computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

МЕТОД ОЦІНЮВАННЯ ТА ПРОГНОЗУВАННЯ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МОДЕЛІ З ДИНАМІЧНИМ ПОКАЗНИКОМ ВЕЛИЧИНИ ПРОЕКТУ

Максим Сенів, Віталій Яковина, Ярослав Чабанюк, Дмитро Федасюк

Національний університет "Львівська політехніка" вул. С. Бандери, 12, Львів, 79013, Україна
e-mail: max1sudden@gmail.com, yakovyna@polynet.lviv.ua, yaroslav_chab@yahoo.com, fedasyuk@lp.edu.ua,
http://lp.edu.ua/IKN/pz/

Резюме: *в роботі розглядається метод оцінювання та прогнозування надійності програмного забезпечення на основі моделі з динамічним показником величини проекту. Розроблена процедура підтримки прийняття рішень при виробництві програмних продуктів на основі критерію достатності тестування програмного забезпечення. Наведено приклад використання розробленого методу на основі даних тестування промислового програмного продукту.*

Ключові слова: *надійність програмного забезпечення, життєвий цикл програмного забезпечення, модель надійності, припущення моделі, програмна інженерія.*

THE METHOD OF SOFTWARE RELIABILITY EVALUATION AND PREDICTION BASED ON THE MODEL WITH DYNAMIC INDEX OF PROJECT SIZE

Maksym Seniv, Vitaliy Yakovyna, Yaroslav Chabanyuk, Dmytro Fedasyuk

Lviv Polytechnic National University 12 Bandery St., Lviv, 79013, Ukraine
e-mail: max1sudden@gmail.com, yakovyna@polynet.lviv.ua, yaroslav_chab@yahoo.com, fedasyuk@lp.edu.ua,
http://lp.edu.ua/IKN/pz/

Abstract: *The software reliability evaluation and prediction method based on the model with dynamic index of project size is examined in this article. The support decision making procedure during software production which is based on the adequacy criterion of software testing is developed. The example of developed method based on testing data of industrial software product is proposed.*

Keywords: *software reliability, software life cycle, reliability model, model assumption, software engineering.*

ВСТУП

На сучасному етапі розвитку інженерії програмного забезпечення підвищуються вимоги до надійності програмних продуктів, виникає потреба у скороченні затрат на тестування та, відповідно, у оцінюванні та прогнозуванні надійності розроблюваного програмного забезпечення. Адекватний аналіз і прогнозування надійності програмного забезпечення неможливі без урахування в моделях надійності характеристик і параметрів реального програмного продукту.

В попередніх дослідженнях авторами було запропоновано модель надійності ПЗ [2], яка відноситься до класу моделей на основі кількості помилок і узагальнює пуассонів розподіл на випадок динамічного показника величини програмного проекту на відміну від існуючих моделей [1, 3–5].

Важливим прикладним аспектом моделей надійності ПЗ може стати встановлення кількісного критерію достатності процесу тестування програмного продукту, який би дозволив керівникам програмних проектів більш обґрунтовано приймати рішення про виділення

ресурсів на тестування та про завершення цього етапу розробки ПЗ. В роботі [6] формалізовано критерій достатності процесу тестування ПЗ на основі використання моделі з динамічним показником величини проекту [2].

В роботі [7] проведено дослідження основних припущень та обмежень моделі надійності програмного забезпечення з динамічним показником величини проекту, які доповнюють математичний формалізм моделі; здійснено обґрунтування та подані рекомендації стосовно застосування моделі в умовах реального циклу виробництва програмного забезпечення.

Метою цієї роботи є розроблення методу оцінювання та прогнозування надійності ПЗ, а також процедури підтримки прийняття рішень на основних стадіях життєвого циклу ПЗ на основі запропонованої моделі. Застосування методу ілюструється прикладом на основі експериментальних даних тестування промислового програмного продукту.

1. МЕТОД ОЦІНЮВАННЯ І ПРОГНОЗУВАННЯ НАДІЙНОСТІ ПЗ НА ОСНОВІ РОЗРОБЛЕНОЇ МОДЕЛІ

Загальна процедура підбору моделі надійності ПЗ та процесу прийняття рішень наведена в [1]. Ця процедура носить загальний і рекомендаційний характер, не прив'язана до конкретної моделі, і надає загальні рекомендації стосовно процесу вибору моделі для оцінювання надійності ПЗ.

Розглянемо метод оцінювання і прогнозування надійності ПЗ, а також процедуру прийняття рішень при розробці програмного продукту на основі моделі [2]. Схема цього методу наведена на рис. 1 і складається з 7 кроків.

Крок 1. Підготовка та отримання даних про помилки ПЗ.

Вихідні дані для моделі отримуються в процесі тестування програмного забезпечення. Для даної моделі [2, 6] використовується кількість помилок на певних часових інтервалах. Однак перед початком застосування моделі слід переконатись у дотриманні вимог та спрощень моделі і ретельно проаналізувати вхідні дані.

Першим етапом перед аналізом даних є представлення їх у вигляді кількості помилок на вибраних часових інтервалах. Для цього діапазон вхідної статистичної вибірки ділять на k інтервалів однакової довжини ($k \le 5 \cdot \lg n$, де n – загальна кількість виявлених помилок в процесі тестування).

