

Визначення можливості застосування концепції повторного тестування для підвищення достовірності діагностування програмного забезпечення користувацьких інтерфейсів

Тетяна Говорущенко

Кафедра системного програмування, Хмельницький національний університет, УКРАЇНА, м.Хмельницький, вул.Інститутська, 11, E-mail: tat_yana@ukr.net

Abstract – In work dualism of term "user interface", what influences on user interface testing, was considered. User interface testing types, software retesting place in software life cycle, software retesting conception, software retesting process neuronet categorical model and software retesting process neuronet technique were described. The conclusion about possibility software retesting conception using for user interface software testing reliability increase was inferred.

Ключові слова – користувацький інтерфейс, діагностування програмного забезпечення, концепція повторного тестування.

проведення тестування на зручність використання надзвичайно важливі та взаємозв'язані. Вдалий проект інтерфейсу ще не гарантує, що продукт буде зручно використовувати, і в той же час тестування за участю користувачів в жодній мірі не замінює якісної розробки. Обидва питання складають частину процесу розробки інтерфейсу, яка називається проектування зручності використання.

Категорії зручності використання:

- 1) легкий у використанні;
- 2) легкий у вивченні;
- 3) легкий в повторному вивченні;
- 4) легко забувається;
- 5) не приносить шкоди;
- 6) легко підтримується;
- 7) легко контролюється;
- 8) легко розподіляється всередині групи;
- 9) легко впроваджується в існуючі операції.

Зручність використання КІ (usability) - показник його якості, який визначає кількість зусиль, необхідних для вивчення принципів роботи з ПС за допомогою даного інтерфейсу, її використання, підготовки вхідних даних та інтерпретації вихідних даних. Іншими словами, зручність використання визначає ступінь простоти доступу користувача до функцій системи, наданих через людинно-машинний (користувацький) інтерфейс.

Вже на початковому етапі розробки тестування на зручність використання дозволить нам зрозуміти, чи зможуть споживачі взагалі використовувати цей продукт. Для відповіді на це питання необхідно одержати підтвердження того, що створюваний додаток виконуватиме потрібні користувачам функції та забезпечить їм додаткові переваги в бізнесі.

Міжнародна організація стандартизації (ISO) дає наступне визначення [1]: "Зручність використання - це ефективність, рентабельність та задоволення, з яким користувачі можуть виконати ті чи інші задачі в заданому середовищі". Тестування на зручність використання проводиться для того, щоб оцінити якість роботи продукту і виявити, наскільки він ефективний, рентабельний та чи задоволені ним користувачі.

Тестування на зручність використання здійснюється на різних етапах розробки продукту, щоб забезпечити зворотній зв'язок з користувачами. Це допомагає вдосконалювати весь проект в цілому, скорочує кількість помилок, проводить порівняльний аналіз продуктів і версій, а також підтверджує відповідність продукту вимогам, які до нього висуваються.

I. Вступ

Термін "користувацький інтерфейс" має дуальну природу, а відтак і дуальне сприйняття.

З одного боку, користувацький інтерфейс (КІ) - частина програмної системи (ПС). Відповідно, на КІ пишуться функційні та низькорівневі вимоги, за якими потім складаються тест-вимоги та тест-плани. При цьому, як правило, вимоги визначають реакцію системи на кожне введення користувача та вигляд інформаційних повідомлень системи, які виводяться на екран, пристрій друку або інший пристрій виведення. При верифікації таких вимог мова йде про перевірку функційної повноти КІ - наскільки реалізовані функції відповідають вимогам, чи коректно виводиться інформація на екран. Тобто при такому сприйнятті КІ повинен тестуватись як звичайне програмне забезпечення - методами функційного тестування.

З іншого боку, КІ - "обличчя" системи, і від його продуманості залежить ефективність роботи користувача з системою. Фактори, які впливають на ефективність роботи, слабо підлягають формалізації у вигляді конкретних вимог до окремих елементів, однак повинні бути враховані у вигляді загальних рекомендацій та принципів побудови КІ ПС. Перевірка інтерфейсу на ефективність людинно-машинної взаємодії одержала назву перевірки зручності використання (usability verification).

II. Тестування зручності використання користувацького інтерфейсу

Зручність використання повинна бути частиною проекту і проходити тестування в процесі розробки та проектування. Зручність використання повинна мати операційне визначення (щоб її можна було виміряти) та забезпечувати тестування. Головне при тестуванні - правильно обрати методи, за допомогою яких воно проводитиметься. Проблема розробки якісного КІ та

Тестування є частиною загального процесу розробки зручності використання.

Тестування зручності використання КІ не належить до класичних методів тестування ПС.

