

УДК 004.891.3: 004.3

Т.О. ГОВОРУЩЕНКО

*Хмельницький національний університет, Україна*

## СИСТЕМА ПОВТОРНОГО ТЕСТУВАННЯ ПРИКЛАДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В статті запропонована система повторного тестування програмного забезпечення, а також способи опрацювання вхідної інформації та метод формування логічного висновку про необхідність та методику повторного тестування.

**повторне тестування прикладного програмного забезпечення (ПЗ), основне тестування прикладного ПЗ, приховані помилки ПЗ, рівень категорійності прихованих помилок ПЗ, система повторного тестування прикладного ПЗ, метод формування висновку про необхідність повторного тестування ПЗ, метод формування висновку про методику повторного тестування ПЗ**

### Вступ

Надійність роботи програмного забезпечення комп'ютерних систем є запорукою успішного вирішення задач, що розв'язуються системами, та довготривалої експлуатації цих систем. Основний спосіб досягнення надійності безвідмовного функціонування обчислювальної техніки та розв'язку прикладних задач з вірними результатами – це діагностування програмного забезпечення (ПЗ) на різних етапах його життєвого циклу. Взагалі, надійність програмного забезпечення визначається як властивість програми, яка виражається у виконанні заданих функцій у заданих умовах роботи і на заданій ЕОМ. Під помилкою ПЗ розуміють неправильність його функціонування, що призводить до хибного чи спотвореного стану комп'ютерної системи або спотворення обчислювального процесу [1].

**Формулювання проблеми.** Розвиток та впровадження нових архітектур та апаратна складність комп'ютерних систем на сьогодні випереджає розроблення методів і засобів діагностування програмного забезпечення.

Існуючі діагностичні програми та методи [1 – 5] не завжди повністю враховують зростаючі вимоги до розробки програм за причини постійного ускладнення ПЗ через постійне ускладнення архітектур.

Низька якість окремих діагностичних програм знижує ефективність використання існуючого ПЗ комп'ютерних систем. Відмови ПЗ можуть бути зумовлені прихованістю помилок. В цьому випадку помилка проявляється тільки в окремих комбінаціях, що рідко зустрічаються. Тому такі помилки виявляються тільки у процесі тривалої експлуатації ПЗ. Приховані помилки є найбільш небезпечними.

Отже, **головною метою дослідження** є розроблення методів і засобів підвищення ефективності тестування програмного забезпечення за рахунок виявлення прихованих помилок ПЗ у процесі повторного тестування. Повторне тестування проводиться як окремий технологічний процес після розробки і налагодження ПЗ.

### Вирішення проблеми. Система повторного тестування

Для прийняття рішення про необхідність повторного тестування та його методику в процесі дослідження була розроблена так звана система повторного тестування, на вхід якої подається звіт про основне тестування (метод тестування, операція тестування, тип виявленої помилки), а результатом роботи є висновок про необхідність та метод повторного тестування.

Структурна схема системи повторного тестування програмного забезпечення надана на рис. 1.

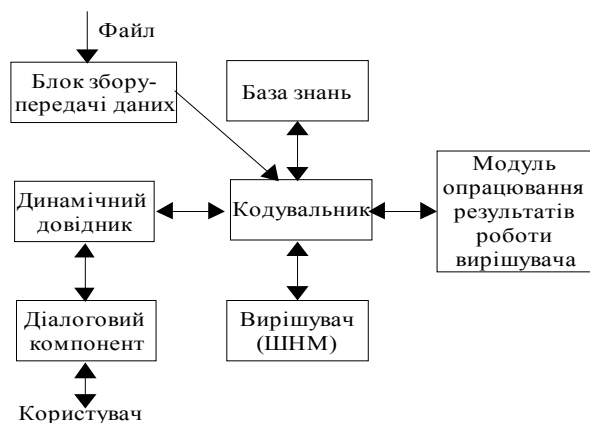


Рис. 1. Структурна схема системи повторного тестування програмного забезпечення

Система повторного тестування складається з наступних компонентів:

