

## МОДЕЛІ І МЕТОДИ ТЕСТУВАННЯ ПРОГРАМНИХ СИСТЕМ

У роботі підсумовано основні результати досліджень з проблематики тестування програмних систем (ПС). Зазначено місце і роль тестування, наведено основні методи та моделі інженерії тестування ПС оброблення даних. Запропоновано структуру базового процесу тестування та метод його оцінювання.

### Вступ

Тестування – невід’ємна складова процесу програмної інженерії, один з методів подальшого вдосконалювання якості розроблених програмних засобів за допомогою усунення дефектів, що залишилися не виявленими іншими видами перевірок.

Метою здійснення досліджень з інженерії тестування програмного забезпечення, виконуваних в ІПС НАН України в 1992–2006 роках за темами [1 – 4], було розроблення моделей та методів тестування ПС оброблення даних та створення теоретичних і практичних концепцій інженерії тестування. Основні напрями досліджень:

- аналіз та систематизація методів тестування;
- аналіз існуючих моделей надійності ПС, розроблення алгоритмів та програм їх реалізації;
- розробка ефективних методів та моделей інженерії тестування;
- визначення структури базового процесу, що регламентує всі дії з підготовки, здійснення та оцінювання результатів тестування, розроблення методики виконання процесу тестування ПС оброблення даних;
- аналіз сучасних моделей та методів вдосконалення процесу тестування ПС та розроблення методики оцінювання процесу тестування.

Результати досліджень знайшли відображення в інженерії тестування, як на пряму програмної інженерії, опубліковані у звітах, методичних матеріалах, наукових виданнях та доповідалися на науково-практичних конференціях.

### 1. Постановка проблеми

З огляду на неможливість вичерпного тестування великих програмних систем, а також обмеження щодо часу та вартості тестування, перед розробниками завжди поставали проблеми впорядкування дій з тестування у вигляді зв’язаного процесу задля раціонального розподілу ресурсів, а також прийняття обґрунтованих рішень щодо початку та завершення тестування ПС.

Програмна інженерія, як інженерна дисципліна розробки програмних проєктів, робить головний наголос на підвищення якості та продуктивності ПС за рахунок використання нових і вдосконалених моделей і методів, регламентації та стандартизації процесів, вимірювання та оцінювання, інструментальної підтримки – тобто, всіх видів діяльності, щодо створення та експлуатації програмного продукту [5, 6].

Сучасний інженерний підхід, в основу якого покладено процесо-орієнтовану програмну інженерію, висуває ряд вимог і до тестування ПС, головними з яких є:

- зміщення часу початку тестування з завершальних стадій життєвого циклу (ЖЦ) проєкту ПС на ранні. Це призводить до відходу від традиційної каскадної моделі ЖЦ у бік ітеративних моделей та керованого тестуванням процесу розробки;
- вибір і використання методів і стратегій тестування, адекватних не лише об’єктам та рівням тестування, але і ресурсним обмеженням проєктів;
- стандартизація процесу тестування з урахуванням вимог процесо-орієнтованої програмної інженерії;
- визначення кількісних критеріїв завершення тестування.

За умов обмежених ресурсів проєктів великого значення набуває керування ризиком, зокрема ризиком зриву термінів і перевищення вартості тестування, а це стосується не лише процесів організації і управління тестуванням, але і визначення раціонального об'єму тестування і впорядкування цілей тестування з урахуванням ризику відмови ПС при експлуатації.

Отже, проблема формування інженерного підходу до тестування – *інженерії тестування*, – важлива та актуальна складова програмної інженерії.

## 2. Аналіз основних досліджень і публікацій

Упродовж свого розвитку проблематика тестування розвивалася паралельно у декількох напрямках, зокрема:

- дослідження та розроблення методів тестування та критеріїв адекватності тестування (відповідно до методів);
- визначення кількісних метрик тестування та критеріїв його завершення;
- створення програмних інструментів підтримки тестування;
- формування моделей оцінювання процесу тестування та його вдосконалення.

На початку 90-років минулого сторіччя відбулося переосмислення місця і ролі тестування у програмній інженерії. Традиційне уявлення про тестування як про відокремлений етап випробувань «майже» готової ПС з метою усунення помилок, що залишилися після кодування і відлагодження, поступилося місцем усвідомленню тестування як цілеспрямованої багатопланової діяльності на всіх стадіях (і в будь-яких моделях) ЖЦ ПС [7].