Виділяють наступні етапи тестування зручності використання КІ [2]:

1) дослідницьке тестування - проводиться після формулювання вимог до системи та розроблення прототипу інтерфейсу; основна мета - провести високорівневе обстеження інтерфейсу і вияснити, чи дозволяє він з достатньою ефективністю розв'язувати задачі користувача;

2) оціночне тестування - проводиться після розробки низькорівневих вимог та деталізованого прототипу КІ; поглиблює дослідницьке тестування та має ту ж мету; проводяться кількісні вимірювання характеристик КІ: вимірюється кількість звернень до системи допомоги по відношенню до кількості виконаних операцій, кількість помилкових операцій, час усунення наслідків помилкових операцій і т.і.;

3) валідаційне тестування - проводиться ближче до етапу завершення розробки; проводиться аналіз відповідності КІ ПС стандартам, які регламентують питання зручності інтерфейсу ([3, 4]), проводиться загальне тестування всіх компонент КІ (програмна реалізація, система допомоги, керівництво користувача) з точки зору кінцевого користувача, а також перевіряється відсутність дефектів зручності використання КІ, виявлених на попередніх етапах;

4) порівняльне тестування - може проводитись на будь-якому етапі розробки КІ; порівнюються два чи більше варіантів реалізації КІ.

III. Функційне тестування користувачького інтерфейсу

Функційне тестування КІ складається з 5 фаз:

- 1) аналіз вимог до КІ;
- 2) розробка тест-вимог та тест-планів для перевірки КІ;
- 3) виконання тестових прикладів та збір інформації про виконання тестів;
- 4) визначення повноти покриття КІ вимогами;
- 5) складання звітів про проблеми у випадку неспівпадання поведінки системи та вимог або у випадку відсутності вимог на окремі інтерфейсні елементи.

Всі ці фази такі ж, як і у випадку тестування будь-якого іншого компонента ПС. Відмінності полягають в трактуванні деяких термінів щодо КІ та в особливостях автоматизованого збирання інформації на кожній фазі.

Тест-плани для перевірки КІ представляють собою сценарії, які описують дії користувача при роботі з системою.

При збиранні інформації про виконання тестових прикладів звичайно застосовуються технології аналізу екранних форм та їх елементів (у випадку графічного інтерфейсу) або тексту на екрані (у випадку текстового інтерфейсу), а не перевірка значень тих чи інших змінних, встановлюваних ПС.

Під повнотою покриття КІ розуміють те, що в результаті виконання всіх тестових прикладів кожен інтерфейсний елемент був використаний хоча б один раз у всіх доступних режимах.

Звіти про проблеми в КІ можуть включати в себе як описи невідповідностей вимог та реальної поведінки системи, так і описи проблем у вимогах до КІ. Основне джерело проблем в цих вимогах - їх тестоне придатність, викликана розпливчатиими формулюваннями та неконкретністю.

Функційне тестування КІ може проводитись різними методами - як вручну при безпосередній участі оператора, так і за допомогою різного інструментарія, який автоматизує виконання тестових прикладів.

IV. Оцінка практичності користувацького інтерфейсу

Практичність та загальний рівень задоволеності користувачів залежить від багатьох факторів. Практичність ПЗ - це функція можливостей, КІ, продуктивності, надійності, простоти інсталяції та інформаційної підтримки продукту. Перераховані змінні є базовими факторами. Збір користувацьких відгуків за кожним з цих напрямків важливий для причинного аналізу проблем та загальної оцінки практичності.

Оцінка практичності висуває наступні основні цілі:

- 1) прогнозування користувацької задоволеності;
- 2) аналіз підходів, пов'язаних з зовнішнім виглядом, поведінкою та користувацькою взаємодією;
- 3) виявлення та вирішення проблем;
- 4) перевірка критеріїв;
- 5) конкурентна оцінка.

Оцінки призначені для визначення рівня короткотривалої та довготривалої практичності.

Методи оцінки практичності [5]:

- 1) перегляд (review) - неформальний та евристичний;
- 2) наскрізний контроль (walk-through);
- 3) лабораторні випробування за участю типових користувачів (lab-based test);
- 4) випробування на території замовника за участю реальних замовників (in-your-house test);
- 5) експлуатаційні випробування (field test).

V. Місце повторного тестування програмного забезпечення в життєвому циклі програмного забезпечення

Розглянемо методи оцінки виконання вимог до ПЗ. Така оцінка може проводитись не лише експертами в процесі ліцензування, але й розробником (в ході тестування і верифікації) та замовником ПЗ (при прийнятті ПЗ в експлуатацію). Метою проведення оцінки ПЗ є перевірка його відповідності встановленим вимогам. Не менш важливою метою роботи експерта є здійснення реального впливу на підвищення рівня якості і надійності ПЗ. Для цього всі зауваження та рекомендації експертів повинні передаватись розробникам для оперативного

усунення виявлених недоліків. В результаті сумісної діяльності розробників і експертів можуть вноситись корективи в проект і, таким чином, знижуватись кількість невиявлених дефектів ПЗ. Застосовувані методи можуть включати проведення додаткових незалежних випробувань.