1) блок збору-передачі даних – підключає і зчитує інформацію з файлу користувача з результатами основного тестування, представленими у вигляді журналу “Метод тестування – Операція тестування – Тип виявленої помилки”;

2) кодувальник – виконує перетворення вхідних даних з лінгвістичної форми в кількісну форму, заповнення бази знань вхідними даними та формування вхідних векторів вирішувача. Перевіряє дані файлу на вірність та достатність при формуванні вхідних векторів; якщо дані недостатні або невірні, то кодувальник передає на динамічний довідник своє повідомлення з пропозицією сформувати ще один файл такої ж форми, як і попередній, з додатковими результатами, які перетворюються в кількісну форму аналогічно даним основного файлу, після чого заносяться в базу знань. Здійснює заповнення бази знань вихідними даними, перетворення результатуючих векторів вирішувача з кількісної в лінгвістичну форму та передачу їх на модуль опрацювання результатів роботи вирішувача;

3) база знань – містить таблиці з вхідними даними системи, допоміжні таблиці, а також правила

для формування висновку про необхідність та метод повторного тестування;

4) вирішувач – штучна нейронна мережа, на входи якої подається інформація про методи і операції основного тестування та типи виявлених під час основного тестування помилок, а на виході одержується рівень категорійності прихованих помилок;

5) модуль опрацювання результатів роботи вирішувача – на основі правил та таблиці “Результати роботи вирішувача”, взятих з бази знань, генерує висновок про необхідність та метод повторного тестування, який передається через кодувальник, динамічний довідник та діалоговий компонент користувачу;

6) динамічний довідник – надає користувачу довідку про відомі системі методи і операції основного тестування ПЗ, типи виявлених під час основного тестування помилки ПЗ, про формат вхідного файлу, а також представляє всі повідомлення будь-якого з компонентів системи;

7) діалоговий компонент – візуалізує повідомлення динамічного довідника та видає їх користувачу в зрозумілій для користувача формі.

Запропонована система повторного тестування прикладного програмного забезпечення дозволяє користувачу, подаючи в систему звіт про результати основного тестування, вирішити задачу прийняття рішення про необхідність повторного тестування, тобто про наявність прихованих помилок, та про метод, яким має здійснюватися повторне тестування.

В даній системі кодувальник є інтерфейсом між користувачем і системою, а динамічний довідник є інтерфейсом між користувачем та кодувальником.

**Спосіб представлення інформації про основне тестування в кількісному вигляді.** Вхідні дані для реалізації повторного тестування подаються у вигляді матриці

$$VD = \begin{pmatrix} m_1 & o_1 & p_1 \\ \cdot & \cdot & \cdot \\ m_i & o_i & p_i \\ \cdot & \cdot & \cdot \\ m_n & o_n & p_n \end{pmatrix},$$

де  $m_i, o_i, p_i$  – елементи множин  $M, O, P$  відповідно.

Використовуючи такі матриці:

$$MN = \begin{pmatrix} 1 & m_1 \\ \cdot & \cdot \\ i & m_i \\ \cdot & \cdot \\ s & m_s \end{pmatrix},$$

де  $m_i$  – елемент множини  $M$ ;

$$ON = \begin{pmatrix} 1 & o_1 \\ \cdot & \cdot \\ i & o_i \\ \cdot & \cdot \\ v & o_v \end{pmatrix},$$

де  $o_i$  – елемент множини  $O$ ;

$$PN = \begin{pmatrix} 1 & p_1 \\ \cdot & \cdot \\ i & p_i \\ \cdot & \cdot \\ z & p_z \end{pmatrix},$$

де  $p_i$  – елемент множини  $P$ , які представляють собою відповідність номерів методам тестування, операціям тестування та типам виявлених помилок відповідно,  $[i, 1]$ -й елемент матриці  $VD$ , представлений у вигляді тексту, піддається перетворенню для представлення його у кількісній формі.