Новий погляд на тестування пізніше знайшов відображення у V-подібній моделі (V-моделі) тестування [8], яка систематизує всі види пов'язаних з ним дій. V-модель упорядковує два послідовні підпроцеси тестування: підготовку до його здійснення на всіх рівнях випробувань (ліва гілка V-моделі) і власне здійснення тестування при випробуваннях всіх рівнів (права гілка V-моделі).

Кроки здійснення тестування традиційні, інтегровані в усі відомі моделі

ЖЦ ПС, регламентовані стандартами ДСТУ [9, 10] і множиною методів для різних класів ПС.

Кроки з підготовки до тестування розподіляються за етапами розроблення ПС, що передують випробуванням. На кожному кроці підготовки здійснюється аналіз артефактів відповідного етапу розробки для означення цілей, об'єктів, сценаріїв і ресурсів тестування, адекватних рівню випробувань. Такий аналіз дає могутній побічний ефект. Це, з одного боку – поліпшення якості аналізованих напрацювань через усунення їх некоректності, двозначності, несумісності і. т.п. (що фактично є задачами верифікації ПС), а з іншого – уточнення множини функцій, вимог і сценаріїв функціонування ПС при моделюванні тестових ситуацій і оптимізації критеріїв завершення тестування (за тривалістю, вартістю, надійністю тощо).

З поширенням ітераційних моделей ЖЦ відбулося зміщення виконання тестування на ранні стадії ЖЦ, оскільки ітераційні моделі з їх періодичними випусками працездатних версій, дозволяють розпочати тестування вже за готовністю до тестування першої версії. Тестування кожної наступної версії охоплює також попередні, що забезпечує більш ретельне тестування всієї системи.

В ітераційних моделях ЖЦ стратегії тестування будуються з урахуванням можливих умов застосування ПС (зокрема, сценаріїв та частоти використання). Ідея побудови стратегій тестування, адекватних умовам застосування ПС, тобто таких, які враховують можливі сценарії та частоту використання ПС, здобула розвиток у концепції профілів ПС [11] – базовому положенні інженерії надійності. Під профілем слід розуміти повну множину альтернатив (наприклад, множину альтернативних категорій користувачів, функцій, сценаріїв роботи ПС тощо), для кожної з яких існує певна ймовірність появи. В ході аналізу проблемної області, виділення множини вирішуваних задач і декомпозиції системи на складові програмні компоненти можуть будуватися різні види профілів (профіль

користувача, профіль системних режимів, функціональний профіль, операційний профіль, виконавчий профіль тощо) [12].

З огляду на обмеженість часу тестування – завжди постає питання знаходження оптимального моменту випуску ПС. Очевидно, що оптимальний час випуску ПС суттєво залежить від ступеня досягнутого компромісу між такими чинниками як надійність, вартість та час. Для вирішення цієї проблеми запропоновано низку підходів, що базуються на різних припущеннях, цільових критеріях та умовах експлуатації ПС, зокрема [13–15].

Сучасні складні ПС складаються з багатьох розподілених програмних застосувань (компонентів, модулів), які розробляються незалежними групами в різних умовах та середовищах. Для таких систем проблема визначення оптимального моменту випуску може формулюватися як комбінаторна проблема оптимізації з відомими вартістю, надійністю, трудовитратами та іншими атрибутами компонентів системи.

Найбільш відомі підходи – максимізація надійності ПС, або мінімізація вартості (ресурсів). Така задача відома для вбудованих систем і вирішується методами динамічного програмування [16–19].

Запропоновано підходи, також, до оптимального розподілу надійності за компонентами ПС до початку тестування, з метою максимізації задоволення користувача при відомих технічних та вартісних обмеженнях [20 – 23].

В усіх розглянутих підходах не враховується серйозність наслідків відмов, а лише їхня ймовірність. Крім того, на той час дослідження у різних напрямках тестування не були об'єднані у цілісну концепцію.

Тому основні напрями досліджень були зосереджені на формуванні інженерії тестування програмних систем оброблення даних, яка об'єднує всі види діяльності з підготовки та здійснення тестування в єдиний базовий процес з визначеними методами, ресурсами та критеріями завершення.