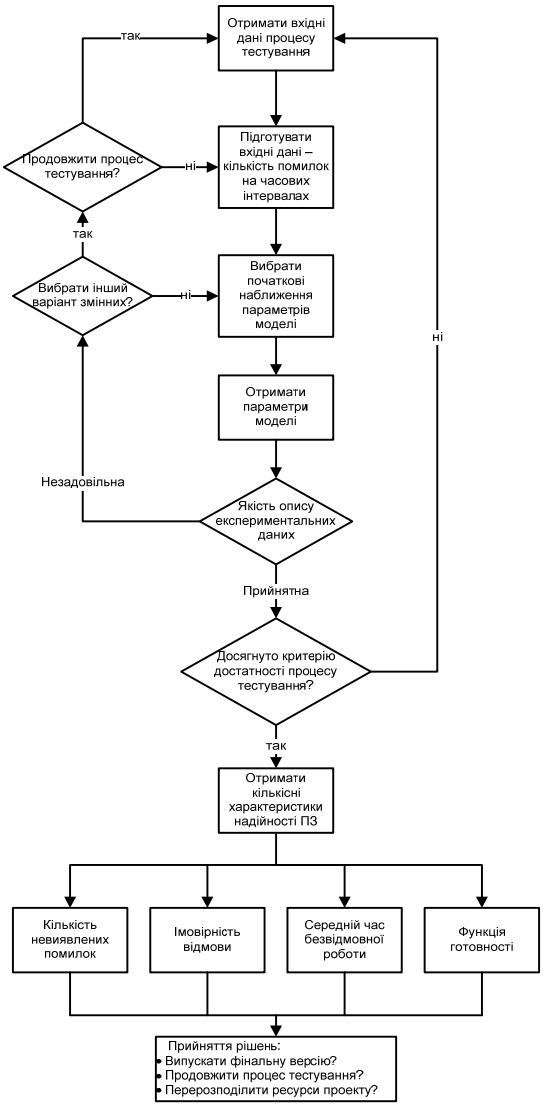

Рис. 1 – Ілюстрація методу оцінювання та прогнозування надійності ПЗ

Якщо процес тестування відбувався нерівномірно в часі, чи в процесі була задіяна різна кількість виконавців, вхідні дані слід привести до однорідного, рівномірного масштабу, наприклад до кількості людино-годин, витрачених на процес тестування. В якості часової шкали також може бути використана кількість тестових випадків (див. напр. [7]) чи процесорний час [1]. Для аналізу вхідних даних щодо придатності зазначеної моделі надійності рекомендується зобразити ці дані на графіку у вигляді кумулятивної кількості помилок як функції вибраного часового масштабу. Основним завданням на цьому етапі є встановлення відповідних змінних, які будуть використовуватись в моделі, зокрема аргументу функції кумулятивної кількості та інтенсивності виявлення помилок. Наприклад, залежно від вхідних даних та інформації про процес розробки, в якості часової шкали може виявитись більш придатною кількість пройдених тестових

наборів ніж, скажімо, календарний час. Інколи доцільним може бути поєднання декількох мір часу чи нормалізація даних, наприклад для урахування змін розміру системи під час тестування.

Крок 2. Визначення початкових наближень параметрів моделі.

Оскільки параметри моделі отримуються в результаті розв'язку системи рівнянь ітераційним методом, слід встановити початкові наближення для параметрів моделі α, β, s [2]. Зрозуміло, що у кожному конкретному випадку, для кожних вхідних даних початкові наближення будуть різними, однак можна дати деякі загальні рекомендації стосовно вибору початкових наближень.

- В якості початкового наближення параметру α можна вибрати значення кількості помилок, виявлених в процесі тестування;
- в якості початкового наближення параметру β можна вибрати обернене значення тривалості процесу тестування у вибраній часовій шкалі;
- в якості початкового наближення параметру s можна скористатись рекомендаціями щодо співвідношення цього параметру з величиною програмного проекту [2], і в якості початкового наближення брати значення із середини відповідного інтервалу.

Необхідно зазначити, що початкові наближення для параметрів моделі α, β, s можливо встановити тільки після проведення тестування хоча б на одному часовому інтервалі.

Крок 3. Отримання параметрів моделі.

Для отримання параметрів моделі в загальному можна використати різні методи в залежності від природи наявних даних [1]. В даній роботі використовується метод максимальної правдоподібності оскільки він має дуже хороші статичні властивості. Система рівнянь для визначення точкових оцінок моделі наведена в [2], для розв'язку цієї системи використовується модифікований метод Ньютона з початковими наближеннями, отриманими на кроці 2. Після розв'язку системи рівнянь, отримані точкові оцінки підставляють в модель і отримують числовий опис моделі на основі наявних даних про поведінку помилок.

Крок 4. Перевірка якості опису експериментальних даних моделлю.