Відбувається пошук елемента в другому стовпці матриці  $MN$ , одержується порядковий номер  $j$  рядка елемента,  $[j, 1]$ -й елемент матриці  $MN$  заноситься в  $[i, 1]$ -й елемент матриці

$$VDM = \begin{pmatrix} mn_1 & on_1 & pn_1 \\ \cdot & \cdot & \cdot \\ mn_i & on_i & pn_i \\ \cdot & \cdot & \cdot \\ mn_n & on_n & pn_n \end{pmatrix},$$

де  $mn_i, on_i, pn_i$  – кількісне представлення значень елементів множин  $M, O, P$  відповідно.

Далі піддається перетворенню в кількісне представлення  $[i, 2]$ -й елемент матриці  $VD$ . Відбувається пошук елемента в другому стовпці матриці  $ON$ , одержується порядковий номер  $j$  рядка елемента.  $[j, 1]$ -й елемент матриці  $ON$  заноситься в  $[i, 2]$ -й елемент матриці  $VDM$ .

Останнім піддається перетворенню в кількісне представлення  $[i, 3]$ -й елемент матриці  $VD$ . Відбувається пошук елемента в другому стовпці матриці  $PN$ , одержується порядковий номер  $j$  рядка елемента.  $[j, 1]$ -й елемент матриці  $PN$  заноситься в  $[i, 3]$ -й елемент матриці  $VDM$ .

**Спосіб формування вхідних векторів для ШНМ.** Після одержання кількісного представлення значень кожного елемента матриці  $VD$  формується набір вхідних векторів для ШНМ в бінарному вигляді. Для формування вхідного вектора ШНМ на основі  $i$ -го рядка матриці  $VDM$  «1» подаються на входи  $q_{mn_i}$  ( $mn_i$  –  $[i, 1]$ -й елемент матриці  $VDM$ ),  $x'_{on_i}$  ( $on_i$  –  $[i, 2]$ -й елемент матриці  $VDM$ ),  $x_{pn_i}$  ( $pn_i$  –  $[i, 3]$ -й елемент матриці  $VDM$ ), тобто номери входів  $q$ , на які подаються „1”, відповідають номерам методів тестування, які були застосовані при основному тестуванні, номери входів  $x'$  – номерам операцій тестування, номери входів  $x$  – номерам типів виявлених під час основного тестування помилок. На всі інші входи ШНМ подаються «0».

Метод вирішення задачі повторного тестування (метод функціонування ШНМ) був описаний в роботах [6 – 8].

**Спосіб лінгвістичного представлення результатів роботи вирішувача.** Результатом роботи вирішувача є матриця вихідних векторів

$$VV = \begin{pmatrix} rk_{11} & rk_{12} & rk_{13} & rk_{14} \\ \cdot & \cdot & \cdot & \cdot \\ rk_{i1} & rk_{i2} & rk_{i3} & rk_{i4} \\ \cdot & \cdot & \cdot & \cdot \\ rk_{n1} & rk_{n2} & rk_{n3} & rk_{n4} \end{pmatrix},$$

де  $i$ -й рядок містить  $i$ -й вихідний вектор; елемент  $rk_{i1}$  містить значення «нуль» або «одиниця» для рівня категорійності з номером 1  $i$ -го вихідного вектора; елемент  $rk_{i2}$  містить значення «нуль» або «одиниця» для рівня категорійності з номером 2  $i$ -го вихідного вектора; елемент  $rk_{i3}$  містить значення «нуль» або «одиниця» для рівня категорійності з номером 3  $i$ -го вихідного вектора;  $rk_{i4}$  містить значення «нуль» або «одиниця» для рівня категорійності з номером 4  $i$ -го вихідного вектора.