Окремим напрямом досліджень є створення моделей оцінювання самого процесу тестування [24 – 26], в яких дії з

тестування впорядковуються за рівнями «зрілості» (досконалості). Ці моделі призначені як для здійснення аналізу стану процесу тестування, так і для визначення напрямків його вдосконалення.

### 3. Основні результати досліджень

#### 3.1. Методи інженерії тестування.

Використання адекватних методів – необхідна складова процесу тестування та запорука його ефективності. Дослідження методів тестування прикладного програмного забезпечення в системах обробки даних (ППЗ СОД) виконувалися в рамках НДР [1 – 2]. Для проведення функціонального тестування ППЗ СОД був адаптований та запропонований до застосування метод “Розбиття на категорії” [27]. Результати дослідження та практичного застосування методу викладені в [28].

Методи тестування розрізняються підходами до проектування тестів. Традиційно методи тестування розділяють на дві категорії - «чорний ящик» (без доступу до початкового коду) і «білий» ящик» (з доступом до початкового коду) [29]. Розширену класифікацію методів тестування, базовану на підходах до проектування тестів, наведено в [30] та показано на рис.1.

Кожний метод надає відповідний критерій покриття набором тестів вхідного простору. Вибір найбільш ефективних методів тестування за певних умов та на різних рівнях тестування є складною проблемою та пов'язаний з аналізом ризиків відмов ПС. Оскільки застосовність методів є однією з задач, вирішуваних при побудові базового процесу тестування, проаналізовано та систематизовано основні методи, які враховують специфіку типів ПС. Результати подальших досліджень методів тестування з метою їх систематизації та застосування в процесі тестування викладено в [31].

За відсутності на початку 90-х років у широкому практичному використанні потужних CASE-інструментів, у яких автоматизовано функції специфікації ПС, перевірки взаємної відповідності створюваних об'єктів та збору статистичної інформації, у ході досліджень було запропоновано використання ітераційного прототи-

