

РОЗРОБКА МЕТОДУ ОЦІНЮВАННЯ І ПРОГНОЗУВАННЯ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВРАХУВАННЯМ КРИТЕРІЮ ДОСТАТНОСТІ ПРОЦЕСУ ТЕСТУВАННЯ

В роботі описано розробку методу оцінювання та прогнозування надійності ПЗ на основі кількості виявлених помилок та процедуру підтримки прийняття рішень на основних стадіях життєвого циклу програмного забезпечення (ПЗ), базуючись на моделі надійності ПЗ з динамічним показником складності проекту. Застосування методу ілюструється прикладом з використанням експериментальних даних тестування промислового програмного продукту.

This paper describes the development method of assessment and prediction of software reliability based on the number of mistakes and how to ppe promotional decisions on major life-cycle of software (SW), based on software reliability model with a dynamic metric complexity of the project. Application of the method is illustrated using an example of experimental data testing of industrial software.

1. ВСТУП

Розвиток інформаційних технологій та комп'ютерної техніки, зокрема поширення її в усіх сферах людської життєдіяльності, ставить щоразу складніші завдання перед розробниками програмного забезпечення. Програмні продукти стають дедалі складнішими, багатокomпонентними і вимагають спеціалізованого підходу для їх розроблення.

Разом з цим підвищуються і вимоги до надійності програм, виникає потреба у скороченні затрат на тестування та, відповідно, у прогнозуванні надійності розроблюваного програмного забезпечення (ПЗ).

В багатьох дослідженнях поняття надійності ПЗ виокремлюють, тому, що при застосуванні понять надійності до програмних засобів варто враховувати особливості і відмінності цих об'єктів від традиційних технічних систем, для яких спершу розроблялася теорія надійності. Принципова відмінність програмного забезпечення від апаратного полягає в тому, що програма у процесі використання не зношується, а навпаки, з плином часу виявляються і виправляються помилки, які не

⁹ Національний університет «Львівська політехніка»

були знайдені раніше, тому ПЗ з часом вдосконалюється і покращується.

Для розв'язання задач оцінювання та прогнозування надійності ПЗ в даний час використовуються відповідні до ситуації моделі надійності. Модель надійності ПЗ – це математична модель, побудована для визначення залежності характеристик надійності програмного забезпечення від певних параметрів. Ці моделі можна класифікувати за різними ознаками. Зараз широко використовують тип моделей, які описують поведінку ПЗ на основі кількості виявлених помилок у програмі.

В основу значної кількості моделей надійності ПЗ, предметом дослідження яких є кількість помилок у визначеному часовому інтервалі, покладено розподіл Пуасона. Використання цього розподілу випадкових величин добре зарекомендувало себе в багатьох областях, де основна зацікавленість полягає в обчисленні кількості незалежних подій протягом інтервалів часу.

Але в моделях цього типу або зовсім не враховується такий показник, як складність програмного проекту, або враховується тільки в часткових випадках (має апріорно встановлені значення, не є параметром моделі). Тому існує потреба в розробленні засобів для прогнозування надійності ПЗ з використанням показника складності програмного проекту.

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

Метою цієї роботи є розробка методу оцінювання і прогнозування надійності програмного забезпечення, який би враховував значення критерію достатності процесу тестування та давав змогу зменшити похибку при визначенні кількісних характеристик надійності ПЗ.

2. РОЗРОБКА МЕТОДУ ОЦІНЮВАННЯ І ПРОГНОЗУВАННЯ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.

Загальна процедура підбору моделі надійності ПЗ та процесу прийняття рішень наведена в роботі [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.

Розгляньмо 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] використовується кількість помилок на певних часових інтервалах. Однак перед початком застосування моделі слід забезпечити дотримання 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 ($k \leq 5 \cdot \lg n$, де n – загальна кількість виявлених помилок в процесі тестування).

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 чи процесорний час [1]. Для аналізу вхідних даних щодо придатності зазначеної моделі надійності доцільно 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 ролі 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 отримують як розв'язки системи рівнянь [2] 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 ролі початкового наближення параметру α , можна вибрати значення кількості помилок, виявлених від початку процесу тестування;

Як початкове наближення параметру β , можна вибрати обернене значення тривалості процесу тестування у вибраній часовій шкалі;

При виборі початкового наближення параметру S можна скористатися рекомендаціями щодо співвідношення цього параметру з величиною програмного проекту [2], і в ролі 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. Обчислення параметрів моделі.

