

найкращому випадку класифікація здійснює абсолютно правильне розбиття документів на дві групи: один клас відповідає природнім, інший – техногенним надзвичайним ситуаціям.

Результати застосування різних класифікаторів надано у табл. 1.

Таблиця 1

## Експериментальне дослідження класифікаторів

Вид класифікатора Показник	Статистичні класифікатори	Класифікатори, які засновані на функціях подібності	Нейромережні класифікатори	
			Багатошаровий перцептрон	Мережа Кохонена
Точність ( $V$ )	0,25	0,43	0,68	0,89
Повнота ( $U$ )	0,34	0,45	0,75	0,81

## Висновки

Отже, у роботі надано класифікатори текстових документів на підставі нейромережних моделей. Проведено їх дослідження для здійснення класифікації текстових документів, які свідчать, що розроблені моделі дають більшу точність та повноту результатів порівняно зі статистичними класифікаторами.

Наступним кроком дослідження є обґрунтування та вибір алгоритмів класифікації, їх можлива модифікація, що найкращим чином буде відповідати умовам автоматизації текстових документів.

## Література

1. Классификация и кластер: [сб. ст. / под ред. Дж. В. Райзина]. – М.: Мир, 1980.
2. Кочетков П. А. Краткий курс теории вероятностей и математической статистики: [учебное пособие] / Кочетков П. А. – М.: МГИУ, 1999. – 51 с.
3. Елисеєва И. И. Общая теория статистики: [учебник] / И. И. Елисеєва, М. М. Юзбашев; под ред. И. И. Елисеєвой. – [5-е изд., перераб. и доп.]. – М.: Финансы и статистика, 2004. – 656 с.
4. Солтон Дж. Динамические библиотечно-поисковые системы / Солтон Дж. – М.: Мир, 1979.
5. Осовский С. Нейронные сети для обработки информации / С. Осовский; [пер. с польского И. Д. Рудинского]. – М.: Финансы и статистика, 2002. – 344 с.
6. Круглов В. В. Искусственные нейронные сети / В. В. Круглов, В. В. Борисов. – М.: Горячая линия – Телеком, 2002. – 382 с.

Надійшла 19.3.2011 р.

УДК 004.413.5; 004.416.6

В.С. ЯКОВИНА, Я.М. ЧАБАНЮК, М.М. СЕНІВ, У.Т. ХІМКА

Національний університет "Львівська політехніка"

## ОЦІНЮВАННЯ ТА ПРОГНОЗУВАННЯ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МОДЕЛІ З ІНДЕКСОМ СКЛАДНОСТІ ПРОЕКТУ

*В роботі описано процес оцінювання та прогнозування надійності програмного забезпечення на основі моделі з індексом складності програмного продукту. За статистичними характеристиками опису експериментальних даних різними моделями встановлено, що модель з індексом складності продукту більш адекватно описує реальні експериментальні дані ніж S-подібна модель та модель Goel-Okumoto. Встановлено, що залежності параметрів моделі для різних тестових профілів є близькими між собою і виявляють однакову поведінку, що підтверджує ефективність використання цих параметрів для критерію достатності процесу тестування та визначення точки переходу вибірки вхідних даних до пуассонового розподілу.*

*This paper describes the process of evaluation and prediction of software reliability based on a model with software project complexity index. According to the statistical characteristics of experimental data description by the different models it has been determined that the model with product complexity index more adequately describes the real experimental data than the S-shaped model and the Goel-Okumoto model. It has been determined that the dependencies of the model parameters for different test profiles are close to each other and show the same behavior. It confirms the effectiveness of using these parameters for the testing process adequacy criterion and for the determination of the transition point of sample input data to Poisson distribution.*

Ключові слова: надійність програмного забезпечення, життєвий цикл програмного забезпечення, моделювання надійності, програмна інженерія.

## Вступ

Підвищення рівня складності сучасної техніки та все ширше її використання в усіх галузях людської життєдіяльності висувають все вищі вимоги до її надійності та експлуатаційної безпеки. Широке розповсюдження різноманітних програмно-апаратних комплексів та обчислювальних систем, що замінюють

виключно апаратні рішення, потребує окремого ретельного розгляду надійності програмного забезпечення (ПЗ).

Існує багато точок зору на означення надійності ПЗ та його кількісну оцінку [1]. Дехто вважає, що ця оцінка повинна бути бінарною за своєю природою, так що недосконала програма повинна мати нульову надійність, а досконала – одиничну. Ця точка зору проводить паралель з верифікацією програми, в якій програма є або коректною, або некоректною. Інша точка зору полягає в тому, що надійність ПЗ повинна бути означена через відносну частоту успішних запусків програми, протягом яких вона працює так, як очікується. Ця точка зору подібна до тієї, що лежить в основі тестування, де відсоток успішних випробувань використовується в якості міри якості програми.

Більш строгі визначення надійності ПЗ наведено в [1]. Нехай  $F$  – доволіно визначений клас помилок, а  $T$  – відносна міра часу, одиниці вимірювання якої визначаються конкретною програмою, надійність якої оцінюється. Тоді надійністю програмного продукту по відношенню до класу помилок  $F$  і відносно метрики часу  $T$  є імовірність того, що жодної помилки цього класу не виникне протягом виконання програми за наперед заданий відносний період часу.