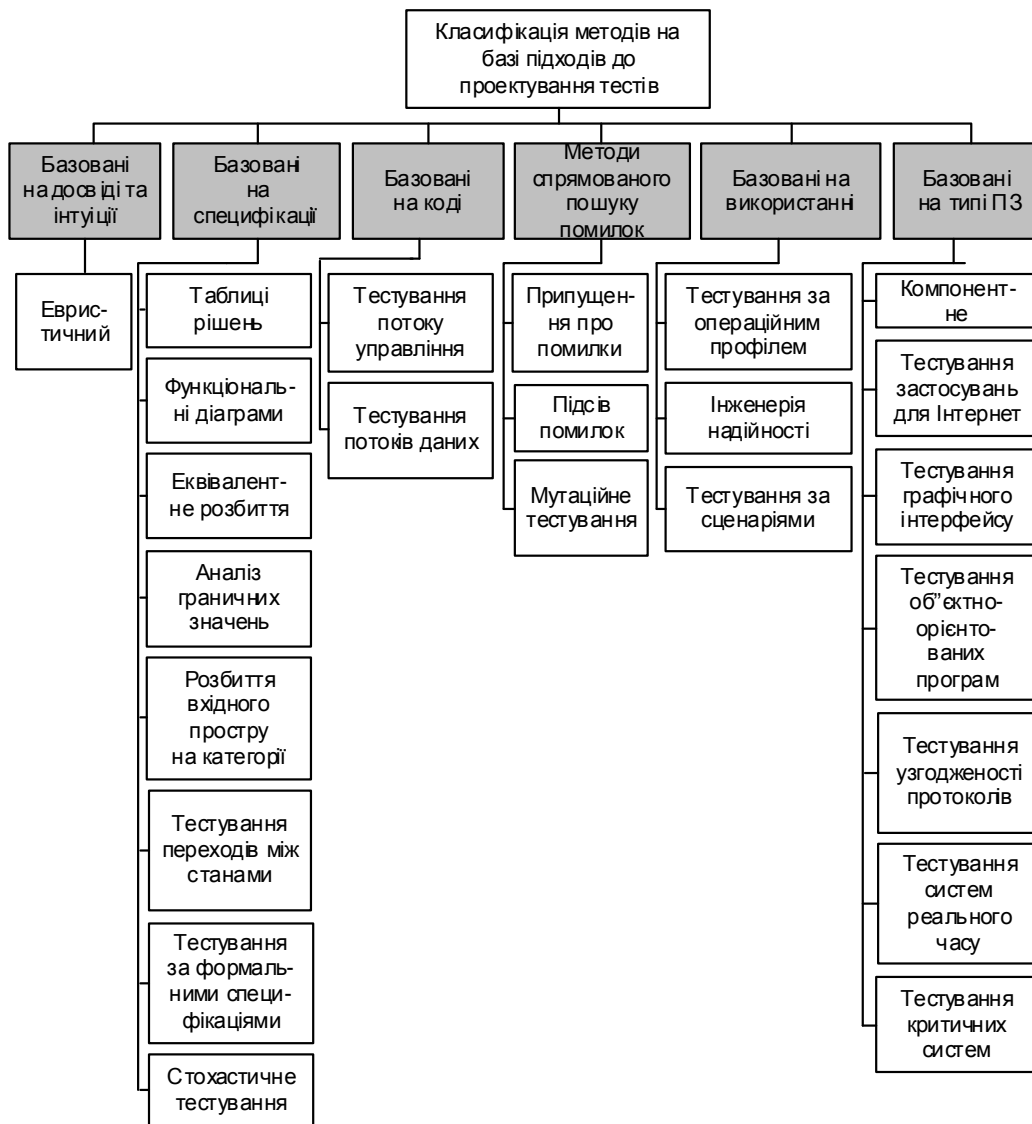


Рис. 1. Класифікація методів тестування

пування як методу специфікування, швидкого отримання перших версій діючих прототипів та їх застосування для планування тестування з урахуванням моделі використання ПС, а також для здійснення досліджень об'єму ПС, її інформаційної та програмної складності [32].

**3.2. Стратегії тестування базовані на аналізі ризику відмов.** Однією з найважливіших проблем інженерії тестування є визначення обсягів тестування та кількісних критеріїв його завершення. Аналіз існуючих критеріїв показав, що більшість з них базуються або на засадах інженерії надійності ПС, або на критеріях покриття і пов'язані з відповідними методами тестування.

У рамках досліджень проблем тестування ПС, побудованих за модульним

принципом, запропоновано підхід до вирішення задачі оптимального випуску програмних систем, базований на аналізі ризику відмов програмних компонентів і врахуванні операційного профілю функціонування системи при формуванні критерію оптимізації [33]. В основу підходу покладено розроблені метод оцінки ризику відмов програмних компонентів та модель для визначення оптимального часу тестування програмних модулів, яка враховує ризику їх відмов під час експлуатації.

Побудова операційного профілю ПС є складовою запропонованого методу оцінки ризику відмов ПС та стратегії тестування, яка полягає у визначенні програмних компонентів (ПК), які потребують більш тривалого тестування [33, 34]. На

