth models: the case of log-logistic test effort function. *The 17th IASTED international conference on Modelling and simulation*. Montreal, Canada, 2006, pp. 540- 545.

15. Ahmad, N., Khan, M. G., Rafi, L. S. Analysis of an inflection S-shaped software reliability model considering log-logistic testing-effort and imperfect debugging. *International Journal of Computer Science and Network Security*, vol. 11, 2011, pp. 161-171.

16. Kapur, P., Shatnawi, O., Aggarwal, A., Kumar, R. Unified framework for developing testing

effort dependent software reliability growth models. *Wseas Trans. on Systems*, vol. 4, 2009, pp. 521-531.

17. Kuo, Sy-Yen, Huang, Chin-Yu, Lyu, Michael R. Framework for Modeling Software Reliability, Using Various Testing-Efforts and Fault-Detection Rates. *IEEE Transactions on Reliability*, vol. 50, no. 3, September 2001, pp. 310-320.

18. Yamada, Shigeru, Ohtera, Hiroshi, Narihisa, Hiroyuki. A testing-effort dependent software reliability model and its application. *Microelectron. Reliab*, vol. 27, no. 3, 1987, pp. 507-522.

19. Ahmad, N., Imam, Md. Zafar. Software Reliability Growth Models with Log-logistic Testing-Effort Function: A Comparative Study. *International Journal of Computer Applications (0975 – 8887)*, vol. 75, no. 12. August 2013. pp. 6-10.

20. Parr, F. An Alternative to the Ravleigh Curve Model for Software Development Effort. *IEEE Transactions on Software Engineering*, vol. 6, no. 3, May 1980, pp. 291-296.

21. Peng, R., Li, Y. F., Zhang, W. J., Hu, Q. P. Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction. *Reliability Engineering and System Safety*, vol. 126, 2014, pp. 37–43.

22. Ohba, M. Software reliability analysis models. *IBM J.RES. DEVELOP*, vol. 28, no. 4, July 1984, pp. 428-443.

*Надійшла до редакції 3.09.2016, розглянута на редколегії 16.09.2016*

#### АНАЛИЗ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ РАСПРЕДЕЛЕНИЯ УСИЛИЙ ТЕСТИРОВАНИЯ ДЛЯ ОЦЕНИВАНИЯ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С УЧЕТОМ НЕСОВЕРШЕННОЙ ОТЛАДКИ

*В. С. Яковина, В. А. Мацелюх, О. А. Нитребич*

Ряд моделей надежности программного обеспечения с учетом несовершенной отладки базируются на функции распределения усилий тестирования. В этой статье сделан обзор наиболее распространенных математических моделей распределения усилий тестирования и проведен их сравнительный анализ с использованием двух эмпирических наборов данных. Показано, что математическая модель, описывающая усилия тестирования в виде логарифм логистической функции распределения не уступает своим аналогам в точности описания эмпирических данных тестирования и лучше описывает программное обеспечение на этапе поддержки и сопровождения.

**Ключевые слова:** программное обеспечение, модель надежности, несовершенная отладка, распределение усилий тестирования.

#### THE ANALYSIS OF TESTING EFFORT MODELS FOR SOFTWARE RELIABILITY ASSESSMENT WITH IMPERFECT DEBUGGING

*V. S. Yakovyna, V. A. Matselyukh, O. O. Nytrebych*

Various software reliability models with imperfect debugging use testing-effort functions. In this paper, the review of the most important testing effort functions is made. A comparative analysis to evaluate the effectiveness for the reviewed models is performed using two empirical data sets. Comparative results show that the Log Logistic testing effort model is not less accurate in describing empirical testing data than existing models and more adequately describes testing effort distribution on the phase of software system support and maintaining.

**Key words:** software, reliability model, imperfect debugging, testing effort distribution.

**Яковина Віталій Степанович** – д-р техн. наук, доцент, завідувач кафедри програмного забезпечення Національного університету «Львівська політехніка», e-mail: vitaliy.s.yakovyna@lpnu.ua.

**Мацелюх Володимир Андрійович** – аспірант, асистент кафедри програмного забезпечення Національного університету «Львівська політехніка», e-mail: volodymyr.matselyukh@yandex.ru.

**Нитребич Оксана Олександрівна** – канд. техн. наук, асистент кафедри програмного забезпечення Національного університету «Львівська політехніка», e-mail: ksenija.volynj@gmail.com.

**Yakovyna Vitaliy** – Doctor of Technical Science, Head of Software Department, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: vitaliy.s.yakovyna@lpnu.ua.

**Matselyukh Volodymyr** – postgraduate at Software Department, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: volodymyr.matselyukh@yandex.ru.

**Nytrebych Oksana** – Candidate of Technical Science, Assistant Professor at Software Department, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: ksenija.volynj@gmail.com.