Для розв'язання таких задач оцінки та прогнозування надійності на даний час використовують моделі надійності ПЗ [2– 4]. Модель надійності програмного забезпечення передбачає побудову математичної моделі для оцінки залежності надійності програмного забезпечення від певних параметрів. Зокрема, параметрами, що пов'язані з деякою гілкою програми на підмножині наборів вхідних даних, за допомогою яких ця гілка контролюється [4]. Іншими такими параметрами є частота помилок, які дозволяють оцінити якість систем реального часу що функціонують в безперервному режимі, і в той же час отримувати непряму інформацію про надійність ПЗ [4].

В основу значної кількості моделей надійності, предметом дослідження яких є кількість помилок у визначеному часовому інтервалі (при цьому вважають, що кількість помилок відповідає відомому стохастичному процесу з дискретною або неперервною інтенсивністю виявлення помилок, що залежить від часу), покладено розподіл Пуассона. Використання цього розподілу випадкових величин добре зарекомендувало себе в багатьох областях, де основна зацікавленість полягає в обчисленні кількості подій протягом інтервалів часу [1].

До таких моделей відносяться моделі: Shooman [5], Jelinski–Moranda [6], Schick–Wolverton [7], Musa [8], Schneidewind [9], Goel–Okumoto [10], S-подібна модель зростання надійності [11] тощо. Крім того нещодавно була представлена "узагальнена модель негомогенного пуассонівського процесу" [12]. Для підвищення точності цієї моделі автори [12] пропонують використовувати таку форму кривої інтенсивності виявлення несправностей, у якій введено додатковий параметр  $n$  для оцінки величини проекту, де вибір параметру  $n$  залежить від процесу проведення тестування. Основним недоліком моделей [5– 12] є те, що внаслідок припущень і спрощень, вони не достатньо повно відображають процес тестування, а їх результати не завжди збігаються з практичними [12, 13].

Авторами було побудовано модель надійності ПЗ [14, 15], яка відноситься до класу моделей на основі кількості помилок і узагальнює пуассонів розподіл на випадок дійсного індексу складності програмного продукту на відміну від існуючих моделей [5– 12]. Крім того, було розроблено метод оцінювання та прогнозування надійності програмного забезпечення на основі запропонованої моделі з дійсно значним індексом складності продукту [16].

Метою цієї роботи є аналіз ефективності застосування моделі і методу оцінювання та прогнозування надійності ПЗ на основі експериментальних даних тестування програмного продукту.

#### Опис моделі надійності програмного забезпечення.

В запропонованій моделі [14] вважається, що кількість виявлених помилок розподілена за законом Пуассона. Крім того індекс складності продукту є параметром моделі та визначається на основі експериментальних даних і набуває значення з дійсного діапазону і завжди більший від нуля.

Пропонується наступний вигляд функції інтенсивності виявлення несправностей:

$$I(t) = ab^{s+1}t^s \exp(-bt),$$

де  $a$  – коефіцієнт, що визначає загальну кількість помилок в ПЗ,  $b$  – коефіцієнт, що характеризує загальну тривалість процесу виявлення помилок,  $s$  – індекс складності продукту.

Для заданої функції інтенсивності функція кумулятивної кількості несправностей [11] має вигляд:

$$m(t) = \int_0^t I(t) dt = a \left[ -b^s t^s e^{-bt} + s \Gamma_{br}(s) \right],$$

де  $\Gamma_z(p) = \int_0^z t^{p-1} e^{-t} dt$ , ( $\text{Re } p > 0$ ) – неповна гама-функція. Зауважимо, що при  $s = 1$  функція

інтенсивності виявлення несправностей та кумулятивна функція співпадають з виглядом відповідних функцій S-подібної моделі [11].

Таким чином, загальна кількість помилок в ПЗ визначається через кумулятивну функцію при  $t \rightarrow \infty$  наступним співвідношенням:

$$m(\infty) = a s \Gamma(s),$$

де  $\Gamma(s)$  – гама-функція.

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

Проведені попередні дослідження дозволяють встановити наступні інтервали для індексу складності програмного продукту:

- при  $s \in (0; 0,7)$  програму можна вважати нескладною,
- при  $s \in [0,7; 1,5)$  програму можна вважати середньої складності,
- при  $s \in [1,5; 2,2)$  програму можна вважати складною,
- при  $s \in [2,2; e]$  – дуже складною.

Важливим прикладним аспектом моделей надійності ПЗ може стати встановлення кількісного критерію достатності процесу тестування програмного продукту, який би дозволив керівникам програмних проєктів більш обґрунтовано приймати рішення про виділення ресурсів на тестування та про завершення цього етапу розробки ПЗ. Такі дослідження на основі побудованої моделі проводились авторами в роботі [15].

В цій роботі показано, що на пізніх етапах тестування ПЗ, коли корельовані помилки виявлені та усунені, а кількість тих помилок, що виявляються на проміжку  $[t_i; t_{i+1}]$  розподілена за пуассонівським законом, якісна характеристика розподілу (параметр  $s$ ) вже практично не змінюється, а змінюються в основному кількісні характеристики (параметри  $a$  та  $b$ ), що дозволило формалізувати критерій достатності процесу тестування ПЗ наступним чином [15]:

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

Зауважимо, що усі інші моделі надійності ПЗ на основі розподілу Пуассона не дозволяють отримати такий критерій внаслідок відсутності необхідних параметрів моделей, які б описували якісну зміну форми інтенсивності виявлення помилок.

Запропонований критерій достатності процесу тестування можна використати у поєднанні із визначенням залишкової кількості помилок в програмному продукті (як і в інших моделях) і прийняти обґрунтоване рішення про розподіл ресурсів проєкту зі створення програмного продукту.

