

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ОЦІНЮВАННЯ ЯКОСТІ ПРИКЛАДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Проведено аналіз сучасних проблем наукового оцінювання якості прикладного програмного забезпечення (ПЗ), результат якого дав змогу сформулювати рекомендації щодо модифікації наявних методів і засобів побудови ПЗ, технологій та моделей аналізу якості ПЗ. Розглянуто наявні методи та засоби оцінювання якості прикладного ПЗ та моделі управління його якістю. Описано методологію та обґрунтовано потребу застосування технології розроблення прикладного ПЗ через тестування (Test Driven Development – TDD) для підвищення якості та надійності його роботи. З'ясовано, що технологія TDD дає змогу: проводити модульне тестування та тестування функціональності ПЗ; виявляти помилки в процесі виконання ПЗ; проводити аналіз програмного коду щодо покриття тестами ПЗ.

Ключові слова: програмне забезпечення (ПЗ), якість прикладного ПЗ, надійність прикладного ПЗ, моделі та атрибути якості ПЗ, розроблення ПЗ через тестування, ітерації тестування ПЗ.

Вступ. Відомо [8], що на сьогодні процес розроблення прикладного програмного забезпечення (ПЗ) – найбільша сфера діяльності у світовій економіці, в якій зайнято близько 3 млн. фахівців (програмістів, розробників програмних продуктів та ін.), водночас як інші 5 млрд осіб безпосередньо залежать від їхньої успішної діяльності. За останні декілька десятиліть завдання, які потрібно виконати під час розроблення ПЗ, рівень їхньої складності та форми подання отриманих результатів кардинально змінилися. Але й донині наявні інформаційні технології виготовлення якісного ПЗ не стали повсякденною нормою.

Джерела неполадок сучасного прикладного ПЗ вкрай різноманітні, і це тільки ускладнює проблему, а також збільшує її масштаб та вартість. На думку Д. Паттерсона [14], світове гоніння за продуктивністю призвело до залежності людини від технологій, тому людству пора створювати такі інформаційні технології, на які світ дійсно може покластися, повністю довіряючи їм. Звісно, хаотичний період удосконалення ПЗ, коли значно більше уваги приділялось саме програмному коду, а не його якості, став відходити у минуле. За останні роки програмна індустрія досягла такого рівня удосконалення, при якому вимоги до забезпечення якості стали обов'язковим пунктом договорів на предмет розроблення прикладного ПЗ, оскільки саме його якість є найважливішою характеристикою з точки зору користувача – безпосереднього споживача ПЗ.

Однак, у сфері забезпечення якості ПЗ існує неприхована криза. Згідно з маркетинговими дослідженнями фірми Standish Group International [5], витрати на розроблення ПЗ становлять близько 275 млрд \$, але тільки 72 % програмних проектів сягають етапу впровадження та усього 26 % із них завершуються успіхом. Це означає, що тільки 71,5 млрд \$ витрачаються на успішні проекти, а решта 200 млрд \$ задіяні на провальні або незавершені розробки. За цією ж статистикою 18 % програмних проектів ніколи не завершуються, 53 % проектів завершуються з перевищеннями на 56 % і перевищенням термінів на 84 % і тільки 29 % проектів вкладаються у терміни та бюджет.

Проведений ще у 1993 р. [16] аналіз програмних проектів, що зазнали невдачі, дав змогу виділити 10 причин провалів, причому 7 із них повністю залежали від прийнятих рішень керівниками на початкових етапах їх реалізації:

- 1) керівники проектів не повністю розуміють вимоги замовника;
- 2) масштаби реалізації проекту визначено неправильно;
- 3) зміни до проекту впроваджуються з великими труднощами;
- 4) змінам піддають обрану технологію проектування;
- 5) замовник часто змінює вимоги до проекту;
- 6) встановлений термін виконання проекту нереальний;
- 7) замовник не приймає деяких рішень, без яких неможлива подальша реалізація проекту;
- 8) інвестиції на проект затримуються або повністю витрачено;
- 9) для реалізації проекту не вистачає виконавців;
- 10) менеджери проекту не застосовують прогресивних методів управління.