Перед подальшим опрацюванням даних слід переконатись в тому чи підлягає експериментальний розподіл помилок використовуваній моделі та наскільки точно експериментальні дані описуються моделлю [2]. Для цього пропонується проведення ряду досліджень, зокрема:

- провести тест Колмогорова–Смірнова, який доцільно застосовувати в тих випадках, коли треба перевірити чи підлягає спостережена випадкова величина деякому закону розподілу, відомому з точністю до параметрів. Це один з основних і найбільш широко використовуваних непараметричних тестів, оскільки він є достатньо чутливим до різниць в досліджуваних вибірках [8, 9];
- другим критерієм є значення квадрату коефіцієнту кореляції Пірсона між експериментальною вибіркою та теоретичною вибіркою, отриманою з використанням моделі в точках, які відповідають експерименту. Коефіцієнт кореляції Пірсона (кореляція моментів добутків) використовується як показник характеру взаємного стохастичного впливу зміни двох випадкових величин, оскільки проводиться вимірювання змінних з інтервальною шкалою. Оскільки в наших дослідження коефіцієнт кореляції дуже близький до одиниці, що свідчить про функціональний зв'язок (лінійну залежність) експериментальних та теоретичних даних, для кількісної оцінки якості опису вхідних даних моделлю використовуємо значення квадрату цього коефіцієнту;
- останнім критерієм [6], що використовується для оцінки якості опису поведінки помилок в програмній системі моделлю [2] є середнє квадратичне відхилення, яке, в свою чергу дозволяє обчислити інтервал довіри та дисперсію.

У випадку успішного проходження згаданих тестів, можна вважати, що експериментальні дані з достатньою точністю описуються запропонованою моделлю і можна продовжувати проводити оцінювання та прогнозування надійності ПЗ. У іншому випадку слід спробувати змінити початкові значення (крок 2) і отримати нові точкові оцінки моделі (крок 3), або ж характер процесу виявлення помилок (можливо внаслідок специфіки розробки чи процесу тестування) не відповідає зазначеній моделі, і слід вибрати інший варіант змінних моделі (часового аргументу) або ж продовжити процес тестування з урахуванням припущень і обмежень моделі (див. вище). Не існує однозначної відповіді про те, скільки потрібно експериментальних даних тестування ПЗ чи як

вибирати часову шкалу чи початкові наближення, крім уже зазначених рекомендацій. Вирішення цих питань потребує чіткого розуміння суті моделі надійності ПЗ та процесу розробки цього програмного продукту.

Крок 5. Перевірка критерію достатності процесу тестування.

З метою більш обґрунтованого прийняття рішення про достатність процесу тестування і випуск фінальної версії продукту крім показників надійності доцільно використати критерій, описаний в [6], який базується на використанні в розробленій моделі дійсного неперервного індексу s, пов'язаного з величиною і характеристиками програмного проекту.

Для перевірки цього критерію слід мати набір значень параметру s в різні моменти часу (не менше трьох точок). Для цього часову шкалу процесу тестування $(0;T]$ розбивають на проміжки $(0;t_1]$, $(0;t_2]$, … $(0;T]$ (де $t_1 < t_2 < ... < T$), кожен з яких включає попередній, і для кожного проміжку незалежно отримують точкові оцінки параметрів моделі. Іншим випадком може бути отримання точкових оцінок параметрів моделі після кожної ітерації процесу тестування ПЗ, якщо такий передбачається проектом розробки системи.

Після цього графічно будується залежність $s(t)$, де в якості абсцис будуть значення $t_1, t_2, ..., T$, а відповідними ординатами – точкові оцінки параметру s, отримані з розв'язку системи рівнянь максимальної правдоподібності з експериментальними даними з відповідного часового інтервалу. Отримана залежність чисельно диференціюється. Критерієм достатності процесу тестування буде прямування похідної до нуля, як це описано в [6]. При виконанні цього критерію процес появи помилок відповідає негомогенному пуассоновому процесу, а модель максимально коректно відповідає покладеним в основу припущенням, і, відповідно, адекватно описує експериментальні дані.

Крок 6. Отримання кількісних характеристик надійності ПЗ.

На цьому етапі можна обчислити різні кількісні характеристики для оцінювання надійності програмної системи, такі як прогнозована загальна кількість помилок в системі, кількість залишкових помилок в системі, середній час до виявлення наступної помилки, імовірність безвідмовної роботи тощо [10]. Також можна обчислити інтервал довіри для кожної з цих характеристик для оцінки ступеня невизначеності обчислених значень показників надійності.

Крок 7. Процес прийняття рішень.

Загальною метою усіх моделей надійності ПЗ є використання їх результатів для прийняття рішень стосовно програмної системи, наприклад стосовно випуску фінальної версії чи продовження процесу розробки, перерозподілу ресурсів проекту тощо. На цьому етапі приймаються такі рішення на основі інформації, отриманої на попередній кроках описаного методу.

При цьому процедура прийняття рішень обов'язково повинна враховувати критерій достатності процесу тестування як відправну точку для прийняття рішень. Використання цього критерію робить процес прийняття рішень більш обґрунтованим порівняно з існуючими моделями надійності та процедурами прийняття рішень, описаними, наприклад, в [1]. Кількісно мірою процесу прийняття рішень може бути кількість залишкових помилок в системі, середній час до виявлення наступної помилки, імовірність безвідмовної роботи тощо та відповідні інтервали довіри цих величин. Крім того процес виявлення помилок різних типів (критичних, блокуючих, тривіальних тощо) може бути описаний на основі тієї ж моделі [2] зі своїми числовими значеннями параметрів з використанням того самого критерію достатності процесу тестування по кожному типу помилок. В такому разі процес прийняття рішень може базуватись, наприклад, на залишковій кількості критичних помилок, імовірності безвідмовної роботи стосовно блокуючих помилок і т. ін.