#### **Опис вхідних експериментальних даних**

З метою коректного порівняння ефективності використання моделі надійності з індексом складності програмного продукту в цій роботі використано дані другого експерименту, описаного в роботі [13], стосовно яких було зроблено висновок щодо невідповідності їх розподілу пуассоновому, а відповідно моделі, в основі яких лежить такий розподіл, недостатньо коректно описують поведінку експериментальних даних [13].

В якості об'єкту дослідження використовувався програмний засіб під назвою SESD (Software Environment for Software Data collection) [13]. Функціональність цього засобу полягала в граматичному аналізі, що використовується в середовищах програмування для збирання програмних даних. Цей програмний засіб містив 3559 рядків коду мовою C++, серед яких 3179 – виконуваного коду. Вхідними даними для засобу SESD є будь-яка програма мовою C. Для такого входу засіб створює п'ять вихідних значень: кількість рядків коду; загальна кількість використання операторів; загальна кількість використання операндів; кількість унікальних використань операторів; кількість унікальних використань операндів. Засіб SESD був реалізований одним програмістом [13], а потім був предметом незалежного тестування. Під час процесу тестування було виявлено і задокументовано 28 дефектів.

Для отримання експериментальних даних виявлені 28 дефектів були повторно введені в програму SESD. Після цього програма пройшла процес автоматизованого випадкового тестування з використанням платформи SRATE (Software Reliability Analysis, Testing, and Evaluation), розробленої авторами роботи [13]. Програма SESD без уведених 28 дефектів використовувалась в якості еталонної для оцінювання коректності отриманих результатів. Будь-яке відхилення між результатами виконання тестового випадку (test case) досліджуваною програмою та результатами виконання тестового випадку еталонною програмою розцінювалось як помилка програми. Один і тільки один дефект, що спричиняв помилку, вилучався з програми після виявлення помилки. Множина тестів містила 5477 тестових випадків, в якій один тестовий випадок відповідав одній програмі мовою C [13]. Ці програми мовою C були завантажені з мережі Інтернет. Під час тестування було виявлено, що 2 з 28 дефектів не виявлялись на даній множині тестів. Оскільки в реальних умовах індустрії програмного забезпечення процес тестування може бути припинено до виявлення усіх дефектів, в цьому експерименті кожен раунд тестування припинявся після виявлення 25 дефектів, незалежно, скільки часу це займало. Було проведено 40 раундів тестування. Для кожного раунду

створювався відповідний профіль тестування. Цей профіль являв собою деяку випадкову послідовність (з однорідним розподілом) вхідних файлів з множини тестів, які подавались на вхід програми SESD [13].

В якості часової шкали використовувалась кількість ітерацій (тестових випадків), як універсальна міра, що не залежить від зовнішніх факторів та умов тестування [1, 13]. В цій роботі, для використання розробленої моделі, експериментальні дані з роботи [13], які представляли у вигляді номерів ітерацій, на яких було виявлено помилки, були розбиті на інтервали по 10 ітерацій, для яких рахувалась кількість помилок на відповідному інтервалі, після чого будувалась кумулятивна функція помилок. Для досліджень, пов'язаних з визначенням критерію достатності процесу тестування, вхідні дані додатково опрацьовувались наступним чином:

- раунд тестування розбивали на фази, в даному випадку по 50 ітерацій;
- після кожних 50 ітерацій тестування на інтервалі  $(0, t_i]$  отримували точкові оцінки параметрів

моделі  $\mathcal{A}$ ,  $\mathcal{B}$  та  $\mathcal{E}$ .

Такі дії мали на меті імітувати використання моделі надійності на етапі тестування проектів з розробки програмного забезпечення, де проводиться певна кількість тестувань, після чого кожного разу приймається рішення про продовження чи достатність етапу тестування.

Дослідження проводились як для усереднених за усіма 40 раундами значень кількості помилок, так і для значень кількості помилок, отриманих протягом 10-го, 20-го, 30-го і 40-го раундів з метою встановлення статистичної однорідності отриманих результатів та точкових оцінок параметрів моделі. Крім моделі з індексом складності програмного продукту, для опису експериментальних даних використовували модель Goel– Okumoto [10] та S-подібну модель [11].

#### Опис та аналіз отриманих результатів

Першим кроком досліджень був опис кумулятивної функції помилок відповідними функціями моделей надійності ПЗ та отримання точкових оцінок параметрів моделей. Точкові оцінки параметрів моделей отримували методом максимальної правдоподібності. Точкові оцінки параметрів розглянутих моделей надійності та прогнозована кількість помилок при  $t = \infty$  для кожної моделі наведені в табл. 1. Як видно з цієї таблиці усі три моделі дають однаковий прогноз загальної кількості помилок в програмі і не дозволяють виявити помилки, які не проявляються при тестуванні. Таким чином ці результати не дають можливості порівняти якість опису експериментальних даних розглянутими моделями надійності ПЗ.

Таблиця 1

Точкові оцінки параметрів моделей.

Модель	$\alpha$	$\beta$	s	Прогноз помилок
Модель Goel– Okumoto	25,08	0,016	–	25,08
S-подібна модель	25,08	0,028	–	25,08
Модель з індексом складності продукту	26,75	0,013	0,137	25,08

Усереднене за 40 раундами тестування значення кумулятивної функції помилок та результати опису цієї функції різними моделями надійності ПЗ зображено на рис. 1. На цьому рисунку крапками зображено експериментальні дані, крива 1 відповідає кумулятивній функції моделі з індексом складності продукту, крива 2 – моделі Goel– Okumoto, крива 3 – S-подібній моделі. Як видно з рис. 1 на пізніх етапах тестування усі моделі практично однаково описують експериментальні дані, що узгоджується з результатами роботи [13]. Однак на початкових етапах тестування модель Goel– Okumoto, так само як і S-подібна модель, дещо гірше описують дані тестування реального ПЗ, на відміну від моделі з індексом складності продукту (крива 1).