Підвищити достовірність тестування і відповідно якість ПЗ можна не тільки шляхом тестування дефектів на етапах розроблення та налагодження, а й шляхом повторного тестування з метою виявлення прихованих помилок у програмах після основного тестування. Це підтверджується тим, що, як достовірність тестування, так і якість ПЗ залежать від кількості виявлених у ньому помилок, у тому числі і прихованих.

Для проведення оцінки замовником ПЗ з метою перевірки його відповідності встановленим вимогам та підвищення рівня якості і надійності ПЗ можна використовувати повторне тестування - тестування з метою виявлення прихованих помилок, яке здійснюється після розроблення та налагодження ПЗ і є окремим технологічним процесом [6, 7].

Повторне тестування здійснюється на етапі вхідного контролю, який здійснює замовник, тобто допомагає замовнику оцінити якість тестування програмного забезпечення, яке приймається, і вказує на наявність в ньому прихованих помилок [8, 9].

VI. Концепція повторного тестування ПЗ.

Нейромережна категорійна модель процесу повторного тестування ПЗ [6-9]

Всі приховані помилки розподілимо за їх небезпечністю та ступенем впливу на ПЗ на незначні (НПП), помірні (ППП), серйозні (СПП) та катастрофічні (КПП) приховані помилки. Незначними прихованими помилками (НПП) програмного забезпечення є такі, що не впливають на дії користувача, програмний продукт з їх наявністю придатний для використання. Помірними прихованими помилками (ППП) програмного забезпечення є такі, що впливають на дії користувача. Програмний продукт з їх наявністю придатний для використання з частковою втратою функційності. Серйозними прихованими помилками (СПП) програмного забезпечення є такі, що призводять до помилкових результатів, внаслідок чого програмний продукт непридатний до використання. Катастрофічними прихованими помилками (КПП) програмного забезпечення є такі, що призводять до спотворення інформації (даних), внаслідок чого програмний продукт непридатний до використання і намагання його опрацювати призводить до відмови програмної системи. Незначним прихованим помилкам присвоєно найнижчий рівень категорійності – перший. Помірним прихованим помилкам присвоєно, відповідно, рівень 2; серйозним – рівень 3. Найвищим рівнем вважають катастрофічний – рівень 4. Таким чином, рівнів прихованих помилок буде чотири.

На основі запропонованої концепції повторного тестування і розподілу прихованих помилок за категорійністю з врахуванням порогів допустимої кількості помилок і важливості помилок [6-9] розроблено математичну модель процесу повторного тестування, в основі якої лежить штучна нейронна мережа (ШНМ) типу прямонапрявленого перцептрону.

Вибір апарату ШНМ мотивований тим, що штучні нейронні мережі за рахунок можливості апроксимації нелінійних функцій дають можливість враховувати важливість (ваги) кожного типу неприхованих та прихованих помилок, порогови граничної кількості допустимих помилок кожної категорії, взаємний вплив прихованих помилок одних типів на помилки інших типів. Визначення вихідного функціоналу кожного із шарів ШНМ, що відповідають категорійності помилок, дає можливість оцінити сумарний вплив кожної категорії помилок на якість розробленого ПЗ, яке пройшло основне тестування, і зробити висновки щодо необхідності повторного тестування з метою виявлення і усунення помилок тієї чи іншої категорії. Важкоформалізованою задачею повторного тестування є визначення ваг впливу помилок різних типів однієї категорії [6, 8] на помилки іншої категорії, причиною яких є помилки попередніх категорій [6, 8]. Ця задача може бути вирішена за допомогою використання навченої ШНМ.

Зазначений підхід відображено узагальненою складною ШНМ, в якій структура багатозарового перцептрона типу MLP (multi-layer-perceptron) поєднується зі структурою простого перцептрона Розенблатта (одношарова ШНМ, де ваги першого шару незмінні, і зважені компоненти вхідного вектора на вході нейрона другого шару сумуються). Структура цієї ШНМ представлена на рис. 1 [6, 7].