2. ПРИКЛАД ЗАСТОСУВАННЯ ЗАПРОПОНОВАНОГО МЕТОДУ НА ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ

Для ілюстрації використання запропонованого методу та з метою порівняння моделей надійності ПЗ використаємо експериментальні дані, описані в [1]. Програмним продуктом є система управління реального часу, розроблена Bell Laboratories. Дані тестування є результатом виявлених помилок під час тестування системи протягом 25 годин процесорного часу. В роботі [1] для опису цих даних була застосована модель негомогенного пуассонового процесу Goel–Okumoto [11] завдяки її простоті та широкому використанню при дослідженні надійності ПЗ.

Крок 1: Початкові дані були представлені у вигляді часу між помилками. Оскільки як модель [2], так і модель Goel–Okumoto [11]

передбачають роботу з даними у вигляді кількості помилок на часових інтервалах, а також з метою усунення можливих залежностей між цими величинами, дані були представлені у вигляді кількості помилок за годину процесорного часу [1]. Ці початкові експериментальні дані наведені в табл. 1. (Зауважимо, що згідно з критерієм, наведеним при описі методу кількість інтервалів доцільно обрати не більше 10, оскільки максимальна виявлена кількість помилок становить 136, однак для максимально коректного порівняння отриманих результатів з результатами [1], надалі будемо працювати з 25 інтервалами, як наведено в [1]).

Таблиця 1. Вхідні дані тестування програмного продукту

Попередні оцінки даних Час, год.	Кількість помилок	Кумулятивна кількість помилок
1	27	27
2	16	43
3	11	54
4	10	64
5	11	75
6	7	82
7	2	84
8	5	89
9	3	92
10	1	93
11	4	97
12	7	104
13	2	106
14	5	111
15	5	116
16	6	122
17	0	122
18	5	127
19	1	128
20	1	129
21	2	131
22	1	132
23	2	134
24	1	135
25	1	136

Вважаємо доцільним зробити ще одне зауваження стосовно експериментальних даних. З табл. 1, на нашу думку, видно, що процес тестування ще не можна вважати завершеним, оскільки навіть на 25-й годині тестування було виявлено помилку, і функція кумулятивної кількості помилок ще не виходить на насичення, що може впливати на результати оцінювання та прогнозування надійності ПЗ згаданими моделями.

Графічна залежність кумулятивної функції помилок від часу наведена у вигляді точок на рис. 2.

Крок 2: Згідно з описаними рекомендаціями встановимо в якості початкових наближень наступні значення: $\alpha_0 = 150$, $\beta_0 = 0,04$, $s_0 = 0,35$ оскільки ми не маємо докладних відомостей про програмний проект.

Попередні оцінки даних тестування дозволяють припустити, що цей проект можна віднести до категорії невеликих через специфіку процесу виявлення помилок, а саме – більшість помилок виявляється вже на перших етапах тестування, тобто інтенсивність виявлення помилок практично строго спадає з часом, в той час як для великих і дуже великих проектів характерною рисою є початкова невелика інтенсивність виявлення помилок, яка поступово досягає свого максимуму, а потім строго спадає.

Крок 3: З використанням методу максимальної правдоподібності знаходимо розв'язок системи рівнянь для отримання точкових оцінок параметрів моделі [2]. Отримані після розв'язку системи точкові значення параметрів становлять $\hat{\alpha} = 150,0$, $\hat{\beta} = 0,126$, $\hat{s} = 0,120$. Зауважимо, що значення індексу величини проекту (параметр s) підтверджує припущення про невелику величину проекту. Прогнозована загальна кількість помилок в програмній системі становить $\hat{\alpha} \cdot \hat{s} \cdot \Gamma(\hat{s}) = 142$, що опосередковано підтверджує припущення про недостатність процесу тестування.

У роботі [1] були отримані наступні значення параметрів моделі Goel–Okumoto для опису цих експериментальних даних: $\hat{\alpha} = 142,32$, $\hat{\beta} = 0,1246$. В цій моделі прогнозована загальна кількість помилок дорівнює параметру $\hat{\alpha}$ і становить 142,32, що узгоджується з результатами моделі [2]. Близькість значень параметрів $\hat{\beta}$ моделі [2] та $\hat{b}$ моделі Goel–Okumoto пояснюється їх спільним фізичним змістом: кількість помилок виявлених на помилку за годину [1].

Графіки функцій кумулятивної кількості помилок для моделі [2], моделі Goel–Okumoto з отриманими значеннями параметрів та експериментальні точки наведені на рис. 2 (крива 1, крива 2 та точки відповідно).

