

де x – оцінка характеристики, наприклад зручність роботи, зручність навчання і т.п .

Інформація про зміщення оцінки характеристики якості ПЗ дозволяє приймати рішення про стан якості програмного продукту та швидко реагувати на негативні зміни.

Після знаходження середньої оцінки характеристики якості ПЗ, потрібно переконатися наскільки вона є точною, тому оцінюють розсіювання оцінки значень метрик атрибутів [3]:

$$S^2 = \overline{(x - \bar{x})^2} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2. \quad (4)$$

Ефективною вважається оцінка, яка при заданому обсязі вибірки n має найменшу дисперсію.

Безпосередньо кінцевого користувача ПЗ цікавить експлуатаційна якість програмного продукту, тому окрім технічної точки зору на якість важливою є оцінка з позиції користувача, тобто оцінка користувачем зручності використання та практичності (usability). Тоді атрибутами цієї характеристики є легкість навчання, привабливість, зрозумілість для користувача. Метриками цих атрибутів будуть відповідно: час навчання, оцінка привабливості, час на виконання поставлених завдань для групи користувачів. Оцінка usability на основі запропонованого методу дозволяє реагувати на зміни в продукті з точки зору користувача та розробляти продукт для користувачів.

У роботі запропоновано метод оцінки якості програмного продукту, який дає можливість відстежувати оцінку характеристик програмного продукту на різних етапах розробки програмного забезпечення. Перевагою запропонованого методу є відображення інформації про відхилення значень характеристик та можливість надання оцінки якості продукту під час його розробки, що дозволяє швидко реагувати на зміни в якості.

Список використаних джерел

1. Software engineering – Product quality [Електронний ресурс] // International Organization for Standardization. – 2001. – Режим доступу до ресурсу: http://www.iso.org/iso/home/store/catalogue_tc
2. Software engineering – Product quality Part 4: Quality in use metrics [Електронний ресурс] // International Organization for Standardization. – 2004. – Режим доступу до ресурсу: http://www.iso.org/iso/home/store/catalogue_tc
3. Гмурман В. Е. Теория вероятностей и математическая статистика / В. Е. Гмурман. – Москва: Высшая школа, 2004. – 404 с.

УДК 004.054

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ПРОЦЕСУ ВІДСТЕЖЕННЯ ПОМИЛОК ПРИ РОЗРОБЦІ ПРОГРАМНОГО ПРОДУКТУ

Струбицька І.П.¹⁾, Хома Ю.В.²⁾

Тернопільський національний економічний університет

¹⁾ к.т.н., доцент; ²⁾ магістрант

І. Актуальність теми

На сьогодні в Україні створення програмного забезпечення стає все більш розвиненим. Для підвищення конкурентоспроможності при створенні програмних продуктів українські розробники повинні перш за все приділяти увагу якості цих продуктів. Головною задачею перевірки якості продукту є пошук можливих помилок в програмах і їх документування. У зв'язку з цим, все більш гострою стає проблема вибору програмного забезпечення, яке полегшує і пришвидшує документування помилок програмних продуктів.

Дефекти програмного забезпечення та процес відстеження помилок досліджуються багатьма вченими, наприклад Савін Р., А. Тріфу, Д. Реч, М. Мантула, О.Кіупке.

При розробці програмного забезпечення дуже важливим є наявність системи відстеження помилок (Bug tracking system), тому ці системи набули широко застосування. Використання таких систем вважається однією з «ознак хорошої команди програмістів». Основою для системи відстеження помилок є база даних, яка зберігає всю інформацію про помилки. Структура такої бази даних може бути різною,

залежно від реалізації конкретної системи чи потреб користувача. Головна перевага системи відстеження помилок полягає в забезпеченні огляду всіх внесених дефектів та їх стану.

II. Мета дослідження

Метою дослідження є створення математичного та програмного забезпечення для підтримки процесу відстеження помилок при розробці програмного продукту, яке дасть змогу покращити процес роботи користувача з базою даних про дефекти, в той же час надаючи зручний інтерфейс для роботи з нею.

III. Практичне значення одержаних результатів

Практичне значення одержаних результатів полягає в тому, що впровадження розробленого програмного забезпечення дозволить підвищити ефективність відстеження помилок і значного полегшить роботу з документуванням дефектів під час розробки програмного продукту.

IV. Розробка користувацького інтерфейсу на основі методу, орієнтованого на використання

Одним з найбільш технологічних підходів для розробки зручного користувацького інтерфейсу є метод, орієнтований на використання (usage-centered design), який запропонований Л. Константайном і Л. Локвудом (L. Constantine, L. Lockwood).