Вихідні вектори потрібно піддати перетворенню для одержання результатів у лінгвістичній формі. Для цього використовується матриця присвоєння рівнів категорійності типам прихованих помилок

$$RK = \begin{pmatrix} 1 & rk_1 \\ 2 & rk_2 \\ 3 & rk_3 \\ 4 & rk_4 \end{pmatrix},$$

де  $rk_i$  – тип прихованих помилок.

Перетворенню з кількісної в лінгвістичну форму піддається окремо кожен вихідний вектор, тобто окремо кожен рядок матриці  $VV$ . Для перетворення  $i$ -го рядка в ньому відбувається пошук «одиниці», запам'ятовується номер стовпця  $h$  та знаходиться  $[h,2]$ -й елемент матриці  $RK$ . Знайдений елемент  $rk_h$  є лінгвістичним представленням одержаного результату. Цей елемент заноситься в множину результатів  $R = \{rk_k | k = 1..n\}$ .

**Спосіб формування бази знань.** База знань системи повторного тестування складається з двох частин: розділ даних і розділ правил.

*Розділ даних* містить:

1) таблицю присвоєння номерів методам основного тестування (матриця  $MN$ ) (табл. 1);

Таблиця 1

Присвоєння номерів методам основного тестування прикладного ПЗ

Номер	Метод тестування ПЗ
1	Функційне тестування
2	Тестування елементів
3	Тестування незалежних шляхів (гілок)
4	Низхідне тестування
5	Висхідне тестування
6	Тестування спрягань між елементами
7	Тестування правильності

2) таблицю присвоєння номерів операціям основного тестування (матриця  $ON$ ) (табл. 2);

Таблиця 2

Присвоєння номерів операціям основного тестування прикладного ПЗ

Номер	Операція тестування ПЗ
1	Перевірка, чи виконує програмна система (ПС) очікувані функції
2	Перевірка, чи виконує ПС поставлені вимоги
3	Перевірка, чи виконують модулі ПС очікувані функції
4	...

3) таблицю присвоєння номерів типам виявлених під час основного тестування помилок (матриця  $PN$ ) (табл. 3);

Таблиця 3

Присвоєння номерів типам виявлених помилок прикладного ПЗ

Номер	Тип виявленої помилки
1	Помилки логічних умов
2	Помилки незалежних маршрутів програми
3	Помилки у гілках true, false для всіх логічних рішень
4	...

4) таблицю присвоєння рівнів категорійності прихованим помилкам (матриця  $RK$ ) (табл. 4);

Таблиця 4

Присвоєння рівнів категорійності типам прихованих помилок

Рівень категорійності	Тип прихованих помилок
1	Незначні (Н)
2	Помірні (П)
3	Серйозні (С)
4	Катастрофічні (К)

5) таблицю кількісного представлення вхідних даних (матриця  $VDM$ ) (табл. 5);

Таблиця 5

Кількісне представлення вхідних даних

Метод тестування	Операція тестування	Тип виявленої помилки
№ методу тестування 1-го рядка матриці $VD$ вхідних даних	№ операції тестування 1-го рядка матриці $VD$ вхідних даних	№ типу виявленої помилки 1-го рядка матриці $VD$ вхідних даних
...	...	...
№ методу тестування $i$ -го рядка матриці $VD$ вхідних даних	№ операції тестування $i$ -го рядка матриці $VD$ вхідних даних	№ типу виявленої помилки $i$ -го рядка матриці $VD$ вхідних даних
...	...	...
№ методу тестування $n$ -го рядка матриці $VD$ вхідних даних	№ операції тестування $n$ -го рядка матриці $VD$ вхідних даних	№ типу виявленої помилки $n$ -го рядка матриці $VD$ вхідних даних

6) таблицю лінгвістичного представлення результуючих векторів штучної нейтронної мережі (множина  $R$ ) (табл. 6);

Таблиця 6

Лінгвістичне представлення результуючих векторів ШНМ

Результуючі рівні категорійності
(1)
Рівень категорійності, одержаний для 1-го рядка матриці $VD$ вхідних даних
...
Рівень категорійності, одержаний для $i$ -го рядка матриці $VD$ вхідних даних
...
(1)
Рівень категорійності, одержаний для $n$ -го рядка матриці $VD$ вхідних даних