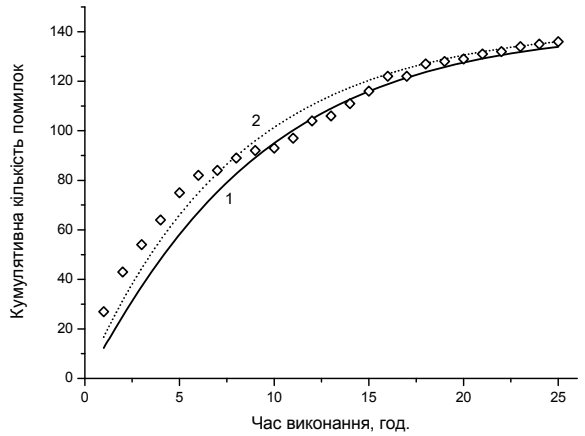

Рис. 2 – Кумулятивна кількість помилок в залежності від часу тестування для моделі [2] (крива 1), моделі Goel–Okumoto (крива 2). Точки – експериментальні значення (табл. 1)

Крок 4: Тест Колмогорова–Смірнова дає статистичне порівняння між експериментальним даними та моделлю, отриманою на кроці 3 і використовується для перевірки адекватності моделі. Обидві моделі успішно проходять такий тест, що підтверджує, що вони адекватно описують експериментальні дані з табл. 1 [1]. З рис. 2 також можна помітити, адекватність обох моделей для опису кумулятивної кількості помилок в системі. Значення квадрату коефіцієнту кореляції Пірсона (R2) та середнього квадратичного відхилення (Δ^2) наведено в табл. 2. Як видно з табл. 2 середнє квадратичне відхилення є майже вдвічі меншим у випадку використання моделі Goel–Okumoto, однак коефіцієнт кореляції Пірсона є помітно ближчим до одиниці у випадку використання моделі [2], що дозволяє зробити висновок про більш адекватний імовірнісний опис експериментальних даних моделлю [2], зауваживши також коректність моделі Goel–Okumoto та можливість її використання для опису цього процесу.

Таблиця 2. Параметри опису експериментальних даних дослідженими моделями

Модель	R2	Δ^2
Модель Goel–Okumoto	0,982	35,5
Модель з індексом величини проекту 2]	0,985	70,9

Крок 5: Для перевірки достатності процесу тестування точкові оцінки параметрів моделі були отримані після 5, 10, 15, 20, 21, 22, 23 та 24 годин тестування. Графік, отриманий в результаті чисельного диференціювання отриманої залежності параметру $\hat{s}$ від часу припинення процесу тестування наведено на рис. 3.

Як видно з рис. 3 залежність похідної параметру $\hat{s}$ по часу (критерію достатності процесу тестування) далекий від досягнення критерію достатності, згідно з яким ця залежність повинна прямувати до нуля [6]. Таким чином цей факт можна вважати прямим підтвердженням припущення про те, що процес тестування досліджуваного програмного продукту ще не завершений, і рішення, прийняті на цьому етапі можуть містити істотну помилку. Ще раз зауважимо, що інші моделі не дають такого критерію, який, на нашу думку, є важливим засобом підтримки в процесі прийняття рішень стосовно етапу тестування програмних продуктів.

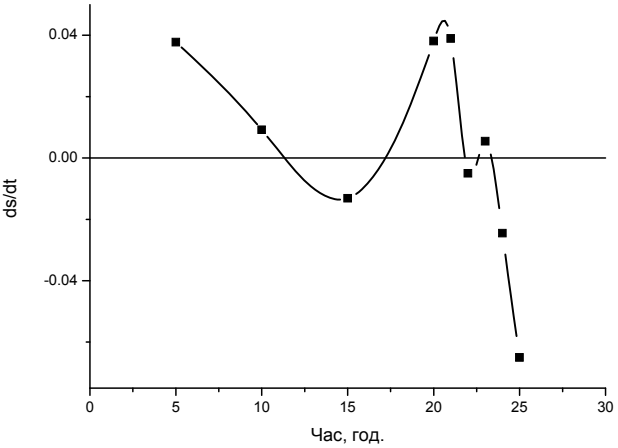

Рис. 3 – Залежність критерію достатності процесу тестування від часу припинення тестування

Крок 6: Розраховуємо згідно виразів, наведених в [7] значення таких показників надійності як імовірність безвідмовної роботи на час $t = 26$ год. $(P(26))$, середній час напрацювання на відмову m_t, дисперсію середнього часу напрацювання на відмову D_t та коефіцієнт готовності станом на час $t = 26$ год. $(K_\Gamma(26))$. Значення цих показників надійності наведено в табл. 3.

Таблиця 3. Значення показників надійності досліджуваного ПЗ після тестування протягом 25 год.

Показник	Значення
$P(26)$	$5 \cdot 10^{-4}$
m_t	0,125 год.
$\sigma_t = \sqrt{D_t}$	1,2 год.
$K_\Gamma(26)$	$1{,}2 \cdot 10^{-3}$

Як видно з табл. 3, недостатність процесу тестування також підтверджується значеннями

кількісних показників надійності. Так, імовірність безвідмовної роботи вже наприкінці наступної години після завершення процесу тестування є дуже мала, що підтверджується середнім часом напрацювання на відмову (0,125 год.), а коефіцієнт готовності станом на час $t = 26$ год. є, відповідно, вкрай низьким. Все це свідчить про те, що з великою імовірністю принаймні одна помилка буде виявлена протягом першої ж години за умови експлуатації чи продовження тестування програмного продукту.

Для ілюстрації, в якості одного з показників надійності, в роботі [1] була побудована залежність прогнозованої кількості залишкових помилок від часу тестування (у вигляді $f(t) = \mu(\infty) - \mu(t)$). Така залежність за даними [1] моделі Goel–Okumoto та даними, отриманими для моделі [2] наведена на рис. 4.