Дослідження 50 проектів, на які витрачено понад 400 людинороків і які містили майже 3000000 рядків коду, проведені у Лабораторії проектування ПЗ NASA [4], показали, що підвищена увага до раннього контролю якості дає змогу істотно знизити рівень помилок, але не знижує загальних витрат на процес його розроблення. Водночас, у роботі [7] зазначено, що саме в кінці етапу проектування можна й варто виявляти та усувати до 55 % всіх помилок майбутнього ПЗ. Отже, забезпечення можливості виявлення помилок у ПЗ як на ранніх, так і пізніх етапах його життєвого циклу дали б можливість не тільки зменшити витрати на його розроблення, але й уникнути багатьох інцидентів, причини яких можна виявити на етапах формулювання вимог щодо проектування.

На сьогодні розроблено чимало методів і засобів, технологій та моделей забезпечення якості прикладного ПЗ. При цьому в основу процесу розроблення ПЗ покладено фундаментальну ідею: його проектування є формальним процесом, який можна вивчати і вдосконалювати. Хоча для цього залучаються кращі фахівці як в Україні – В.С. Харченко, Г.І. Коваль, В.Г. Тоценко, О.В. Поморова та ін.; у країнах СНД – В.В. Липаєв, Ю.Д. Корольков та ін., так і за кордоном – В.В. Boehm, А. Avizenis, Н. Trauboth та ін., але якість такого ПЗ, як і раніше, залежить від знань та досвіду роботи самих розробників.

Однак, через прагнення розробників ПЗ зменшити собівартість процесу його розроблення, формуються нові концепції щодо розроблення прикладного ПЗ шляхом застосування таких новітніх технологій як: Rapid Application Development (RAD), Extreme Programming (XP), Agile Software Development (ASD), з використанням таких методів, як: Test Driven Development (TDD), Behavior Driven Development (BDD) та Feature Driven Development (FDD). В зв'язку з цим набуває актуальності завдання підвищення якості ПЗ шляхом модифікації наявних методів, технологій та моделей якості.

Отже, теорія та практика наукового оцінювання якості прикладного ПЗ потребує кардинальних змін, позаяк якісне виконання цієї роботи сприяє запобіганню техногенним катастрофам [8, 9], появою різних надзвичайних ситуацій і навіть звичайних інцидентів, викликаних помилками роботи ПЗ.

Об'єкт дослідження – якість прикладного ПЗ на різних етапах його життєвого циклу.

Предмет дослідження – методи і засоби аналізу і оцінювання якості прикладного ПЗ на різних етапах його життєвого циклу.

Мета роботи полягає в аналізі сучасних проблем наукового оцінювання якості прикладного ПЗ, результат якого дасть змогу сформулювати рекомендації щодо модифікації наявних методів, технологій та моделей якості ПЗ.

Викладення основного матеріалу. Криза у сфері забезпечення якості ПЗ була помітною ще понад пів століття тому, коли великі проекти стали виконуватися з відставанням від графіка виконання робіт або з перевищенням кошторису витрат, розроблений програмний продукт не мав потрібних функціональних можливостей, продуктивність його була низька, якість прикладного ПЗ не влаштувала споживачів. При наявності ряду методів і засобів, залученні кращих фахівців для розроблення технологій та стандартів забезпечення якості ПЗ, його якість, як і раніше, залежала і залежить від знань та досвіду самих розробників. В табл. 1 наведено помилки прикладного ПЗ з катастрофічними наслідками [6], які були внесені на ранніх етапах життєвого циклу.

