

## ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РИЗИК-ОРІЄНТОВАНОГО ОЦІНЮВАННЯ ФУНКЦІОНАЛЬНОСТІ WEB-ОРІЄНТОВАНИХ СИСТЕМ

Мельник А.М.<sup>1)</sup>, Посуляк Р.М.<sup>2)</sup>, Буца В.Р.<sup>2)</sup>

Тернопільський національний економічний університет

<sup>1)</sup> к.т.н., доцент; <sup>2)</sup> магістрант

### I. Постановка проблеми

Для сучасних web-орієнтованих інформаційних систем, що характеризуються великим обсягом даних і складною функціональністю, актуальною є задача забезпечення якості, зокрема, коректності і надійності. Досягненню необхідного рівня якості перешкоджають негативні наслідки реалізації ризиків розробки інформаційних систем (ІС), найбільш критичним з яких багато дослідників вважають ризик порушення календарного планування [1]. Для зниження їх негативного впливу необхідно враховувати їх рівень на стадії планування при оцінці календарних термінів, а також постійно контролювати прогрес розробки для визначення відповідності встановленим термінам і вимогам зацікавлених осіб на кожній наступній ітерації.

Ефективним методом, використовуваним для контролю ходу розробки і якості ІС, є регресійне тестування [2] - повторне тестування частин розроблюваної системи, які залежать від внесених до неї змін. Даний вид тестування є дуже дорогим і трудомістким, що обумовлює необхідність скорочувати обсяги тестованого матеріалу за рахунок прогнозування найбільш критичних модулів системи.

### II. Мета роботи

Метою дослідження є розробка інформаційної технології ризик-орієнтованого оцінювання функціональності web-орієнтованих систем.

Для реалізації даної інформаційної технології необхідно розробити модель оцінки функціональності системи, що дозволяє:

1) врахувати вплив невизначеності на процес розробки; 2) оцінити відповідність артефактів системи вимогам на кожному етапі життєвого циклу розробки; 3) оцінювати рівень ризиків розробки та визначати в процесі розвитку ІС найбільш критичні модулі для тестування; 3) контролювати прогрес розробки та визначати функціональні модулі, розробка яких не відповідає термінам календарного плану, використовуючи комплекс критеріїв, що дозволяють оцінювати проміжний прогрес виконання завдання.

### III. Критерій оцінки якості та прогресу виконання проекту

Введемо поняття «функціональна одиниця системи» - елементарна структурна складова ІС, яка реалізує закінчений функціональний блок, для перевірки якого може бути розроблений один або більше автоматизованих або автоматичних тестів. Функціональну одиницю необхідно вибирати залежно від виду ІС. Наприклад, якщо ІС вимагає зв'язку з базою даних або звернення до віддаленого додатку, що має інший формат даних, функціональною одиницею проекту є функція обміну інформацією з базою даних (віддаленим додатком).

На відміну від функціональної точки [2], функціональна одиниця дає можливість використовувати більш укрупнений функціональний блок, характерний для конкретного типу ІС, що дозволяє спростити процедуру планування і тестування даних елементів.

Таким чином, якщо прийняти за  $F$  - множину виділених в ІС функціональних одиниць, а за  $F_k$  - множину реалізованих і які пройшли тестування в  $k$ -ій версії, то прогрес розробки ІС в першому наближенні можна оцінити за допомогою критерію (1):

$$M_{progress1} = \frac{F_k}{F} \times 100 \quad (1)$$

Даний критерій розраховується на кожному етапі життєвого циклу проекту за допомогою автоматизованого тестування проекту, що дозволяє перевіряти відповідність кількості сторінок усіх рівнів та зв'язку між ними. Запропоновані показники дозволяють визначити обсяг виконаних завдань на

кожній ітерації, на підставі чого, у разі, якщо трудовитрати не відповідають запланованим, приймається рішення про переоцінку ризиків розробки.

#### IV. Ризик-орієнтована модель оцінки функціональності системи на основі прогнозування ризиків

Запропоновані критерії лягли в основу ризик-орієнтованої моделі оцінки функціональності системи. На рисунку 1 представлена бізнес-модель основних етапів розробки в разі вибору методології SCRUM (при виборі будь-якої іншої моделі розробки з ітераційним життєвим циклом концепція аналогічна - при плануванні кожної ітерації використовується інформація про виділені ризики та значення критеріїв прогресу виконання проекту). У даній моделі ризики оцінюються трудовитратами на планове доопрацювання і додаткове тестування у разі реалізації ризику.



Рисунок 1 – Ризик-орієнтована модель оцінки функціональності ІС

Рівень ризику зміни вимог на  $k$ -тій ітерації служить джерелом інформації для розрахунку скоригованої трудомісткості наступної ітерації і оцінюється як:

$$R_{changing\ requirements}^{(k)} = \frac{N_{cancel}}{N} \times 100, \quad (2)$$

де  $R_{changing\ requirements}^{(k)}$  - значення рівня ризику зміни вимог на поточній ітерації;  $N$  - кількість функціональних одиниць, виконання яких було заплановано на поточну ітерацію;  $N_{cancel}$  - кількість скасованих функціональних одиниць, виконання яких було заплановано на поточну ітерацію.