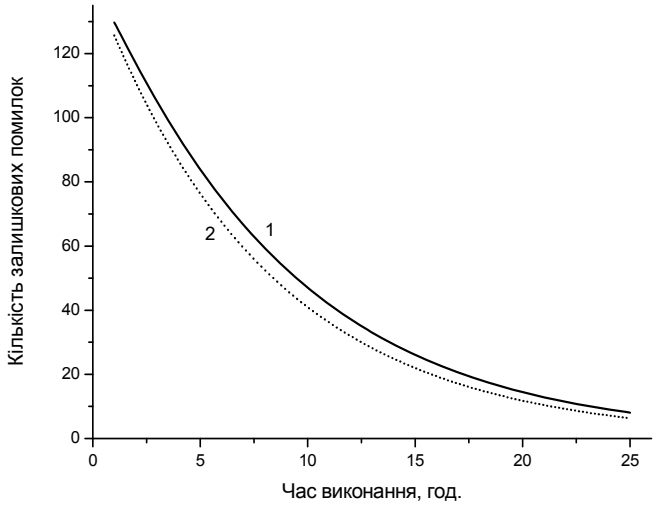

Рис. 4 – Залежність оцінки залишкових помилок в програмному продукті від часу тестування (крива 1 – модель з індексом величини проекту [2], крива 2 – модель Goel–Okumoto)

Як видно з рис. 4 залежності прогнозованої кількості залишкових помилок для обох моделей дуже схожі, однак відмінності у формі кривої кумулятивної кількості помилок приводять до того, що модель [2] дає дещо вищий прогноз залишкової кількості помилок. Таким чином, якщо оцінка та прийняття рішень робиться тільки на основі такої залежності без аналізу параметрів моделі та реальної кількості виявлених помилок, використання кривої 2 на рис. 4 може привести до похибки на 2 залишкових помилки, що становить 25 %. (Зауважимо, що використання для аналізу параметрів моделі – обидві моделі прогнозують наявність 142 помилок в системі, при виявлених в процесі тестування 136 – дозволяє зробити правильні висновки в обох випадках).

В цілому можна зробити висновок, що у цьому випадку обидві моделі адекватно описують експериментальні дані і можуть бути використані для опису поведінки помилок, а також для прогнозування процесу виявлення помилок. Отримана при цьому інформація може бути використана для планування, перерозподілу ресурсів та інших управлінських рішень, як описано нижче.

Крок 7: Описаний вище метод може бути використаний для опису поведінки процесу появи помилок, а також для визначення додаткових ресурсів для тестування та моменту готовності програмної системи до вводу в експлуатацію. Інформація цього плану доступна на різних часових проміжках і не обов'язково очікувати завершення процесу тестування, як вже було описано вище. Для прикладу в [1] розглядається випадок, якщо б процес тестування був припинений після 16 годин. В такому випадку було б виявлено тільки 122 помилки (див. табл. 1), а точкові оцінки параметрів моделі Goel–Okumoto мали б наступні значення $\hat{\alpha} = 138{,}37$, $\hat{\beta} = 0{,}133$ [1]. В такому випадку автори [1] роблять висновок, що прогнозована залишкова кількість помилок становить 16,37.

Використання моделі [2] в цій же ситуації дає наступні результати. Точкові оцінки параметрів моделі становлять $\hat{\alpha} = 158{,}8$, $\hat{\beta} = 0{,}114$, $\hat{s} = 0{,}108$; прогнозована загальна кількість помилок в програмній системі – $\hat{\alpha} \cdot \hat{s} \cdot \Gamma(\hat{s}) = 150{,}5$. Таким чином прогнозована залишкова кількість помилок становить 28,5.

Як бачимо, завдяки завищеній прогнозованій кількості залишкових помилок модель [2] може ввести в оману стосовно необхідних ресурсів для тестування програмного продукту, однак критерій достатності процесу тестування є чітким індикатором того, що до прогнозів моделі слід відноситись з обережністю, а процес тестування ще далекий від свого завершення і, можливо, процес виявлення помилок у цій фазі носить не пуассонів характер (див. напр. [12]).

На основі цих даних можна робити певні висновки стосовно програмного проекту, які повинні враховувати вимоги до надійності продукту. Так, наприклад, якщо критерієм уведення програми в експлуатацію є наявність менше певної кількості прогнозованих залишкових помилок чи середній час безвідмовної роботи більший деякого значення, на основі даних моделі можна отримати кількісне обґрунтування прийнятого рішення.

На практиці визначення часу випуску продукту в експлуатацію, перерозподілу ресурсів, проведення додаткового тестування і т.д. базується на значно складніших міркуваннях, ніж кількість залишкових помилок. Результати використання моделей надійності даються вхідну інформацію для процесу прийняття рішень стосовно програних проектів.

Таким чином, побудована в [2, 13] модель з динамічним показником величини проекту дає нову кількісну характеристику для підтримки процесу прийняття рішень при виробництві програмних продуктів – критерій достатності процесу тестування.

3. ВИСНОВКИ

В роботі розроблено метод оцінювання та прогнозування надійності програмного забезпечення на основі моделі надійності з динамічним показником величини проекту. Проведено дослідження цього методу на прикладі експериментальних даних комерційного програмного продукту та порівняння процесу прийняття рішень, заснованого на цій моделі порівняно з іншими моделями надійності.

Описаний метод може бути використаний для опису поведінки процесу появи помилок, а також для визначення додаткових ресурсів для тестування, моменту готовності програмної системи до вводу в експлуатацію та інших управлінських рішень.