Табл. 1. Аналіз помилок прикладного ПЗ та їхніх наслідків

Подія	Причина	Наслідки
Етап формулювання вимог		
Збій у системі Нью-Йоркського банку	Недостатність пам'яті через неправильні вимоги	32 млрд дол.
У 1990 р. на AT&T відбулась 9-ти годинна аварія	Проблеми з граничними умовами у специфікації	75 млн. нездійснених дзвінків, втрата 60 млн. дол.
У 1998 р. на AT&T відбулась 26-ти годинна аварія	Проблеми з прихованими граничними умовами у специфікації	Непрацездатність служб, пов'язаних з передачею даних
Помилка Y2K – помилка двоцифрового збереження року у даті (1999 р.)	Неправильність або неповнота специфікації	Втрати – 500 млрд дол.
Етап проектування		
У 1983 р. на станції "Серпухов-15" спрацювала система виявлення	Неправильно спроектована система розпізнавання	Світ був на межі ядерної війни
У 1991 р. система протиракетної оборони Patriot не перехопила іракську ракету	Помилка заокруглення – некоректний розрахунок місцезнаходження ракети, що наближалась	Загинули 28 американських солдат та близько 100 чоловік отримали поранення
Помилка Y2K – помилка двоцифрового збереження року у даті	Неправильний проект	Втрати 500 млрд дол.
"Смертельна" терапія у Національному онкологічному інституті у Панама-Сіті в 2001 р.	Некоректне обчислення доз радіації у ПЗ компанії Multidata Systems International	28 пацієнтів зазнали надмірного опромінення, декілька хворих померли
Аварія на заводі з перероблення урану в Західній Австралії в 2001 р.	Логічна помилка у алгоритмі	Викид радіоактивної речовини

У табл. 2 наведено розподіл помилок, які виникають на етапах формулювання вимог, проектування та реалізації прикладного ПЗ різного обсягу [4]. З цієї таблиці видно, що помилки формулювання вимог та проектування станов-

лять 25-55% всіх помилок, причому чим більший обсяг ПЗ, тим більше помилок вноситься саме на ранніх етапах. Варто врахувати також і той факт, що вартість виправлення помилки проектування в 2-4 рази вища вартості виправлення помилки конструювання.

Табл. 2. Розподіл помилок, припущених на різних етапах життєвого циклу

Етап життєвого циклу ПЗ	Обсяг ПЗ, Кбайт				
	2 К	8 К	32 К	128 К	512 К
Формулювання вимог, %	8÷10	12÷15	16÷20	18÷22	18÷23
Проектування, %	12÷15	15÷19	20÷25	22÷28	26÷32
Конструювання, %	60÷75	53÷66	44÷55	40÷50	36÷45

Отже, забезпечення можливості раннього виявлення помилок і оцінювання якості програмного проекту і прогнозування рівня якості розроблюваного за проектом прикладного ПЗ ще на етапі його проектування дали б змогу зменшити витрати на процес його розроблення, а також уникнути ряду катастроф та інцидентів, причини яких були внесені на ранніх етапах його життєвого циклу.

Враховуючи основні тенденції удосконалення інформаційних технологій на сучасному етапі становлення інформаційного суспільства, виникає потреба в підвищенні якості та надійності прикладного ПЗ. Відомі на сьогодні методології та технології контролю якості та надійності ПЗ, які знаходять широке застосування на етапі розроблення прикладного ПЗ (рис. 1), дають змогу ефективно оцінювати характеристики відповідних програмних продуктів [14, 16].