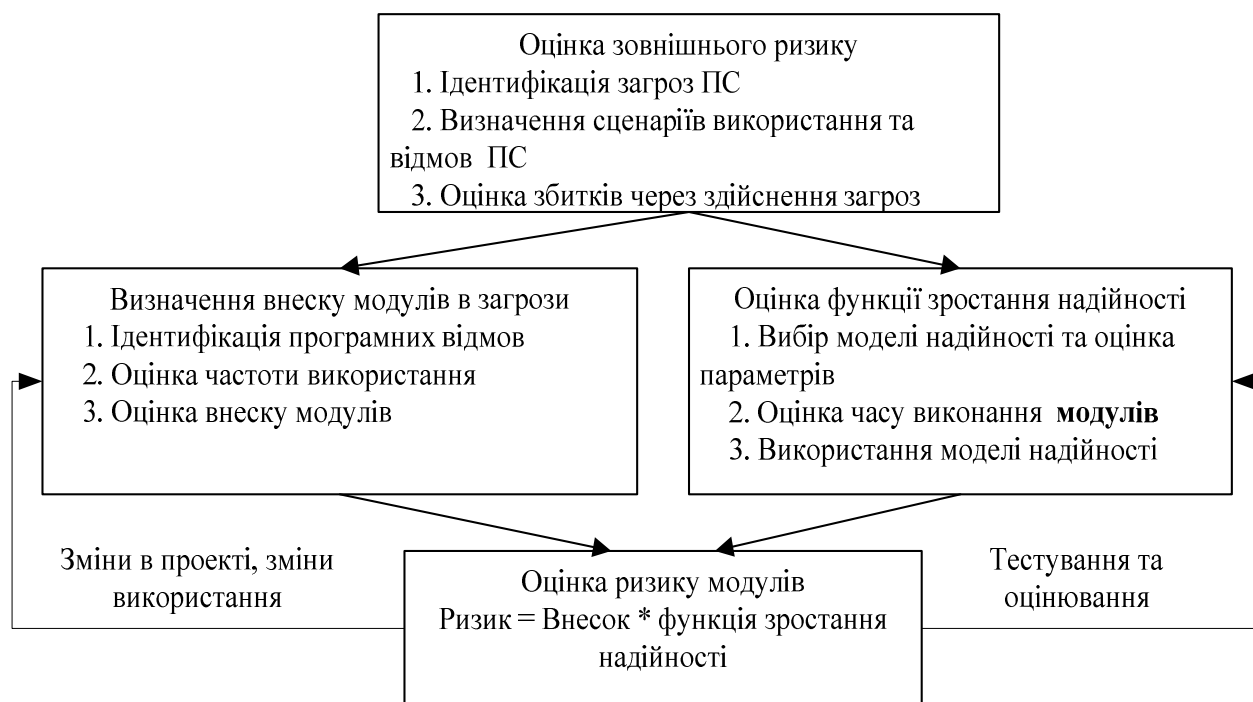


Рис. 2. Етапи методу оцінювання ризику відмов

рис. 2 показані основні етапи методу оцінки ризику відмов.

За цим методом, час тестування кожного модуля виділяється таким чином, щоб вартість тестування відповідала величині ризику використання модуля при роботі ПС.

Перевага даного підходу над відомими в інтегрованих ПС з модульною структурою полягає у тому, що він краще дозволяє мінімізувати експлуатаційні ризики відмов ПС, оскільки в значній мірі враховує як її структуру, так і майбутнє використання [35].

**3.3. Базовий процес тестування.** З появою процесо-орієнтованого підходу до розроблення ПС ідея врахування ризиків відмов ПС при визначенні стратегії тестування, а також розроблені на її підтримку моделі та методи керування процесом тестування покладені в основу базового процесу тестування [36, 37].

Ключові моменти підходу до побудови процесу:

- інтеграція з процесами ЖЦ;
- визначення оптимального часу тестування окремих компонентів;
- зниження кількості відмов при експлуатації шляхом аналізу ризику відмов.

Процес представлений множиною задач з підготовки, здійснення та оцінювання результатів тестування, які розподілені за 10 кроками процесу (рис. 3).

На кожному кроці підготовки здійснюється аналіз робочих продуктів відповідного процесу розроблення (вхідних для даного кроку процесу тестування) для визначення цілей, об'єктів, сценаріїв і ресурсів тестування, адекватних рівню тестування. Результати виконання кроків підготовки тестування фіксуються в планах тестування.

На кожному кроці виконання здійснюється фіксація результатів виконання тестів і їх порівняння з очікуваними результатами.

Результати тестування на кожному рівні аналізуються для того, щоб визначити поточний стан ПС і ухвалити рішення про достатність тестування на даному рівні.

Кожний крок процесу складається з набору вирішуваних задач. Модель визначення оптимального часу тестування використовується для визначення критерію завершення тестування ПЗ, а метод оцінювання ризику відмов є складовою задач з підготовки тестування на всіх рівнях та