Для статистичного оцінювання якості опису експериментальних даних моделями надійності ПЗ було розраховано квадрат коефіцієнта кореляції Пірсона ( $R^2$ ) та середньоквадратичне відхилення від експерименту для кожної моделі. Значення цих величин наведені в табл. 2.

Таблиця 2

Якість опису експериментальних даних дослідженими моделями

Модель	$R^2$	Середньоквадратичне відхилення
Модель Goel– Okumoto	0,974	0,137
S-подібна модель	0,995	0,105
Модель з індексом складності продукту	0,989	0,080

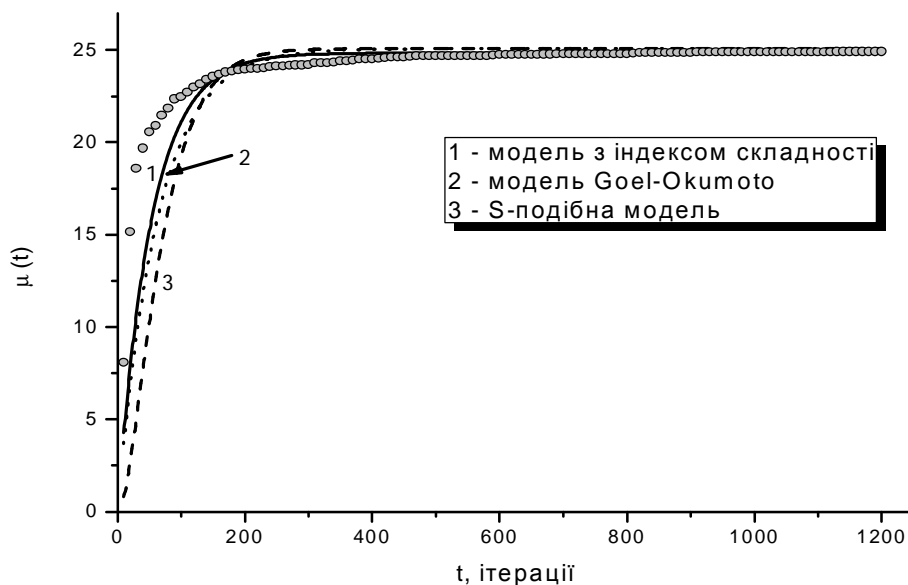
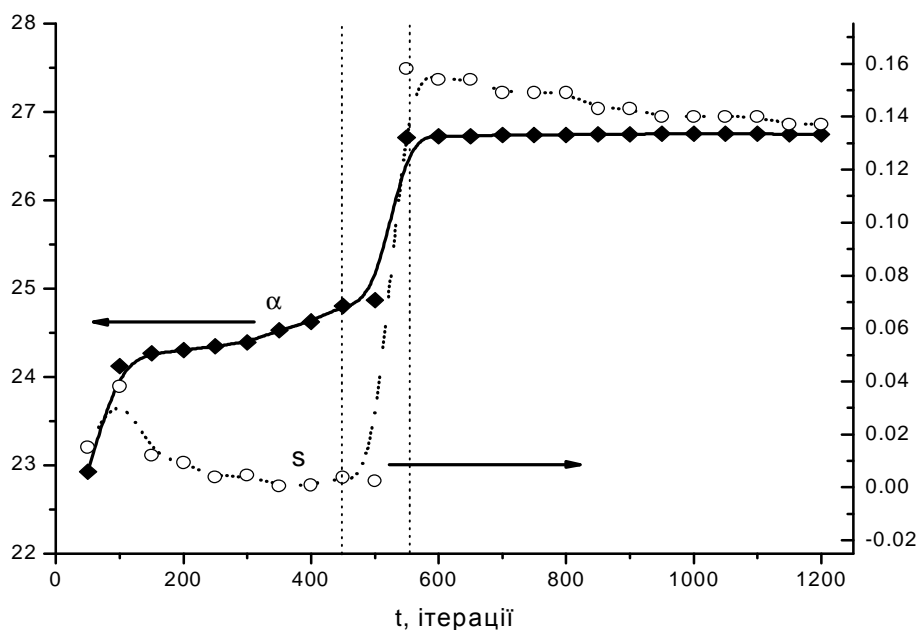


Рис. 1. Опис кумулятивної функції помилок різними моделями надійності

Як видно з табл. 2 найгіршими статистичними показниками якості опису експериментальних даних володіє модель Goel– Okumoto – квадрат коефіцієнту кореляції є найменшим (0,974), а середньоквадратичне відхилення найбільше (0,137). Модель з індексом складності продукту за середньоквадратичним відхиленням суттєво (більш ніж на 20 %) переважає S-подібну модель (0,080 проти 0,105), в той час як за значенням квадрату коефіцієнту кореляції дещо поступається S-подібній моделі (0,989 проти 0,995). Такі результати можна пояснити тим, що коефіцієнт кореляції, на відміну від середньоквадратичного відхилення описує узгодженість моделі з реальними даними з ймовірністю близькою до одиниці, в той час як середньоквадратичне відхилення характеризує апроксимацію моделі до статистичних даних в рівномірній метриці. Таким чином за статистичними характеристиками якості опису експериментальних даних різними моделями можна зробити висновок, що модель з індексом складності продукту більш адекватно описує реальні експериментальні дані порівняно з традиційними S-подібною та моделлю Goel– Okumoto.