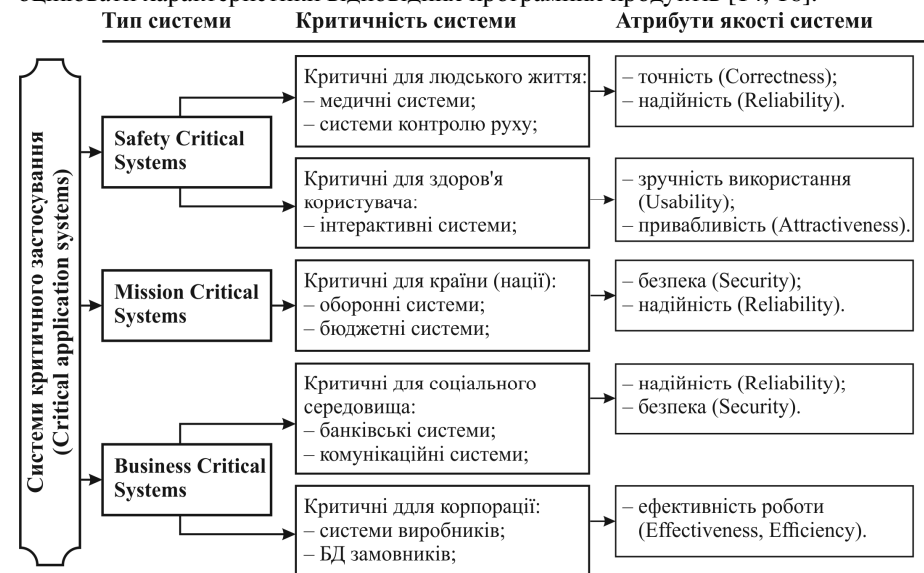


Рис. 1. Типологія програмно-апаратних комплексів критичного застосування [1]

Для формування наукових гіпотез щодо шляхів забезпечення якості прикладного ПЗ розглянемо процес його розроблення з використанням технології тестування Test Driven Development (TDD) [10] та сформуємо загальні ре-

комендації щодо підтримки, оцінювання та підвищення відповідно якості та надійності прикладного ПЗ.

В галузі якості ПЗ існує безліч стандартів – національних, галузевих, відомчих, корпоративних і т.д. Згідно з ДСТУ 2850-94 і ДСТУ 2844-94 [2], під *якістю ПЗ* розуміють сутність властивостей, що визначають ступінь його придатності для використання за призначенням, а також виділяються такі характеристики якості ПЗ: функціональність, надійність роботи, зручність використання, раціональність, можливість супроводу та мобільність.

Аналіз літературних джерел в області якості та надійності ПЗ дає різні визначення його якості відомих фахівців у галузі ІТ, серед яких набуло визнання таке консолідоване визначення: *якість прикладного ПЗ – динамічна характеристика, яка визначає відповідність кінцевого програмного продукту вимогам замовника та забезпечує відсутність дефектів у ньому.*

Огляд наукових досліджень, пов'язаних з якістю ПЗ, дає змогу зробити висновок про те, що тут основною проблемою є розроблення конструктивних підходів до побудови базової моделі якості ПЗ, яка була б придатною для різних класів і методологій його розроблення, та визнавалась одночасно розробником, замовником і користувачами. Шляхом проведення аналізу відомих моделей якості ПЗ визначено еволюцію підходів до оцінювання якості та побудови відповідної моделі якості, яку наведено на рис. 3.



Рис. 3. Еволюція підходів до оцінювання якості прикладного ПЗ [1]

На основі проведеного аналізу наявних моделей якості ПЗ у роботі [1] наведено порівняльну характеристику складових його якості за різними моделями (табл. 3).

Табл. 3. Порівняльна характеристика атрибутів якості ПЗ за різними моделями

Атрибути якості	1	2	3	4	5
Можливість тестування (Testability)	+	+		+	
Правильність (Correctness)		+			
Результативність (Efficiency)	+	+	+	+	+
Зрозумілість (Understand ability)	+			+	
Надійність (Reliability)	+	+	+	+	+
Гнучкість (Flexibility)		+	+		
Функціональність (Functionality)			+	+	+
Інженерна психологія (Human Engineering)	+				
Цілісність (Integrity)		+		+	
Сумісність (Interoperability)		+		+	
Завершеність (Maturity)					+
Ремонтопридатність (Maintainability)	+	+	+	+	+
Можливість модифікування (Changeability)	+				
Можливість перенесення (Portability)	+	+		+	+
Повторне використання (Reusability)		+			+
Зручність використання (Usability)		+	+	+	+

Примітка: 1 – модель Босма; 2 – модель Маккола; 3 – модель FURPS; 4 – стандарт ISO 9126; 5 – модель Дромі