Наявність критерію достатності процесу тестування в моделі з динамічним показником величини проекту, дала можливість модифікувати процес прийняття рішень при виробництві програмних продуктів та увести в нього нову кількісну характеристику, що робить цей процес більш обґрунтованим. Розроблений метод показує шляхи використання критерію достатності процесу тестування програмного забезпечення в процесі прийняття рішень при виробництві програмних продуктів.

На основі експериментальних даних процесу тестування комерційного програмного продукту показано переваги та особливості використання розробленого методу з детальною ілюстрацією процесу прийняття рішень при виробництві ПЗ.

4. СПИСОК ЛІТЕРАТУРИ

[1] Goel A.L. Software reliability models: assumptions, limitations, and applicability. *IEEE Transactions on software engineering* SE-11 (12) (1985). pp. 1411-1423.

[2] Чабанюк Я.М., Яковина В.С., Федасюк Д.В., Сенів М.М., Хімка У.Т. Побудова і дослідження моделі надійності програмного забезпечення з індексом величини проекту, *Інженерія програмного забезпечення* (2010). (в друці)

[3] Shooman M.L. Probabilistic models for software reliability prediction. *in Statistical Computer Performance Evaluation*. W. Freiberger Ed. New York: Academic, 1972. pp. 485-502.

[4] Yamada S., Ohba M., Osaki S. S-shaped reliability growth modeling for software error detection. *IEEE Transactions on Reliability* R-32 (5) (1983). pp. 475-478.

[5] Тимошенко Ю.О., Дідковська М.В. Узагальнена модель негомогенного пуасонівського процесу для оцінювання надійності програмного забезпечення. *Проблеми програмування* (2-3) (2004). с. 480-489.

[6] Яковина В.С., Сенів М.М., Чабанюк Я.М., Федасюк Д.В., Хімка У.Т. Критерій достатності процесу тестування програмного забезпечення. *Вісник Національного університету "Львівська політехніка" Комп'ютерні науки та інформаційні технології* (2010). (в друці)

[7] 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* (50) (2008). pp. 1232-1247.

[8] Ермаков А.А. *Основы надежности информационных систем: учебное пособие*. ИрГУПС. Иркутск, 2006.

[9] Половко А.М., Гуров С.В. *Основы теории надежности*. БХВ-Петербург. СПб., 2006.

[10] Липатов И.Н. *Надежность функционирования автоматизированных систем. Конспект лекций*. Изд-во Пермского государственного технического университета. Пермь, 1996.

[11] Durand J.B., Gaudoin O. Software reliability modelling and prediction with hidden Markov chains. *Statistical Modelling* (5) 1 (2005). pp. 75-93.

[12] Волочій Б.Ю. *Технологія моделювання алгоритмів поведінки інформаційних систем*. Вид-во НУ "Львівська політехніка". Львів, 2004.

[13] D. Fedasyuk, M. Seniv, Y. Chabanyuk, U. Khimka. The Estimation and Prediction Model of the Software Reliability with the Project Size Index. *Proceedings of the Xth International Conference "Modern problems of radio engineering, telecommunications, and computer science"* Lviv-Slavske, Ukraine, 2010. pp.209-210.

***Максим Сенів**, асистент кафедри програмного забезпечення Національного університету "Львівська політехніка", наукові інтереси – оцінка надійності програмного забезпечення.*

***Віталій Яковина**, доцент кафедри програмного забезпечення Національного університету "Львівська політехніка"; кандидат фізико-математичних наук; наукові інтереси – надійність та безпека програмного забезпечення.*

***Ярослав Чабанюк**, професор кафедри обчислювальної математики та програмування Національного університету "Львівська політехніка"; доктор фізико-математичних наук; наукові інтереси – стохастична оптимізація в середовищах з марковськими та напівмарковськими переключеннями.*

***Дмитро Федасюк**, проректор, завідувач кафедри програмного забезпечення Національного університету "Львівська політехніка"; доктор технічних наук; наукові інтереси – автоматизація теплового проектування мікроелектронних систем, технології створення програмного забезпечення.*

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

THE METHOD OF SOFTWARE RELIABILITY EVALUATION AND PREDICTION BASED ON THE MODEL WITH DYNAMIC INDEX OF PROJECT SIZE

Maksym Seniv, Vitaliy Yakovyna, Yaroslav Chabanyuk, Dmytro Fedasyuk

Lviv Polytechnic National University 12 Bandery St., Lviv, 79013, Ukraine
e-mail: max1sudden@gmail.com, yakovyna@polynet.lviv.ua, yaroslav_chab@yahoo.com, fedasyuk@lp.edu.ua,
http://lp.edu.ua/IKN/pz/

Abstract: *The software reliability evaluation and prediction method based on the model with dynamic index of project size is examined in this article. The support decision making procedure during software production which is based on the adequacy criterion of software testing is developed. The example of developed method based on testing data of industrial software product is proposed.*

Keywords: *software reliability, software life cycle, reliability model, model assumption, software engineering.*

1. INTRODUCTION

In the modern stage of the software engineering development the requirements to software reliability are enhancing, appears the need in cost shortage in testing and accordingly in evaluation and prediction of developed software reliability. The adequate analysis and prediction of software reliability is not possible without taking into account the characteristics and parameters of real software product in reliability models.