Для визначення параметрів моделі в загальному можна використати різні методи в залежності від природи наявних даних [1]. Тут 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 Ньютонів з початковими наближеннями, отриманими на кроці 2. Після розв'язку системи рівнянь, обчислені 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 визначити, 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 певному 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 між експериментальною вибіркою та теоретичною вибіркою, отриманою з використанням моделі в точках, які відповідають експерименту. Коефіцієнт кореляції (кореляція моментів добутків) використовується як показник характеру взаємного стохастичного впливу зміни двох випадкових величин, оскільки проводиться вимірювання змінних з інтервальною шкалою. Оскільки в наших дослідженнях 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.

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 названій 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.

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 застосувати критерій, описаний в праці [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 $t_1 < t_2 < \dots < T$), кожен з яких включає попередній, і для кожного проміжку незалежно отримують точкові оцінки параметрів моделі. Іншим випадком може бути визначення 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 ролі абсцисс будуть значення t_1, t_2, \dots, T , а відповідними ординатами – точкові оцінки параметру s , отримані з розв'язку системи рівнянь максимальної правдоподібності з експериментальними даними з відповідного часового інтервалу. Визначена 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. Обчислення 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, ймовірність 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 побудови 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, ймовірність 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.

3. ЗАСТОСУВАННЯ ЗАПРОПОНОВАНОГО МЕТОДУ НА ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ

Для ілюстрації використання запропонованого методу та з метою порівняння моделей надійності ПЗ використаємо експериментальні дані (Табл.1).

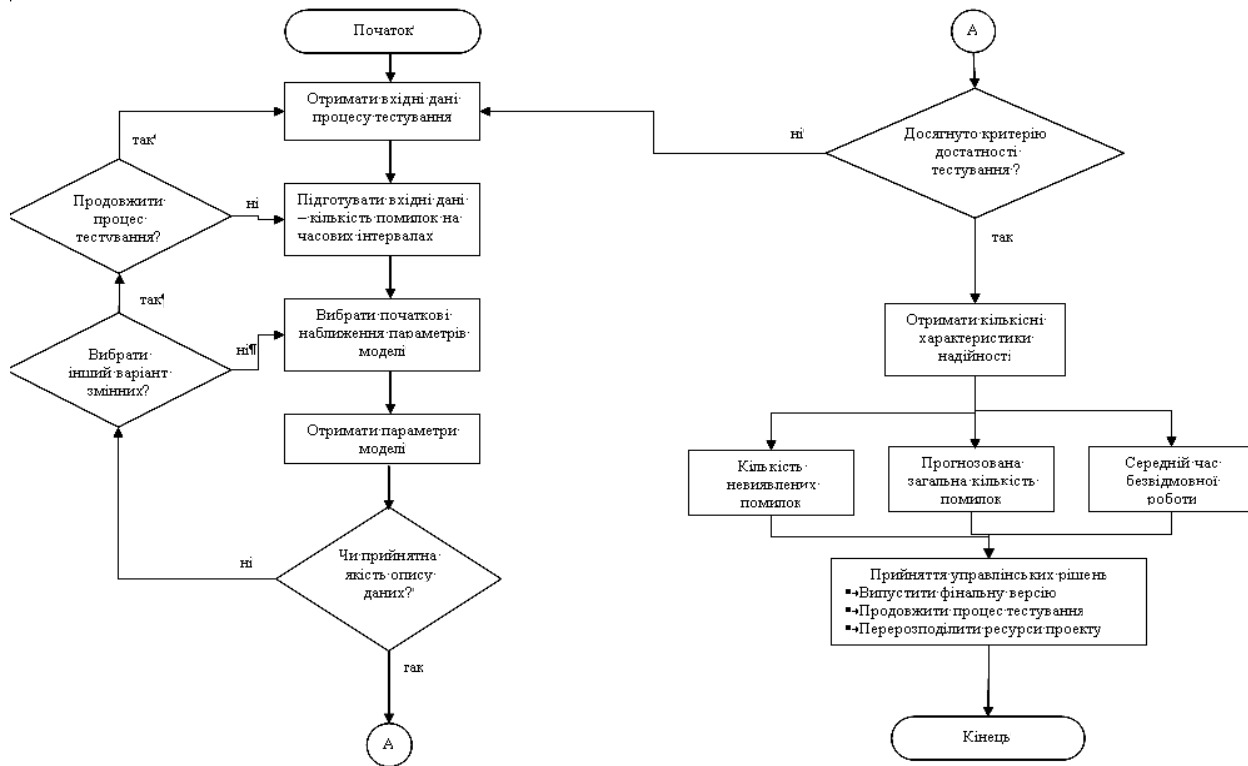


Рис. 1. Ілюстрація методу оцінювання та прогнозування надійності ПЗ

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.