Для ілюстрації та аналізу критерію достатності процесу тестування [15] були побудовані залежності параметрів трьох досліджуваних моделей надійності залежно від ітерацій, на яких було припинено тестування (рис. 2– 4). Побудова таких залежностей відповідає практиці проведення тестування, коли здійснюється певна ітерація (цикл тестування), після якої оцінюється якість програмного продукту. Пропонується після кожного такого циклу описувати отримані результати моделлю надійності, та визначати досягнення критерію достатності процесу тестування, який використовувати як важливий параметр при прийнятті рішень стосовно проекту.

Рис. 2. Залежність параметрів  $a$  і  $S$  моделі з індексом складності від тривалості процесу тестування

На рис. 2 зображено залежність параметрів  $a$  і  $s$  моделі з індексом складності від тривалості процесу тестування. Особливість залежності параметру  $a$  моделі з індексом складності (на відміну від інших моделей – рис. 4) в діапазоні 450–550 ітерацій добре корелює з висновками, які можна зробити з аналізу результатів роботи [13] – після 500 ітерацій кількість виявлених помилок реального ПЗ розподілена за Пуассоном, тоді як в області менших значень кількість помилок перевірка такої гіпотези дає негативний результат. Отже, залежність параметру  $a$  моделі з індексом складності, крім іншого, дозволяє встановити точку переходу до пуассонового розподілу кількості виявлених помилок. Так само як і в роботі [15], залежність індексу складності проекту від тривалості процесу тестування виявляє чітку особливість, яку було покладено в основу критерію достатності процесу тестування. Ця особливість полягає в тому, що при  $t \geq 550$  (що знову таки відповідає переходу до пуассонового розподілу часу виявлення помилок) значення  $s$  наближається до постійної величини, на відміну від такої залежності при  $t < 500$ . Залежність критерію достатності процесу тестування ( $ds/dt$ ) [15], отриману шляхом чисельного диференціювання даних з рис. 2, наведено на рис. 3 (б). Як бачимо така залежність наочно ілюструє критерій достатності процесу тестування ПЗ і показує наближення цього критерію до нуля при переході експериментальних даних до пуассонового розподілу.

На рис. 3 зображено залежність параметру  $b$  (а) та критерію достатності процесу тестування (б) моделі з індексом складності від тривалості процесу тестування (номеру ітерації, на якій було припинено тестування для отримання точкових оцінок параметрів моделі). Так само, як і в роботі [15] для інших експериментальних даних, параметр  $b$  моделі з індексом складності не може бути основою для визначення критерію достатності процесу тестування, оскільки його залежність від ітерацій тестування є практично гладкою функцією без особливостей. В цьому випадку також практично відсутня навіть будь-яка особливість залежності параметру  $b$  для запропонованої моделі в околі 500 ітерацій на відміну від параметру  $a$  (рис. 2). З рис. 2 та 3 видно, що на пізніх етапах тестування ПЗ, коли корельовано помилки виявлені та усунені, а кількість виявлених помилок, що залишилися відповідає пуассоному розподілу, якісна характеристика розподілу (параметр  $s$ ) практично не змінюється, а змінюються в основному кількісні характеристики (параметри  $a$  та  $b$ ).

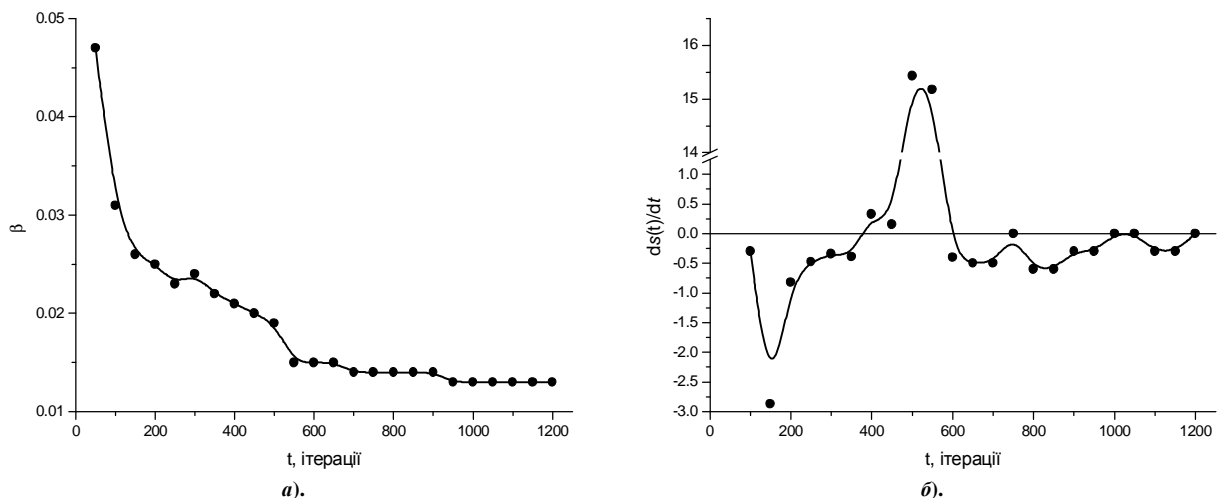


Рис. 3. Залежність параметру  $b$  (а) і критерію достатності процесу тестування (б) моделі з індексом складності від тривалості процесу тестування

На протипагу моделі з індексом складності, S-подібна моделі та модель Goel–Okumoto, залежності параметрів яких від тривалості процесу тестування наведені на рис. 4 (а) і (б) відповідно, не виявляють характерних особливостей, які можна було в використати в якості критерію достатності процесу тестування чи для визначення точки переходу до пуассонового розподілу вхідних величин. Як видно з рис. 4 залежності усіх параметрів обох моделей від тривалості процесу тестування є гладкими кривими, при цьому залежності параметрів моделі Goel–Okumoto проявляють деякий стрибок при  $t = 100$  (рис. 4 б) на відміну від параметрів S-подібної моделі (рис. 4 а), залежність яких є плавною в усьому дослідженому діапазоні.