The aim of this paper is to develop the method of the software reliability evaluation and prediction, and the decisions support procedures at the main stages of software life cycle on the basis of proposed model. The usage of method is illustrated by the example based on experimental testing data of industrial software product.

2. THE METHOD OF THE SOFTWARE RELIABILITY EVALUATION AND PREDICTION BASED ON DEVELOPED MODEL

The general procedure of choosing the model of software reliability and the process of decision-making is overviewed in [1]. The procedure is general and permissive, not tied to the concrete model, and gives the general recommendations about the choosing process of software evaluation reliability model.

We consider the software reliability evaluation and prediction and the procedure of decision-making during the software product development based on the model [2]. The scheme of this method is shown in Figure 1 and consists of seven steps.

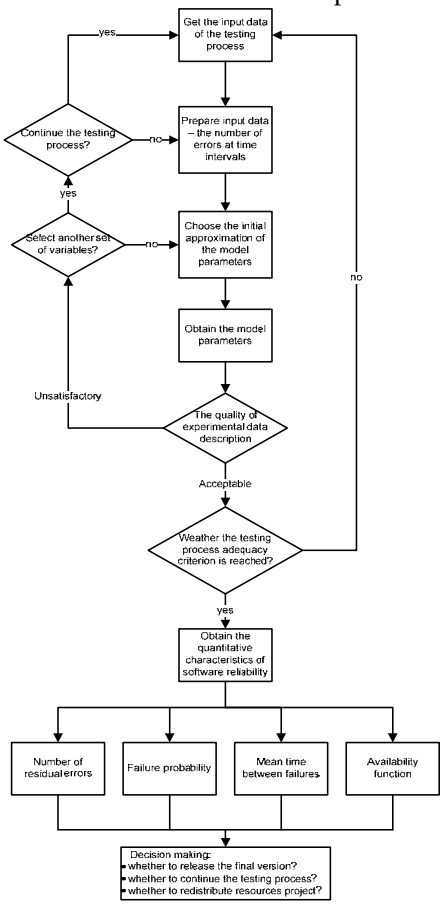

Fig. 1 – The illustration of the software reliability evaluation and prediction method

3. THE EXAMPLE OF PROPOSED METHOD USAGE ON THE BASIS OF EXPERIMENTAL DATA

We apply experimental data, which are described in [1], for illustration of proposed method usage and for comparison of the software reliability models. The program product is the management system in a real time, which was invented by Bell Laboratories. The testing data are the outcomes of determined faults during the 25-hours system testing in processor time. In the paper [1] Goel–Okumoto non-homogeneous Poisson process model [3] has been used due to its simplicity and wide use in software reliability research.

As we see, by means of too high prediction quantity of remained failures the model [2] can mislead concerning needed resources for program product testing. However, the adequacy criterion of software testing is the vivid indicator of the fact that the model's predictions should be treated with caution, and the testing process is not finished, besides, it is possible that on this stage the process of failure determination has not Poisson characteristic [4].

On the basis of these data we can make conclusion about the program project, which should consider the requirements of product's reliability. So, as if the input criterion of the program in realization is the presence of some less number of predicted remain failure or average time of non-stop running is more than some meaning. We can get quantitative confirmation of made solution, based on the data of the model.

At practice, the time determination of product realization, the resource redistribution, the performance of additional testing and etc. are basing on more significant thoughts than the number of remain failure. The results of reliability model usage give input information for making decisions about program projects.

So, built in [2,4] model with dynamic index of project size gives quantitative characteristic for support process of making decisions in program product production – the adequacy criterion of testing process.

4. CONCLUSIONS

In this paper we developed of software reliability evaluation and prediction method based on the reliability model with dynamic index of project size. We researched this method on the example of experimental data of the commercial program product and the comparison of decision-making process, based on this model comparing with other reliability models.

The described method can be used for description of failure appearance process behavior, and for determining of additional testing resources, readiness time of system for industrial realization and making other managerial decisions.

The presence of adequacy criterion of testing process in model with dynamic index of project size gives the opportunity to modify the decision-making process at program product production and to insert a new quantitative characteristic, which makes this process more justified. Developed method illustrates more ways of using of adequacy criterion of software testing process in the decision-making process at program product production.

On the basis of experimental data of testing process of commercial program product the advantages and peculiarities of developed method usage with detailed illustration of making-decision process at software production have been shown.

5. REFERENCES

[1] Goel A.L. Software reliability models: assumptions, limitations, and applicability. *IEEE Transactions on software engineering* SE-11 (12) (1985). pp. 1411-1423.

[2] Y. Chabanyuk, V. Yakovyna, D. Fedasyuk, M. Seniv, U. Khimka. Development and Estimation Model of the Software Reliability with the Project Size Index. *Software engineering* (2010). (in press)

[3] Durand J.B., Gaudoin O. Software reliability modelling and prediction with hidden Markov chains. *Statistical Modelling* (5) 1 (2005). pp. 75-93.

[4] D. Fedasyuk, M. Seniv, Y. Chabanyuk, U. Khimka. The Estimation and Prediction Model of the Software Reliability with the Project Size Index. *Proceedings of the Xth International Conference "Modern problems of radio engineering, telecommunications, and computer science"* Lviv-Slavske, Ukraine, 2010. pp. 209-210.