Таким чином, за допомогою прогнозування ризиків і запропонованих критеріїв оцінки прогресу дана модель дає можливість оцінити прогрес розробки web-системи на будь-якій ітерації в будь-який момент часу, перевірити на відповідність вимогам замовника створені на кожному етапі артефакти і переоцінити рівень ризику для наступної ітерації.

#### V. Інформаційна технологія ризик-орієнтованої оцінки функціональності web- системи

Запропонована модель лягла в основу інформаційної технології ризик-орієнтованого оцінювання функціональності web-орієнтованих систем, яка складається з наступних етапів.

Етап 1. Вибір моделі розробки ІС. Етап 2. Ризик-орієнтована оцінка довжини спринту. Етап 3. Ризик-орієнтоване тестування. Етап 4. Переоцінка ризиків розробки на підставі результатів тестування. Етап 5. Завершення розробки. На заключному етапі відбувається формальний процес тестування, який перевіряє відповідність системи вимогам і проводиться з метою визначення чи задовольняє система встановленим критеріям. Винесення рішення про прийом системи замовником або іншою уповноваженою особою.

#### Висновок

У роботі розглянута інформаційна технологія ризик-орієнтованого оцінювання функціональності web-орієнтованих систем, яка ґрунтується на ризик-орієнтованій моделі оцінки функціональності та включає використання автоматизованих підсистем оцінки ризику і тестування якості розробки web-сайтів. Використання запропонованої інформаційної технології дозволяє знизити рівень ризику розробки проекту, контролювати прогрес розробки, оцінити відповідність еволюційному прототипу і календарному плану на будь-якій ітерації, а також отримати рекомендації з управління ризиками та зміни термінів календарного планування проектних завдань.

Виходячи з цього, можна зробити висновок, що запропонована інформаційна технологія дає можливість розробити систему відповідно до запланованих термінів і висунутих вимог до проекту з найменшим рівнем ризику і заявленим рівнем якості.

### Список використаних джерел

1. Брагина Т.И. Стратегия тестирования web-проектов / Т.И. Брагина, Г.В. Табунщик // Наукові праці Донецького національного технічного університету. Серія "Інформатика, кібернетика та обчислювальна техніка" (ІКОТ-2012). Вип. 15 (203) – Донецьк: ДВНЗ "ДонНТУ". – 2012. – С. 118-124.
2. Bragina T.A Modified Method for Estimating Software Projects Labor Costs [Текст] / T. Bragina, G.Tabunshchik // TCSET'2012: Proc. Of XI Int. Conf. Modern Problems of Radio Engineering, Telecommunications and Computer Science. – Lviv: Lviv Polytechnic National University, 2012. – P. 245.

УДК 004.05

## МЕТОД ОЦІНКИ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Козак О.Л.<sup>1)</sup>, Струбицька І.П.<sup>2)</sup>, Міщанчук М.Д.<sup>3)</sup>

Тернопільський національний економічний університет

<sup>1), 2)</sup> к.т.н. доцент, <sup>3)</sup> магістрант

Якість програмного забезпечення це величина, яка відображає ступінь відповідності програмного забезпечення до вимог та здатність задовольнити потреби замовника.

Програмне забезпечення розробляють вже понад півстоліття і за цей період задачі та методи їх розв'язання кардинально змінилися. Але й сьогодні розробка високоякісного програмного забезпечення є актуальною задачею. Період коли за еталон якості брали якість програмного коду відходить у минуле. Сьогодні програмна індустрія досягла такого рівня, коли оцінка якості програмного забезпечення є обов'язковим пунктом в договорі розробки програмного продукту. Для кількісного встановлення критеріїв якості, за якими буде здійснюватися перевірка і підтвердження відповідності ПЗ заданим вимогам, визначають відповідні зовнішні вимірювані властивості ПЗ, метрики, діапазони зміни значень і моделі їх оцінки.

Актуальною є розробка методів та підходів для оцінки якості програмного забезпечення, яка враховує і інші характеристики якості ПЗ окрім функціональності. Метою роботи є розробка методу для оцінки якості програмного забезпечення, що дозволить реагувати на зміни якості продукту під час розробки та впровадження.

Для вирішення поставленої задачі розроблено підхід, який дає можливість оцінити якість програмного забезпечення з врахуванням окремих характеристик таких як: функціональність, надійність, зручність використання, ефективність, зручність супроводу, мобільність.

Для оцінки якості програмного продукту доцільно проводити моніторинг метрик атрибутів якості на протязі розробки продукту. В основі розробленого методу є оцінка метрик атрибутів характеристик якості [1, 2] на різних етапах розробки програмного забезпечення для отримання вибірки оцінок характеристик програмного продукту.

Для визначення оцінки якості ПЗ шукають середнє значення метрики досліджуваної характеристики:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \dots + x_n}{n}, \quad (1)$$

де  $(x_1, x_2, \dots, x_n)$  – множина значень метрики атрибутів програмного продукту, отримана на різних етапах розробки.

Після знаходження середнього значення оцінки, оцінюють наскільки оцінка є зміщена або незміщена. Незміщеною називають оцінку параметру  $\bar{x}$ , математичне сподівання якої дорівнює значенню параметру, що оцінюється при будь-якому об'ємі вибірки [3], тобто:

$$M[\bar{x}] = x. \quad (2)$$

Зміщеною називається оцінка параметру [3], для якої порушується умова (2):

$$M[\bar{x}] \neq x, \quad (3)$$