Для встановлення однорідності і адекватності використання моделі з індексом складності при використанні на різних наборах тестових профілів було проведено дослідження параметрів моделі для різних раундів тестування та їх статистичний опис. На рис. 5 зображено залежності параметрів  $a$  (а) та  $s$  (б) моделі з індексом складності від тривалості процесу тестування для різних тестових профілів. Як видно з цього рисунку, усі залежності для різних раундів тестування є близькими між собою і проявляють однакові якісні залежності з різким стрибком в околі  $t = 500$ , що підтверджує припущення про ефективність використання цих параметрів для критерію достатності процесу тестування та визначення точки переходу

вибірки вхідних даних до пуассонового розподілу.

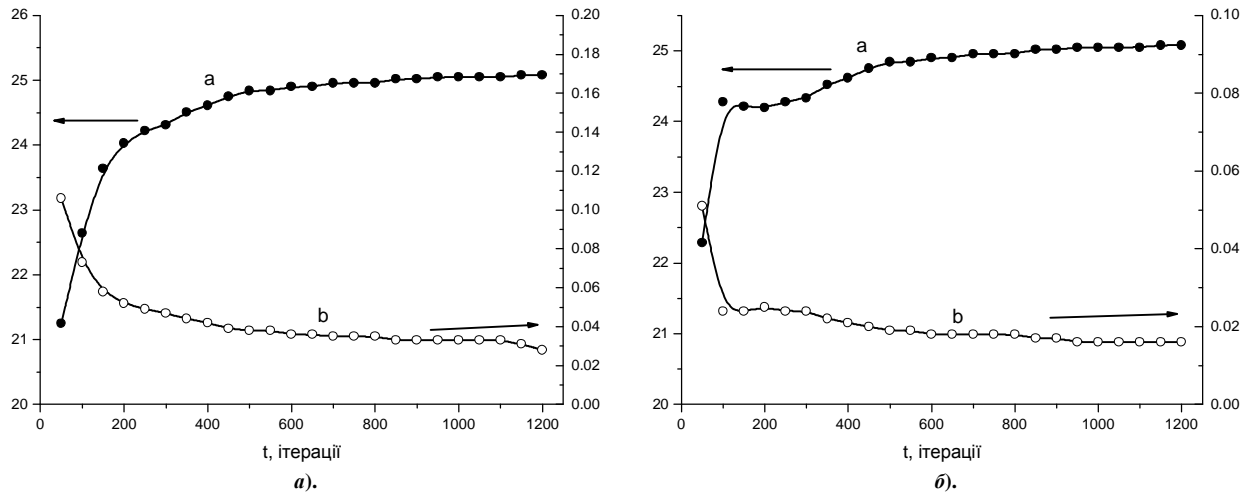


Рис. 4. Залежність параметрів S-подібної моделі (а) та моделі Goel-Okumoto (б) від тривалості процесу тестування

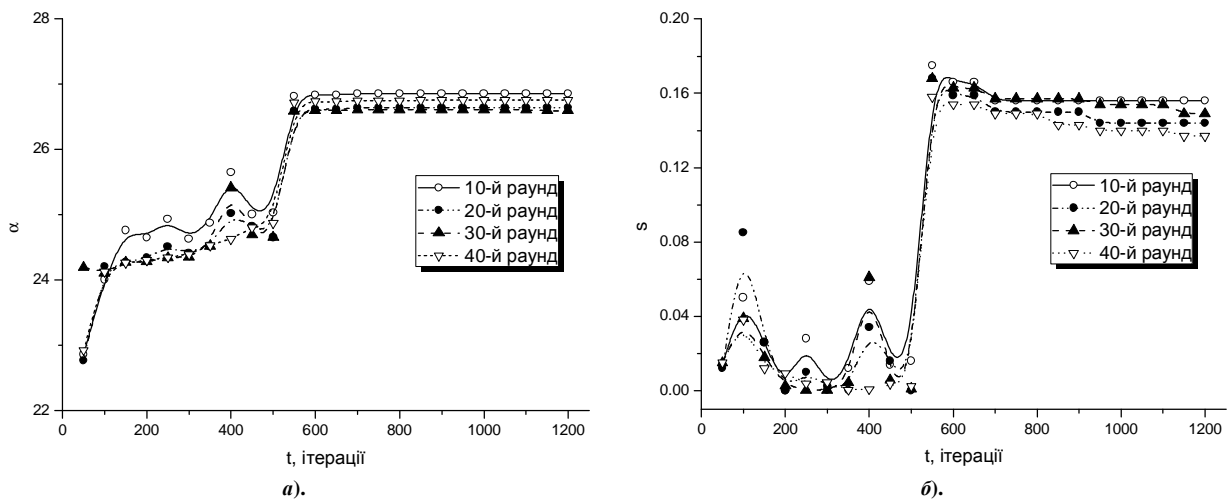


Рис. 5. Залежність параметрів  $\alpha$  (а) та  $s$  (б) моделі з індексом складності від тривалості процесу тестування для різних тестових профілів

Для кількісної характеристики однорідності розподілу точкових оцінок параметрів моделі з індексом складності розраховано статистичні характеристики такого розподілу. Основною характеристикою в цьому випадку є структурна характеристика вибірки – медіана. Вона практично визначає структуру вибіркових даних і визначається через ці дані. Цей показник має статус основного або головного при асиметричному розподілі даних, причому, у випадку асиметрії розподілу медіана бере на себе роль середнього значення, крім того, медіана вважається найбільш стійкою характеристикою вибірки, а тому може бути основою для критерію оптимального розподілу даних в інтервалах. Крім того розмах між мінімальним та максимальним значенням точкової оцінки кожного параметру не перевищує 15% (а у випадку параметру  $\alpha$  – 1%) з дуже низькими значеннями дисперсії та стандартного відхилення, що також свідчить про хорошу однорідність вибірки.