Міжнародним стандартом для визначення якості ПЗ є ISO 9126:2001 [13]. Він складається з таких частин під загальним заголовком "Інформаційна технологія, характеристики та метрики якості програмного забезпечення": Частина 1. "Характеристики та субхарактеристики якості"; Частина 2. "Зовнішні метрики якості"; Частина 3. "Внутрішні метрики якості"; Частина 4. "Метрики якості у використанні". Модель внутрішніх і зовнішніх характеристик якості ПЗ відповідно до ISO 9126:2001 складається із шести груп базових показників, кожна з яких деталізована набором нормативних субхарактеристик. Додатково кожна характеристика супроводжується субхарактеристикою узгодженості, що має відображати відсутність суперечності з іншими стандартами і нормативними документами, а також з іншими показниками в даному стандарті.

З точки зору процесів вимірювання якості, то ця характеристика ПЗ поділяється на внутрішню якість, яка вимірюється за статистичними властивостями коду; зовнішню якість, яка вимірюється за динамічними властивостями коду в процесі виконання; якість використання ПЗ, яка вимірюється за показником, якому вона відповідає за вимогами користувача в робочому середовищі. Взаємозв'язок між різними поданнями якості прикладного ПЗ наведено на рис. 4.

Характеристики, субхарактеристики та атрибути якості прикладного ПЗ із позиції можливості та точності їх вимірювання поділяються на три типи:

- *категорійний* – описовий, що відображає набір властивостей і загальні характеристики об'єкта – його функції, які можуть подаватися номінальною шкалою категорій-властивостей;
- *кількісний* – подається множиною впорядкованих числових точок, що відображають неперервні закономірності функціонування програми та описуються інтервальною або відносною шкалою, які можна об'єктивно виміряти та чисельно зіставити з вимогами;
- *якісний* – складається з декількох впорядкованих або окремих значень – категорій, які характеризуються порядковою або точковою шкалою набору встановлених категорій, вибираються та оцінюються значною мірою суб'єктивно та експертно.

Технологія програмування TDD є однією з основних практик екстремального програмування, при якій модульні тести для всієї програми або її фрагменту пишуться ще до процесу написання програмного коду, та, як наслідок, управляють процесом його розроблення. Розроблення ПЗ в стилі TDD складається з коротких циклів (ітерацій), які наведено на рис. 6. Базові характеристики Test Driven ітерації починаються зі створення набору тестів і завершується успішним виконанням тестів.

Виникнення помилки при компілюванні ПЗ та помилкове виконання набору тестів не є обов'язковим, що, водночас, підтверджує ймовірність успішного виконання TDD ітерації. На рис. 7,а та 7,б продемонстровано можливі варіанти проходження ітерацій за технологією Test Driven Development.

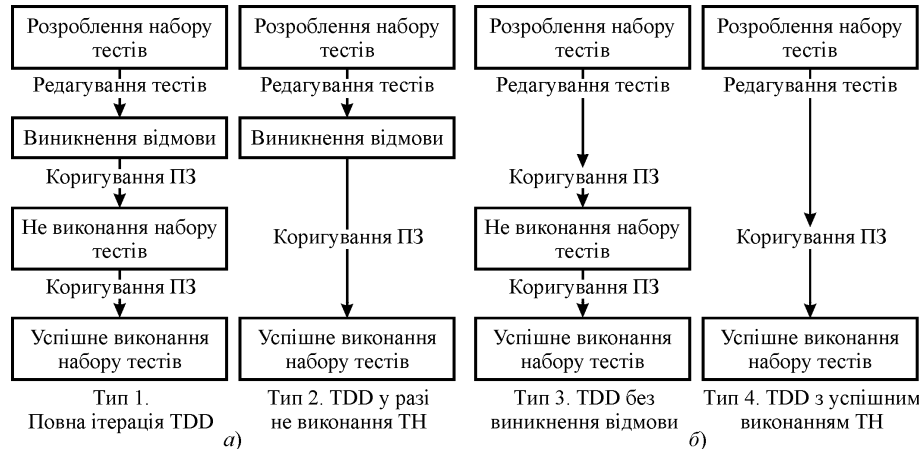


Рис. 7. Можливі варіанти виконання ітерації тестування за технологією TDD

Ця технологія має значні переваги над іншими технологіями (такими як BDD чи FDD), позаяк значно підвищує якість прикладного ПЗ, а саме:

- тести запобігають появі помилок у новому коді, що безпосередньо підвищує якість ПЗ на всьому циклі його розроблення;
- тести дають змогу проводити рефакторинг коду, що дає змогу поліпшити його структуру та повністю замінює такі методики підвищення якості ПЗ, такі як статистичний аналіз програмного коду та архітектури, формальну верифікацію та ін. Окрім цього, рефакторинг коду замінює процес верифікації прикладного ПЗ з використанням таких методик як: наскрізний контроль, колегіальні перевірки та інспекції, що значно підвищує його якість і зменшує часові та виробничі витрати на його розроблення;
- тести можуть використовуватися як документація, що взагалі вилучає потребу цільового вивчення документації для пошуку проблемних місць та перевірки ПЗ відповідності стандартам, що значно підвищує його якість;
- тести сприяють підвищенню кваліфікації розробників (PSP, Personal Software process та TSP, Team Software process) та пришвидшують процес розроблення.

Отже, згідно з вимогами сучасного замовника, та з врахуванням новітніх концепцій розроблення ПЗ проаналізовано проблему підвищення якості та надійності прикладного ПЗ. При цьому за основу використано технологію розроб-

лення TDD, яка дає змогу формулювати рекомендації щодо модифікації наявних методів, технологій та моделей якості ПЗ. Проведений аналіз моделей та підходів до оцінювання якості прикладного ПЗ визначає:

- потребу вдосконалення процесу управління якістю прикладного ПЗ на всіх етапах його ЖЦ та потребу перегляду призначення процесів інженерії якості для новітніх методологій розроблення ПЗ;
- потребу створення конструктивних концепцій побудови базової моделі якості прикладного ПЗ, яка була б придатною для різних класів і сучасних технологій його розроблення.

Висновки та перспектива виконання подальших досліджень

1. З результатів аналізу сучасного стану проблеми наукового оцінювання якості прикладного ПЗ, а також наявних методів і засобів його оцінювання виникає потреба проведення наукових досліджень з проблеми оцінювання та прогнозування якості ПЗ на усіх етапах його життєвого циклу.

2. Актуальність проблеми підвищення якості прикладного ПЗ визначає потребу розроблення фундаментальної теорії та методології системного аналізу його якості та надійності, оцінювання та забезпечення (гарантування) якості, а також потребу розроблення моделей якості ПЗ, які б враховували вплив різних чинників на процес управління якістю ПЗ.

3. Перспективою виконання подальших досліджень є забезпечення повноти та цілісності показників і характеристик якості ПЗ, для чого потрібно:

- визначити функції, які за кількісними оцінками показників якості прикладного ПЗ надаватимуть коректні кількісні оцінки характеристик якості;
- визначити функцію, яка за кількісними оцінками характеристик якості прикладного ПЗ, враховуючи їхні взаємовпливи, надаватиме коректну та достовірну кількісну оцінку якості ПЗ загалом;
- розробити методологію, яка враховуватиме взаємовпливи показників під час оцінювання характеристик якості та взаємовпливи характеристик за комплексного оцінювання якості ПЗ.