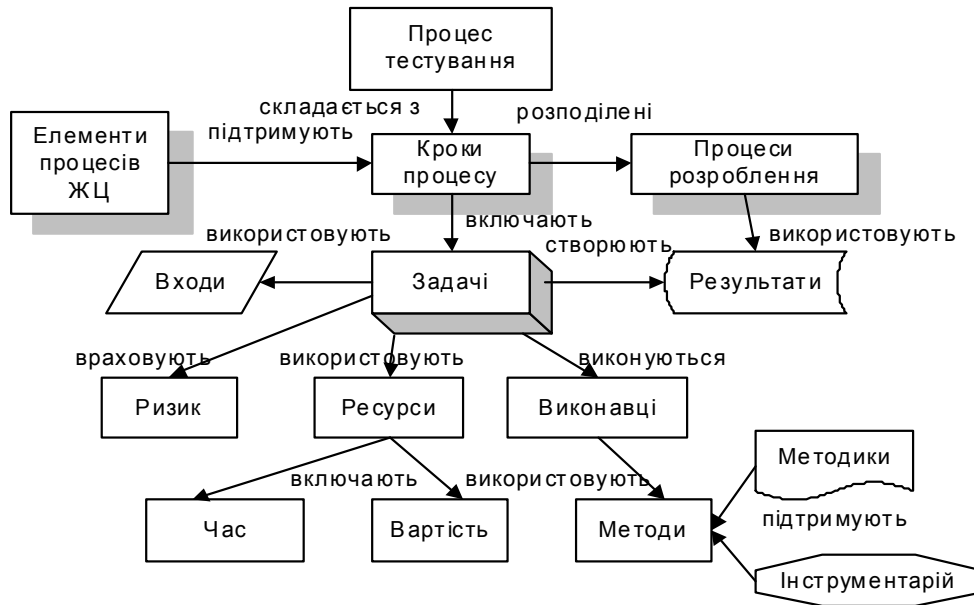


Рис. 3. Модель базового процесу тестування

оцінювання результатів тестування окремих модулів.

Базовий процес враховує: розподіл обов'язків між учасниками процесу, вимоги до професійної підготовки виконавців процесу, стандарти для представлення документів, метрики процесу, застосовні методи для вирішення задач тестування, критерії початку та завершення задач і переходу до наступного кроку процесу. Для документування процесу тестування розроблено шаблони документів, пов'язані з кожним кроком процесу.

**3.4. Моделі та методи вдосконалення процесу тестування.** Одним з напрямів досліджень інженерії тестування, виконуваних з метою визначення складу задач базового процесу тестування та шляхів його вдосконалення, є аналіз та розробка відповідних моделей оцінювання процесу тестування. В результаті досліджень застосовності існуючих моделей до умов вітчизняних проектів розроблено модель зрілості процесу тестування на базі відомої п'ятирівневої моделі ТММ (де 1 – найнижчий, 5 – найвищий) та метод експертного оцінювання процесу [37, 38].

Кожний з п'яти рівнів зрілості пов'язаний з цілями, яких має досягти організація для того, щоб процес тестування

був оцінений як такий, що відповідає даному рівню зрілості. Перехід до вищого рівня вимагає виконання всіх дій на попередніх рівнях.

Характеристики рівнів зрілості описуються в термінах цілей організації і дій з тестування. Кожний рівень (крім першого) має внутрішню структуру, яка складається з:

- набору цілей зрілості, які ідентифікують цілі вдосконалення тестування на кожному рівні;
- підцілей зрілості, які визначають контекст, межі, необхідні результати для досягнення цілей на кожному рівні;
- дій, задач і розподілу відповідальності, щодо реалізації і застосування моделі до умов оцінювання. Дії та задачі визначаються в термінах напрямків діяльності з удосконалення процесу на даному рівні, і пов'язуються з адміністративними заходами.

Метод оцінювання зрілості виконуваних у проектах дій з тестування ПС ґрунтується на представленій моделі ТММ та полягає в анкетуванні провідних фахівців проектів та згортці результатів анкетування [37]. Результати оцінювання використовуються як для визначення стану дію-

чого процесу тестування, так і для визначення напрямів його вдосконалення.

**Висновки та напрямки подальших досліджень.** Основними результатами досліджень є базовий процес тестування програмних систем та розроблені методи і моделі його виконання, а також методи оцінювання процесу. Ці результати є складовою концепцій інженерії тестування.

Отримані результати пройшли випробування при виконанні кількох проєктів створення програмних систем, виконуваних ІПС НАН України та НДІ АКС “Екотех” на замовлення державних організацій. Результати випробування та впровадження процесу тестування показали суттєве підвищення ефективності виконання процесу, зниження кількості і вартості дефектів у розроблених ІПС за рахунок:

- інтеграції процесу тестування та процесів розробки відповідно до обраної моделі ЖЦ (розподіл цілей та задач тестування по процесах та стадіях розроблення);

- упорядкування діяльності членів групи тестування (шляхом надання шаблонів форм документів та інструментальної підтримки тестування);

- використання запропонованих моделей та методів для покращення планування та керування процесом тестування.