Таблиця 3

Статистичні характеристики розподілу точкових оцінок параметрів моделі для різних вхідних наборів тестових даних

Параметр	$\alpha$	$\beta$	$s$	$as\Gamma(s)$
Найбільше значення	26,849	0,015	0,156	25,08
Найменше значення	26,592	0,013	0,137	24,82
Розмах	0,257	0,002	0,019	0,26
Медіана	26,691	0,014	0,147	24,95
Середнє арифметичне значення	26,706	0,014	0,147	24,95
Дисперсія	0,014	$6,7 \cdot 10^{-7}$	$6,4 \cdot 10^{-5}$	0,013
Стандартне відхилення	0,12	$8,2 \cdot 10^{-4}$	$8,0 \cdot 10^{-3}$	0,11

Оскільки, середнє вибірки є незміщеною ефективною і правдивою оцінкою для математичного

сподівання випадкової величини та з близькості значень медіани і середнього арифметичного, можна зробити висновок, що дані є статистично однорідними. Це дозволяє використовувати нашу модель для прогнозування помилок одного типу окремо.

### Висновки

В роботі проведено є аналіз ефективності застосування моделі і методу оцінювання та прогнозування надійності ПЗ на основі експериментальних даних тестування програмного продукту.

Проведено порівняльний аналіз існуючих та розробленої моделей надійності ПЗ на реальних тестових прикладах та показано переваги моделі з індексом складності продукту над S-подібною моделлю та моделлю Goel– Okumoto. Показано, що на пізніх етапах тестування усі моделі практично однаково описують експериментальні дані, однак на початкових етапах тестування модель Goel– Okumoto, так само як і S-подібна модель, дещо гірше описують дані тестування реального ПЗ, на відміну від моделі з індексом складності. За статистичними характеристиками якості опису експериментальних даних різними моделями встановлено, що модель з індексом складності продукту більш адекватно описує реальні експериментальні дані.

Показано, що залежність параметру  $a$  моделі з індексом складності від тривалості процесу тестування дозволяє встановити точку переходу до пуассонового розподілу кількості виявлених помилок, в той час як залежність індексу складності проекту є основою критерію достатності процесу тестування. Показано, що на пізніх етапах тестування ПЗ, коли корельовано помилки виявлені та усунені, а кількість виявлення тих помилок, що залишились відповідає пуассоновому розподілу, якісна характеристика розподілу (параметр  $s$ ) практично не змінюється, а змінюються в основному кількісні характеристики (параметри  $a$  та  $b$ ).

Встановлено, що залежності параметрів моделі для різних тестових профілів є близькими між собою і проявляють однакові якісні залежності, що підтверджує припущення про ефективність використання цих параметрів для критерію достатності процесу тестування та визначення точки переходу вибірки вхідних даних до пуассонового розподілу.

Статистичні характеристики вибірки точкових оцінок параметрів моделі для різних тестових профілів показують високу однорідність даних, що підтверджує адекватність та ефективність використання моделі з індексом складності для різних наборів вхідних даних.

### Література

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. Половко А. М. Основы теории надежности / А. М. Половко. С. В. Гуров. – СПб.: БХВ-Петербург, 2006. – 704 с.
3. Липаев В. В. Надежность программных средств / Липаев В. В. – М.: СИНТЕГ, 1998. – 232 с.
4. Тейер Т., Липов М., Нельсон Э. Надежность программного обеспечения / Тейер Т., Липов М., Нельсон Э.; [пер. с англ.]. – М.: Мир, 1981. – 323 с.
5. Shooman M. L. Probabilistic models for software reliability prediction // in Statistical Computer Performance Evaluation. – W. Freiberger, Ed. – New York: Academic. – 1972. – P. 485– 502.
6. Jelinski Z. and P. Moranda Software reliability research // in Statistical Computer Performance Evaluation. – W. Freiberger, Ed. – New York: Academic. – 1972. – P. 465– 484.
7. Schick G. J. and R.W. Wolverton Assessment of software reliability // Proc. Oper. Res. – Physica-Verlag. – Wurzberg-Wien. – 1973. – P. 395– 422.
8. Musa J. D. A theory of software reliability and its application // IEEE Transactions on Software Engineering. – SE-1 (3). – 1975. – P. 312– 327.
9. Schneidewind N. F. Analysis of Error Process in Computer Software // Sigplan Note. – Vol. 10. – No.6. – 1975. – P.337– 346.
10. Goel A. L., Okumoto K. Time-Dependent Error-Detection Rate Model for Software and other Performance Measures // IEEE Transactions on Reliability. – Vol. R-28. – No. 3. – 1979. – P. 206– 211.
11. 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.
12. Тимошенко Ю. О., Дідковська М.В. Узагальнена модель негомогенного пуассонівського процесу для оцінювання надійності програмного забезпечення / Ю. О. Тимошенко, М. В. Дідковська // Проблеми програмування. – 2004. № 2– 3. – С. 480– 489.
13. K. – Y. Cai, D. – B. Hu, Ch. – G. Bai, H. Hu, T. Jing Does software reliability growth behavior follow a non-homogeneous Poisson process // Information and Software Technology. – Vol. 50. – 2008. – P. 1232– 1247.
14. Чабанюк Я. М. Побудова і дослідження моделі надійності програмного забезпечення з індексом величини проекту / Я. М. Чабанюк, В. С. Яковина, Д. В. Федасюк, М.М. Сенів, У.Т. Хімка // Інженерія програмного забезпечення. – 2010. – № 1 – С. 24– 29.
15. Яковина В. С. Критерій достатності процесу тестування програмного забезпечення / Яковина В. С., Сенів М. М., Чабанюк Я.М., Федасюк Д.В., Хімка У.Т // Вісник Національного університету "Львівська політехніка" Комп'ютерні науки та інформаційні технології. – 2010. – № 672 – С. 346– 358.
16. Сенів М. М. Метод оцінювання та прогнозування надійності програмного забезпечення на