Література

1. Волкова С.О. Дослідження існуючих підходів підвищення якості програмного забезпечення критичного застосування / С.О. Волкова, О.М. Трунов // Радіоелектронні та комп'ютерні програми : наук.-техн. журнал. – Харків : Вид-во НАУ ім. М.Є. Жуковського "Харківський авіаційний інститут". – 2008. – № 6. – С. 202-208.
2. ДСТУ 2844-94. Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення. Введ. 1.08.1995. – К.: Вид-во Держстандарт України, 1995. – 57 с.
3. Канер С. Тестирование программного обеспечения / Сэм Канер, Джек Фолк, Енг Кек Нгуен. – К. : Изд-во "ДиаСофт", 2000. – 554 с.
4. Макконнелл С. Совершенный код. Мастер-класс / С. Макконнелл. – М. : Изд-во "Русская редакция", 2013. – 896 с.
5. Мищенко В.О. CASE-оценка критических программных систем. – В 3-ох т. – Т. 1: Качество / В.О. Мищенко, О.В. Поморова, Т.А. Говорущенко; под ред. В.С. Харченко. – Харьков : Изд-во НАУ им. Н.Е. Жуковского "ХАИ", 2012. – 201 с.
6. Поморова О.В. Аналіз методів та засобів оцінки якості програмних систем / О.В. Поморова, Т.О. Говорущенко // Радіоелектронні та комп'ютерні системи : наук.-техн. журнал. – Харків : Вид-во НАУ ім. М.Є. Жуковського "Харківський авіаційний інститут". – 2009. – № 6 (40). – С. 148-158.
7. Поморова О.В. Сучасні проблеми оцінювання якості програмного забезпечення / О.В. Поморова, Т.О. Говорущенко // Радіоелектронні та комп'ютерні програми : наук.-техн. журнал. –

Харків : Вид-во НАУ ім. М.С. Жуковського "Харківський авіаційний інститут". – 2013. – № 5. – С. 319-327.

8. Скляр В.В. Оценка качества и экспертиза программного обеспечения : лекционный материал / В.В. Скляр. – Харьков : Изд-во НАУ им. Н.Е. Жуковского "ХАИ", 2008. – 204 с.

9. 20 Famous Software Disasters. [Electronic resource]. – Mode of access <http://sandipsandilya.wordpress.com/2011/01/17/20-famous-software-disasters/>. – 18.11.2012 г.

10. Murphy Craig. Improving Application Quality Using Test-Driven Development / Craig Murphy. [Electronic resource]. – Mode of access <http://www.Methodsandtools.co.in/archive/archive.php?id=20>.

11. Dubois, H.F.W. Harmonization of the European vaccination policy and the role TQM and re-engineering could play / H.F.W. Dubois // Quality Management in Health Care. – 2002. – Vol. 10(2). – Pp. 47-57.

12. Introduction to Behavior Driven Development. [Electronic resource]. – Mode of access <http://behaviour-driven.org/>.

13. ISO/IEC 9126-2001 Software engineering – Product quality (Part 1 – 4) : Quality model, 2001. – 26 p.

14. Patterson, D. Recovery oriented computing: a new research agenda for a new century / D. Patterson // Proceedings of Eighth International Symposium on High-Performance Computer Architecture, 2002. – Pp. 125-127.

15. Rae A.K. Software Evaluation for Certification: Principles, Practice and Legal Liability / A.K. Rae, H.L. Hausen, and P. Robert (Editors) // McGraw Hill, International Software Quality Assurance Series, 1995. – Pp. 21-27.

16. Reiter G.J. Software management, IEEE Computer Society Press / G.J. Reiter. – Los Alamos, 1993. – 236 p.

17. Software goes wrong, we all know that, but just how wrong can it go?. [Electronic resource]. – Mode of access <http://www.datareservoir.co.uk/bugs/>. – Названіє с екрана на 18.11.2012 г.

Грыцюк Ю.И., Грыцюк П.Ю. Современные проблемы научной оценки качества прикладного программного обеспечения

Проведен анализ современных проблем оценивания качества прикладного программного обеспечения (ПО), результат которого позволил сформулировать рекомендации по модификации имеющихся методов и средств построения ПО, технологий и моделей анализа качества ПО. Рассмотрены существующие методы и средства оценивания качества прикладного ПО и модели управления его качеством. Описана методология и обоснована потребность применения технологии разработки прикладного ПО путем его тестирования (Test Driven Development – TDD) для повышения качества и надежности работы. Выяснено, что технология TDD позволяет: проводить модульное тестирование и тестирование функциональности ПО; выявлять ошибки в процессе выполнения ПО; проводить анализ программного кода по покрытию тестами ПО.

Ключевые слова: программное обеспечение (ПО), качество прикладного ПО, надежность прикладного ПО, модели и атрибуты качества ПО, разработка ПО через тестирование, итерации тестирования ПО.