7) таблицю відповідності методу основного тестування і типів виявлених під час основного тестування помилок (табл. 7);

Таблиця 7

Відповідність методів основного тестування і типів виявлених під час основного тестування помилок

Метод основного тестування	Типи помилок, виявлених під час основного тестування
Функційне тестування	Некоректні чи відсутні функції
	Помилки інтерфейсу
	Помилки у внутрішніх структурах даних
	Помилки характеристик (необхідної ємності пам'яті та ін.)
Структурне тестування	Програма та її функціонування не відповідає специфікації вимог до ПЗ
	Помилки логічних умов
	Помилки незалежних маршрутів програми
	Помилки у гілках true, false для всіх логічних рішень
...	...

Розділ правил містить правила для формування висновку про необхідність повторного тестування та правила для формування висновку про метод повторного тестування.

Для опису правил формування висновку про необхідність повторного тестування введемо поріг  $a_i$  допустимої кількості прихованих помилок і важливості помилок різних рівнів категорійності, при перевищенні якого необхідно здійснювати повторне тестування з метою виявлення прихованих помилок цього виду. Тоді правила для формування висновку про необхідність повторного тестування мають вигляд: «якщо відношення сумарного значення помилок  $i$ -го рівня категорійності до загальної кількості виявлених під час основного тестування помилок перевищує поріг  $a_i$ , то повторне тестування здійснювати необхідно».

Для опису правил формування висновку про метод повторного тестування введемо поріг  $b_z$  допустимої кількості виявлених під час основного тестування помилок кожного типу. Тоді правила формування висновку про метод повторного тестування матимуть вигляд: «якщо кількість виявлених під час основного тестування помилок типу  $z$  більше 0, то

повторне тестування повинно проводитись методом, який виявляє помилки типу  $z$ », ця відповідність вказана в табл. 7 бази знань.

**Метод формування висновку про необхідність та метод повторного тестування.** На основі правил прийняття рішення про повторне тестування розробимо метод формування висновку про необхідність та метод повторного тестування.

На основі результатів роботи ШНМ формуємо множину  $K = \{k_i | i = 1..4\}$ , де  $k_i$  – сумарні значення помилок кожного з рівнів категорійності. На основі множини  $K$  та множини рівнів категорійності прихованих помилок  $R = \{rk_k | k = 1..n\}$  формуємо множину  $KR = \{kr_i | i = 1..4\}$  відношень кожного елемента вектора  $K$  до загальної кількості  $n$  елементів множини  $R$ , тобто  $kr_x = \frac{k_x}{n}$ . На основі третього стовпця матриці  $VDM$  формуємо множину  $TP = \{tp_k | k = 1..nq\}$ , де  $tp_k$  – кількість виявлених методом  $k$  помилок,  $nq$  – кількість методів тестування, тобто  $nq = 7$ .

Визначаємо порядок перегляду і застосування правил на основі отриманих результатів. Від обраного методу пошуку, тобто стратегії висновку, буде залежати порядок застосування і спрацьовування правил. Процедура вибору зводиться до визначення напрямку пошуку і способу його здійснення. В даному дослідженні використовується метод здійснення пошуку в ширину в прямому напрямку [9], тобто спочатку модуль проаналізує всі правила формування висновку про необхідність повторного тестування і за відомими фактами (елементи вектора  $KR$ ) відшукає заключення, яке з цих фактів слідує, і лише потім, якщо буде сформовано висновок про необхідність повторного тестування, аналізу піддаватимуться правила формування висновку про метод повторного тестування і за відомими фактами (елементи вектора  $TP$ ) відшукається висновок, який з цих фактів слідує.