Крок 1: Початкові дані були представлені у вигляді часу між помилками.

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.

Таблиця 1

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

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 $\alpha=150$, $\beta=0,126$, $s=0,120$. Зауважимо, що значення індексу величини проекту (параметр S) підтверджує припущення про невелику величину проекту. Прогнозована загальна кількість помилок в програмній системі становить α , β , $F(s)=142$, що опосередковано підтверджує припущення про недостатність процесу тестування.

У роботі [1] були отримані наступні значення параметрів моделі Goel–Okumoto для опису цих експериментальних даних: $\alpha=142,132$, $b=0,1246$. В цій моделі прогнозована загальна кількість помилок дорівнює параметру α і становить 142,32, що узгоджується з результатами моделі [2]. Близькість значень параметрів β моделі [2] та b моделі Goel–Okumoto пояснюється їх спільним фізичним змістом: кількість помилок виявлених на помилку за годину [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 (R^2) 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 s від часу припинення 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.

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 , дисперсію середнього часу напрацювання на відмову D_t та коефіцієнт готовності станом на час $t=26$ год. ($K_T(26)$).

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 $a=138,37$, $b=0,133$ [1]. В такому випадку автори [1] роблять висновки, що прогнозована залишкова кількість помилок становить 16,37.

Використання моделі [2] в цій же ситуації дає наступні результати. Точкові оцінки параметрів моделі становлять $a=158,8$, $\beta =0,114$, $s=0,108$; прогнозована загальна кількість помилок в програмній системі – $\alpha \cdot s \cdot \Gamma(s)=150,5$. 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 пусонові характер (див. напр. [12]).

На основі цих даних можна робити певні висновки стосовно програмного проєкту, 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

3. ВИСНОВКИ

В роботі розроблено метод оцінювання та прогнозування надійності програмного забезпечення з врахуванням значення критерію достатності процесу тестування, який дає змогу зменшити на 10% похибку при визначенні кількісних характеристик надійності ПЗ, зокрема кількості залишкових несправностей та коефіцієнта готовності ПЗ.

Розроблено процедуру обчислення кількісних показників надійності з урахуванням основних припущень моделі. Розроблений алгоритм розрахунку числових показників надійності ПЗ дає змогу розрахувати такі параметри надійності ПЗ, як час напрацювання на відмову, загальну кількість помилок, що є придатним для числової оцінки надійності розроблюваного ПЗ на виробництві. Значення критерію достатності процесу тестування ПЗ дає практичні рекомендації керівникам тестових підрозділів ІТ-компаній стосовно кількісної оцінки етапу тестування ПЗ.

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.

1. Goel A.L. *Software reliability models: assumptions, limitations, and applicability* // *IEEE Transactions on software engineering*. 1985, Vol. SE-11, No 12, 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 проєкту* // *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. – P. 485–502.
4. 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.

5. Тимошенко Ю.О., Дідковська М.В. Узагальнена модель негомогенного пуассонівського 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 НУ "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*. – Vol. 50. – 2008. – P. 1232–1247.

8. Ермаков А.А. Основы надежности информационных систем: учебное пособие. – Иркутск: ИрГУПС, 2006. – 151 с.

9. Половко А.М., Гуров С.В. Основы теории надежности. – СПб.: БХВ-Петербург, 2006. – 704 с.

10. Лунатов И.Н. Надежность функционирования автоматизированных систем. Конспект лекций. – Пермь: Изд-во Пермского государственного технического университета, 1996. – 66 с.

11. Durand J.B., Gaudoin O. Software reliability modelling and prediction with hidden Markov chains // *Statistical Modelling*. – Vol. 5 (1). – 2005. – P. 75–93.

12. Волочій Б.Ю. Технологія моделювання алгоритмів поведінки інформаційних систем. – Львів: Вид-во НУ "Львівська політехніка", 2004. – 220 с.