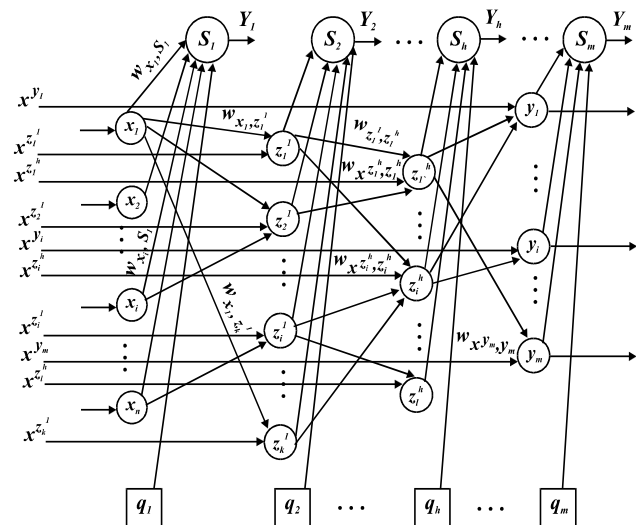


Рис.1. Нейромережна категорійна модель процесу повторного тестування

VII. Нейромережний метод процесу повторного тестування ПЗ [10]

Початковими даними для реалізації повторного тестування є інформація про типи помилок (множина $P = \{p_k | k = 1..n\}$), виявлених під час основного тестування, та методи (множина $M = \{m_k | k = 1..n\}$) і операції, що були застосовані для їх виявлення (множина $O = \{o_k | k = 1..n\}$). Ця інформація береться із звітів про результати основного тестування, які надаються тестувальником у вигляді журналу “Метод тестування – Операція, яка виконується під час тестування – Результат операції (тип виявленої помилки)”. Оскільки основне тестування здійснює тестувальник, то на результати тестування можливий вплив суб’єктивного та людського факторів (врахування «почерку» тестувальника), що може як позитивно, так і негативно впливати на ефективність повторного тестування. Саме для зменшення зазначеного суб’єктивного фактора враховуються не тільки кількість і типи виявлених помилок, а й методи та операції тестування. Принцип застосування ШНМ для процесу повторного тестування описано в [10].

Висновок про можливість застосування концепції повторного тестування ПЗ для підвищення достовірності тестування ПЗ КІ

З вищевикладеного видно, що можливо використати концепцію повторного тестування ПЗ для підвищення достовірності тестування ПЗ КІ. Потрібно лише при тестуванні ПЗ КІ складати звіти основного тестування (функційного тестування та тестування на зручність використання), які потім будуть використовуватись як вхідні дані для повторного тестування. Потрібно буде здійснити навчання ШНМ для діагностування ПЗ КІ, тобто побудувати навчальну вибірку ШНМ, яка складатиметься з методів тестування КІ, операцій тестування КІ та помилок КІ. Реалізація, навчання та дослідження запропонованої моделі в пакеті Matlab описана в [11-13].

Література

- [1] *Тео Мандел.* Разработка пользовательского интерфейса: Пер. с англ. – М.: ДМК Пресс, 2001. – 416 с.
- [2] *Говорущенко Т.О., Янчук О.М.* Підвищення достовірності діагностування програмного забезпечення користувачьких інтерфейсів // Прогресивні інформаційні технології в науці, освіті та економіці. Збірка наукових праць. - Вінниця: Вінницький кооперативний інститут, 2009. - с. 35-41
- [3] *ISO 13407:1999.* Human-centered design processes for interactive systems. - International Organization for Standardization, 01 Jun 1999. - 26 p.
- [4] *ISO/IEC 9126-1:2001.* Software engineering - Product quality - Part 1: Quality model. - International Organization for Standardization / International Electrotechnical Commission, 01 Jun 2001. - 25 p.
- [5] *Р.Дж.Торрес.* Практическое руководство по проектированию и разработке пользовательского интерфейса: пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 400 с.
- [6] *Локазюк В.М., Пантелеева (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.* Підвищення достовірності тестування програмного забезпечення // Вісник Національного університету “Львівська політехніка” “Комп’ютерні науки та інформаційні технології” – Львів: Видавництво Національного університету “Львівська політехніка”, 2007 – с.186-196
- [9] *V.Lokasyuk, O.Pomorova, T.Govorushchenko.* Neural Nets Method for Estimation of the Software Retesting Necessity // Proceedings of the 2008 international workshop on Software Engineering in east and south Europe – Germany, Leipzig, 2008. – pp. 9-14. ISBN 978-1-60558-076-0. (<http://doi.acm.org/10/1145/1370868.1370871>)
- [10] *Говорущенко Т.О.* Система повторного тестування програмного забезпечення // Радіоелектронні і комп’ютерні системи – Харків: НАУ “ХАІ”, 2005. - №4, с.120-126
- [11] *Локазюк В.М., Поморова О.В., Говорущенко Т.О.* Імітаційна модель системи повторного тестування програмного забезпечення // Вісник ХНУ – Хмельницький: ХНУ, 2006. – № 6, с.65-72
- [12] *Govoruschenko T.O.* Model of decision maker of repeated application software testing system // Радіоелектронні і комп’ютерні системи – Харків: НАУ “ХАІ”, 2007 – № 7, с.191-198
- [13] *Говорущенко Т.О.* Дослідження моделі вирішувача системи повторного тестування прикладного програмного забезпечення // Вісник Хмельницького національного університету – Хмельницький: ХНУ, 2007 - №3, т.1, с.236-244