Gryciuk Yu.I., Grytsyuk P.Yu. Contemporary problems of scientific evaluation of the application software quality

The analysis of the modern problems of scientific evaluation of the application software quality has been conducted in this research. The analysis result allows to provide to formulate the recommendations regarding modification of existing methods and tools for building software, technologies and models of quality analysis of software. The implicit methods and tools for evaluating the quality of the application software and models of the quality management software have been considered. The methodology has been described and the need for the introducing application of technology development of application software through its testing has been proved (Test Driven Development – TDD). TDD testing technology significantly improves the quality and reliability of software. It was found that TDD technology enables: to conduct unit testing software and test software functionality; to detect errors in the implementation of software; to conduct code analysis regarding software tests coverage.

Keywords: software, quality of the application software, reliability of the application software, model and attributes of software quality, software development through testing, iteration of software testing.

ДО ВІДОМА АВТОРІВ СТАТЕЙ

Під час підготовки статей до збірника науково-технічних праць "Науковий вісник НЛТУ України" радимо авторам дотримуватись таких рекомендацій.

Вимоги до оформлення. Обсяг тексту статті – 8-16 сторінок. Мова публікації – українська, російська чи англійська. Формат паперу – А4, поля документа – 2 см периметром. Електронний варіант потрібно створювати за допомогою текстового редактора MS Word 2003, або використовувати редактор Word молодших версій, але документ зберігати у форматі *.doc. Шрифт – Times New Roman, розмір – 14 points, рядки – через 1.5 інтервали.

Вимоги до структури статті. На початку статті обов'язково проставляють індекс УДК (Універсальної десятикової класифікації), в заголовку українською мовою зазначають: вчене звання, ініціали і прізвище автора (або авторів), науковий ступінь, назва закладу, в якому виконано роботу, назва статті, анотація та ключові слова. Далі – російською та англійською мовами: ініціали і прізвище автора (або авторів), назва статті, анотація та ключові слова. (кожна анотація має бути не меншою ніж 500 знаків).

Автор поданої до друку статті повинен чітко уявити коло читачів, на яке він розраховує. Рекомендуємо дотримуватись деяких загальних правил побудови науково-технічної статті: чітко і зрозуміло сформулювати постановку задачі; доступно викласти методику її розв'язання; зробити висновки – науковцям або дати практичні рекомендації – виробникам. Наукова праця повинна містити необхідні характеристики описаних конструкцій чи схем, але в ній не має бути ні зайвого опису історії питання, ні відомих з підручників ілюстрацій, даних, математичних викладок.

У процесі підготовки рукопису необхідно користуватися науково-технічними термінами відповідно до чинних стандартів на термінологію, наведений матеріал не повинен дублювати таблиці. Скорочення слів, імен, назв у тексті статті не допускаються. Можливе використання тільки загальноприйнятих скорочень – мір (тільки після цифр), хімічних, фізичних і математичних величин. Назви установ, підприємств, марки механізмів і т.ін., що згадуються в тексті статті вперше, необхідно писати повністю (вказуючи в дужках скорочену назву); надалі цю назву можна наводити у скороченому вигляді.

У таблицях необхідно точно вказувати одиниці фізичних величин, у назвах граф слова скорочувати небажано. Таблиці потрібно виконувати переважно вздовж листа з максимальною насиченістю інформації в рядках. Надто громіздких таблиць складати не рекомендується.

Фотографії та рисунки до статті дозволяється подати у окремому файлі у форматі *.cdr, *.tif або *.jpg, 300 dpi, b/w або Grayscale) чи оформлених у середовищі MS Excel. Зверніть увагу, що вони будуть надруковані у чорно-білому варіанті. У тексті статті посилання на ілюстрації беруть в круглі дужки, позиції на рисунках розташовують за годинниковою стрілкою і вони мають відповідати наведеним у тексті. Окремо подані ілюстрації потрібно на зворотному боці пронумерувати і підписати олівцем.