Надійшла 15.3.2011 р.

УДК 004.932.2: 616-006.04

О.М. БЕРЕЗЬКИЙ, І.Г.ЦМОЦЬ

Тернопільський національний економічний університет, НУ"Львівська політехніка"

## ПРИНЦИПИ ПОБУДОВИ ТА АРХІТЕКТУРА КОМП'ЮТЕРНИХ ЗАСОБІВ АНАЛІЗУ І СИНТЕЗУ ЗОБРАЖЕНЬ

*В статті розглянуті принципи та структури побудови проблемно-орієнтованих комп'ютерних засобів аналізу та синтезу зображень. Проаналізовані рівні та види розробок проблемно-орієнтованих комп'ютерних засобів, а також шляхи підвищення ефективності їх використання.*

*The principles and structures constructions of the problem oriented computer tools analysis and synthesis of images are considered. Levels and types developments of the problem oriented of computer tools, and also paths of increase of efficiency of their use, are analyzed.*

Ключові слова: структура, принцип, проблемно-орієнтовані комп'ютерні засоби, аналіз, синтез, зображення.

### Постановка проблеми

Вимоги, що висуваються до комп'ютерних засобів аналізу та синтезу зображень за оперативністю, продуктивністю та ефективністю опрацювання даних не забезпечуються в рамках традиційних методів обчислення та принципів побудови універсальних комп'ютерних засобів. Одним із основних шляхів підвищення ефективності використання обладнання та забезпечення виконання перерахованих вимог є спеціалізація та проблемна орієнтація комп'ютерних засобів на клас задач аналізу та синтезу зображень. При спеціалізації та проблемній орієнтації здійснюється орієнтація структури, системи команд і принципів організації обчислень в комп'ютерних засобах на розв'язання задач аналізу та синтезу зображень.

В спеціалізованих і проблемно-орієнтованих комп'ютерних засобах аналізу та синтезу зображень алгоритми розв'язання задач залежно від вимог за швидкодією можуть бути реалізовані програмним, мікропрограмним або апаратним шляхом. Кожний із перерахованих видів реалізації алгоритмів має свої особливості, переваги і недоліки. В сучасних комп'ютерних засобах для розв'язання задач аналізу та синтезу зображень використовуються комбіновані підходи з перевагою одного з перерахованих видів реалізації алгоритмів. Даний вид реалізації визначається виходячи із забезпечення вимог конкретного застосування.

Для вибору комп'ютерних засобів аналізу та синтезу зображень для конкретного застосування використовується критерій ефективності використання обладнання  $E$ , який зв'язує продуктивність з витратами обладнання та дає оцінку комп'ютерним елементам за продуктивністю [2]. Кількісна величина ефективності використання обладнання для комп'ютерних засобів аналізу та синтезу зображень визначається наступним чином:

$$E = \frac{R}{t_{p3}W},$$

де  $R$  – складність алгоритмів розв'язання задачі аналізу та синтезу зображень, яка визначається кількістю елементарних арифметичних операцій;  $t_{p3}$  – час розв'язання задачі аналізу та синтезу зображень;  $W$  – витрати обладнання на реалізацію комп'ютерних засобів аналізу та синтезу зображень.

Розробка комп'ютерних засобів аналізу та синтезу зображень з високою ефективністю використання обладнання вимагає широкого використання сучасної елементної бази (напівзмовних і замовних НВІС, однокристальних процесорів обробки сигналів), розробки нових методів, алгоритмів і спеціалізованих структур, орієнтованих на реалізацію базових алгоритмів аналізу та синтезу зображень.

Тому актуальною проблемою є розробка комп'ютерних засобів аналізу та синтезу зображень з високою ефективністю використання обладнання.

### Аналіз останніх досліджень та публікацій

Аналіз останніх досліджень і публікацій у галузі розробки комп'ютерних засобів аналізу та синтезу зображень [1-9] показує, що такі засоби створюються на базі комп'ютерних систем з структурною організацією універсального типу, яка передбачає присутність:

- ієрархічної пам'яті, яка просто адаптується до різних задач з широким діапазоном кількості інформації, що використовується при їх розв'язанні та різними обсягами пам'яті необхідними для зберігання програм, вхідних даних, проміжних і кінцевих результатів обробки;
- користувацького інтерфейсу, що забезпечує зручність введення та виведення інформації, а також оперативний вплив на управління обчислювальним процесом;
- розвинутої системи програмного забезпечення, яка зменшує трудоемність підготовчих робіт, спрощує зв'язок та полегшує їх експлуатацію.