Аналіз правил формування висновку про необхідність повторного тестування відбувається наступним чином. В множині правил типу “якщо–то”  $PR = \{pr_h | h = 1..m\}$  шукаємо правило для кожного з елементів множини  $KR$ . Якщо значення елемента множини задовольняє умові лівої частини правила, то це правило заноситься в множину обраних правил  $OPR = \{opr_y | y = 1..h\}$ . Критерій вибору єдиного правила для множини  $OPR$  неактуальний, тому що у всіх правил однакова права частина (результат), в якій робиться висновок про необхідність повторного тестування. Отже, якщо кількість відібраних правил  $h > 0$ , то робиться висновок про те, що повторне тестування необхідне.

Після формування висновку про необхідність повторного тестування, аналізуємо правила формування висновку про метод повторного тестування. В множині правил  $PR$  шукаємо правило для кожного з елементів множини  $TP$ . Якщо значення елемента множини задовольняє умові, викладеній в лівій частині правила, то це правило заноситься в множину обраних правил  $OP = \{op_e | e = 1..s\}$ . Критерій вибору єдиного правила для множини  $OP$  неактуальний, тому що, якщо в множину обраних правил потрапило  $kl$  правил, то повторне тестування повинно проводитись  $kl$  методами. Отже, висновок про метод тестування формується методом об'єднання правих частин правил множини  $OP$ .

## Висновки

Для підвищення ефективності тестування ПЗ в статті запропоновано методи і засоби процесу повторного тестування.

Вхідними даними для реалізації повторного тестування є інформація про методи і операції основного тестування та типи виявлених під час основного тестування помилок.

Розроблена структура системи повторного тестування програмного забезпечення, основні способи

опрацювання вхідної інформації та метод формування логічного висновку про необхідність повторного тестування та метод повторного тестування.

Наукова новизна розроблених методів та засобів полягає у введенні типів прихованих помилок та рівнів категорійності прихованих помилок, та пропозиції проведення повторного тестування програмного забезпечення.

Практична цінність розробки полягає в підвищенні ефективності тестування прикладного програмного забезпечення за рахунок використання розробленої системи для збільшення кількості виявлених помилок ПЗ під час повторного тестування.

### Література

1. Локазюк В.М., Савченко Ю.Г. Надійність, контроль, діагностика та експлуатація ПК: Посібник. – К.: Академія, 2003. – 376 с.
2. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. – М.: Лори, 2003. – 568 с.
3. Тестирование объектно-ориентированного программного обеспечения: Практическое пособие: Пер. с англ. / Джон Макгрегор, Дэвид Сайкс. – К.: ООО «ТИД «ДС», 2002. – 432 с.
4. Калберстон Р., Браун К., Кобб Г. Быстрое тестирование: Пер.с англ. – М.: Вильямс, 2002. – 384 с.
5. Соммервил И. Инженерия программного обеспечения, 6-е изд.: Пер. с англ. – М.: Вильямс, 2002. – 624 с.
6. Локазюк В.М., Пантелєєва (Говорущенко) Т.О. Категорійна модель процесу повторного тестування дефектів програмного забезпечення // Вісник Технологічного університету Поділля – Хмельницький: ТУП, 2004. – Ч. 1, т. 1. – С. 53 – 58.
7. Lokazyuk V.M., Govoruschenko T.O. Category Model of Process of Repeated Software Testing // Proceedings of the Third IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. – Sofia, Bulgaria: 2005. – P. 241 – 245.
8. Говорущенко Т.О. Оцінка ефективності виявлення прихованих помилок у програмному забезпеченні // Вісник Хмельницького національного університету – Хмельницький. – 2005. – Ч. 1, т. 2, С. 190 – 195.
9. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский – СПб.: Питер, 2001. – 384 с.

*Надійшла до редакції 1.11.2005*

**Рецензенти:** д-р техн. наук, проф. М.А. Філінюк, Вінницький національний технічний університет; д-р техн. наук, проф. В.Я. Мартиненко, Тернопільський державний технічний університет ім. І. Пулюя.