На теперішній час дослідження проблем інженерії тестування спрямовані на розроблення нових методів та процесів тестування для програмних систем нової генерації – інтегрованих компонентів ІПС різних типів, Web-застосувань, Web-сервісів, сімейств програмних продуктів. Ці дослідження базуються на концепціях, створених у рамках проєкту фундаментальних досліджень «Розробка теоретичного фундаменту генеруючого програмування та інструментальних засобів його підтримки», який виконується у відділі програмної інженерії ІПС НАН України. В результаті виконання проєкту інтегроване середовище інженерії предметних областей планується доповнити методами інженерії тестування нових типів систем, створюва-

них у середовищі генеруючого програмування.

1. *Розробка методик, програмних засобів тестування та оцінки надійності програмного забезпечення* // НДР 6.1.3./1702-92, N UA01009618P, 1992. – 1994.
2. *Дослідження та розробка моделей, методів та засобів оцінки та забезпечення підвищених характеристик надійності розробки прикладних програм у системах обробки даних військового призначення* // НДР N 0-92-66, шифр “Вілена-УА”, 1992 – 1994.
3. *Розробка концепції, методів та нормативно-методичного забезпечення інженерії якості прикладних програмних систем* // НДР шифр “Ф1/02К-1998”, ДР 0100U006530, 1998 – 2001.
4. *Розробка концепції, методів та методичного апарату вдосконалення та стандартизації процесів життєвого циклу систем програмного забезпечення* // НДР шифр “1/02-2002”, 2002. – 2006, ДР 0102U005991.
5. *Лаврищева Е.М.* Проблематика программной инженерии // Киев: Знання, 1991. – 19 с.
6. *Лаврищева Е.М., Грищенко В.Н.* Области знаний программной инженерии SWEBOOK и подход к обучению этой дисциплине // УСиМ. – 2005. -№ 1.- С. 38 – 54.
7. *Gelperin D., Hetzel B.* The Growth of Software Testing // Comm. ACM. – 1988. – V. 31, N 6, – P. 687 – 695.
8. *Подход к тестированию и оценке надежности программного обеспечения при управлении проектом.* – Г.И Коваль, Т.М Коротун, Т.Л Яблокова и др. // Проблемы программирования. – 2000. – № 3 - 4. – С. 83 – 88.
9. *ДСТУ 2853-94 «Програмні засоби ЕОМ. Підготовка і проведення випробувань».* – К.: Держстандарт України, 2000. – 29 с.
10. *ДСТУ 2851-94 «Програмні засоби ЕОМ. Документування результатів випробувань».* – К.: Держстандарт України, 2000. – 11 с.
11. *Musa J.D., Everett W.W.* Software-Reliability Engineering: Technology for the 1900 // IEEE Software. – 1990. – № 11. – P. 36 – 43.
12. *Коваль Г.И., Мороз Г.Б., Коротун Т.М.* Концепция профилей в инженерии надежности программных систем // Ма-