Основна ідея цього методу — використання спеціальних моделей, які сприяють адекватному визначенню набору задач, що необхідно розв'язувати користувачем, і способів організації інформації, котрі дозволяють спростити їх рішення.

В рамках цього методу використовується модель задач. Ця модель при проектуванні користувацького інтерфейсу будується на основі істотних варіантах використання (essential use cases). Опис такого варіанта використання відрізняється від звичайного тим, що в рамках його сценаріїв виділяються тільки цілі і задачі користувача, а не конкретні його дії.

Ціллю такого виділення є звільнення від неявних припущень про наявність певних елементів інтерфейсів, що допомагає розробляти їх саме для вирішуваних завдань. Зручно описувати такі сценарії у вигляді двох послідовностей — устремлень користувача (не дій, а завдань, які він хоче вирішити) і зобов'язань системи у відповідь на ці устремління.

В результаті модель задач являє собою набір перероблених варіантів використання зі зв'язками між ними по узагальненню, розширенню та використанню. Деякі з прецедентів оголошуються основними — без них програма втратить значну кількість користувачів. При цьому повинна будь-яка користувацька роль повинна бути пов'язана з одним або декількома варіантами використання.

Приклад опису істотного варіанту використання авторизації користувача приведений в таблиці 1.

Таблиця 1

Варіант використання для авторизації користувача

Задачі	Зобов'язання
Авторизація в системі	
	Перевірка особи
	Відкриття головної сторінки

Як можна побачити в описі варіанту використання в таблиці 1, під час авторизації користувача система повинна спочатку перевірити особу, а потім відобразити головну сторінку програми.

Результатом проектування буде створений інтерфейс для авторизації користувача, який зображено на рисунку 1.

User Login

Username

Password

[Forgot password?](#)

Рисунок 1 - Спроектований інтерфейс авторизації користувача

Висновки

На основі аналізу недоліків існуючих систем відстеження помилок під час розробки програмного продукту розроблено систему, яка дає змогу покращити процес роботи користувача з базою даних про дефекти, в той же час надаючи зручний інтерфейс для роботи з нею.

В ході виконання роботи створене нове програмне забезпечення для підтримки процесу відстеження помилок при розробці програмного продукту, яке виконує стандартний функціонал по відстеженню інформації про дефекти і:

- реалізує удосконалений, зручніший інтерфейс користувача, який покращено за допомогою виправлення недоліків програм-аналогів;
- забезпечує відображення статистики поточного стану дефектів проекту у вигляді графіків.

Система реалізована на основі технології .NET з використанням шаблону ASP .NET MVC 4. Перш за все створено зручний інтерфейс, на основі аналізу недоліків інтерфейсів інших систем. Для цього використані бібліотеки Java Script, такі як JQuery та інші. При цьому велика увага приділяється основному функціоналу та взаємодії з іншими програмами, наприклад імпорту готових даних з Excel.

Також створена можливість формування графічних звітів, тобто відображення графіків з станами помилок по кожному створеному проекту за допомогою бібліотеки DotNet.Highcharts.

Список використаних джерел

1. Канер С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / Канер С., Фолк Дж., Нгуен Е. – К.: Издательство «ДиаСофт», 2001. – 544 с.
2. Савин. Р. Тестирование Дот Ком, или Пособие по жесткому обращению багами в интернет-стартапах / Р. Савин. – Дело, 2007. – 312 с.
3. Соммервилл И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. / Соммервилл И. – М.: Издательский дом "Вильямс", 2002. – 624 с.

УДК 004.434

ВІЗУАЛЬНИЙ КОНСТРУКТОР ГРАФОВИХ МОДЕЛЕЙ ДЛЯ ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ

Шевчук Р.П.¹⁾, Кульчицький – Поливко Б.В.²⁾
Тернопільський національний економічний університет
¹⁾ к.т.н., доцент; ²⁾ магістрант

І. Вступ

Останні роки характеризуються інтенсивним впровадженням у сферу управління різними промисловими процесами мікроконтролерів. Поява стандарту ІЕС 1131-3 безсумнівно внесла значний вклад у розвиток систем розробки програмного забезпечення для промислових мікроконтролерів [1]. Однак системи розробки програм, що поставляються разом з конкретними мікроконтролерами їх підприємствами-виробниками часто не відповідають вимогам даного стандарту [1].