- тематичні машини і системи. – 2004. – № 1. – С. 166 – 184.
13. *Musa J.D., Ackerman A.F.* Quantifying Software Validation: When to stop testing? // IEEE Software May 1989. – P. 19–27.
  14. *Kapur P., Garg R.* Optimum Software release policies for Software reliability growth model under imperfect debugging // Rech. oper. – 1990. – 24, N 3. – P. 295 – 305.
  15. *Ohtera H., Yamada S.* Optimum software-release time considering an error-detection phenomenon during operation // IEEE Trans. Reliab. – 1990. – R 39. – N 5. – P. 596 – 599.
  16. *Мороз Г.Б.* Пуассоновские модели роста надежности программного обеспечения и их приложение. Аналитический обзор // УСиМ. – 1996. – № 1 – 2. – С. 69 – 85.
  17. *Berman O., Ashrafi N.* Optimization models for reliability of modular software systems // IEEE Trans. Software Reliability. – 1993. V. 19. – P. 1119 – 1123.
  18. *Cai, M., Lyu M., Wong K, Ko R.* Component-based software engineering: Technologies, development frameworks, and quality assurance schemes // Proc. Asia-Pacific Software Engineering Conf., Dec. – 2000. – P. 372 – 379.
  19. *Fyffe D., Hines W., Lee N.* System reliability allocation and a computational algorithm // IEEE Trans. Reliability. – 1968. – V. 17. – P. 64 – 69.
  20. *Tillman F. A., Hwang C. L., Kuo W.* Determining component reliability and redundancy for optimum system reliability // IEEE Trans. Reliab. – 1977. V. 26. – P. 162 - 165.
  21. *Zahedi F. and Ashrafi N.* Software reliability allocation based on structure, utility, price, and cost // IEEE Trans. Software Eng. –Apr., 1991. V. 17. – P. 345–355.
  22. *Yacoub S. M., Cukic B., and Ammar H. H.* A component-based approach to reliability analysis of distributed systems // Proc. 18th IEEE Symp. Reliable Distributed Syst. – 1999. – P. 158 – 167.
  23. *Lyu M. R., Rangarajan S. et.* Optimal Allocation of Test Resources for Software Reliability Growth Modeling in Software Development // IEEE Trans. on Reliab. V. 51. – 2002. JUNE – N 2. – P. 183 – 192.
  24. *Koomen, T., and M. Pol.* 1998. Improvement of the test process using TPI. <http://www.iqip.nl>
  25. *Homayen A.* An Assessment Model to Determine Test Process Maturity / Diss., Illinois Institute of Technology, July 1998
  - Test Improvement Model – TPIM. [http://www.swebok.org/stoneman/version\\_0.9/](http://www.swebok.org/stoneman/version_0.9/)
  26. *Drabick R.* Growth of maturity in the testing process. International Software Testing Institute 1999. <http://www.softtest.org/articles/rdrabick3.htm>.
  27. *Richardson D., Clarke L.* A partition analysis method to increase program reliability // in proc. of yet 5th international Conf. Soft. Eng., San Diego, CA, mar.9-12, 1981. – P. 244–253.
  28. *Бернатович О.В., Коваль Г.И., Коротун Т.М.* Функціональне тестування програмного забезпечення інформаційно-расчетного типа // Проблеми програмування. – 1998. – № 3. – С. 51 – 58.
  29. *Бабенко Л.П., Лаврищева Е.М.* Основы программной инженерии. Учебник (укр. язык). – Киев: Знання, 2001. – 269 с.
  30. *Основы инженерии качества программных систем* // Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, В.Ю. Суслов / Под ред. И.В. Сергиенко. – Киев: Академперіодика, 2002. – 504 с.
  31. *Основы инженерии качества программных систем* // Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Суслов / Изд. перераб. и дополн., под ред. И.В. Сергиенко. – К.: Академперіодика, 2007. – 672 с.
  32. *Коваль Г.И., Коротун Т.М., Лаврищева Е.М.* Прототипирование как метод быстрого построения исполняемых спецификаций // Программная инженерия. – Киев. – 1993. – С. 26 – 30.
  33. *Коротун Т.М.* Тестирование информационных систем с учетом риска // Проблеми програмування. – Вып.1-2 (спец. вып. по материалам конференции УкрПРОГ'2000) – 2000. – С. 385 – 392.
  34. *Коротун Т.М.* Модель определения критерия завершения тестирования на основе анализа риска отказа и ее реализация // Проблеми програмування. – 2001. – № 1 – 2. – С. 63 – 68.
  35. *Мороз Г.Б., Коротун Т.М.* Ризико-операційний підхід до вирішення проблеми оптимального випуску програмних систем // Проблеми програмування (Спец. вип. конференції УкрПРОГ-2006). – 2006. – № 2 – 3. – С. 231 – 236.
  36. *Лаврищева Е.М., Коротун Т.М.* Построение процесса тестирования программных систем // Проблеми програмування. – 2002. – № 1 – 2. – С. 272 – 281.



37. *Коротун Т.М.* Моделі та методи інженерії тестування програмних систем в умовах обмежених ресурсів. Дис. канд. фіз.-мат. наук. Інститут кібернетики ім. В.М. Глушкова НАН України, Київ, 2005. – Держ. обл. № 0405U003083.
38. *Коротун Т.М.* Совершенствование процесса тестирования программного обеспечения // Проблемы программирования. – 1998. – Вып. 3. – С. 59 – 64.

Отримано 19.04.2007

### ***Про автора:***

*Коротун Тетяна Михайлівна,*  
кандидат фізико-математичних наук,  
старший науковий співробітник.

### ***Місце роботи автора:***

Інститут програмних систем НАН України,  
тел.:(044) 526 2735.  
mail: t.korotune@ekotex.kiev.ua