

УДК 004.415.2.045

\* О.Г.Харченко, \*\* В.В. Яцишин, \* І.Е. Райчев

## ІНСТРУМЕНТАЛЬНИЙ ЗАСІБ РОЗРОБКИ ТА КОМУНІКАЦІЇ ВИМОГ ЯКОСТІ ДО ПРОГРАМНИХ СИСТЕМ

\* Національний авіаційний університет

\*\* Тернопільський національний технічний університет

[raichev@i.com.ua](mailto:raichev@i.com.ua)

*В статті проаналізовані проблеми пов'язані з забезпечення якості програмних систем. Детально розглянуті технології розробки вимог якості та їх комунікації на етапи життєвого циклу ПЗ, вказані їх недоліки. Приведені результати подальшого розвитку методу розробки вимог якості, на основі використання моделей якості стандарту SQUARE, який дозволяє будувати моделі вимог в формалізованому уніфікованому вигляді. Розроблено алгоритм комунікації вимог на етапи життєвого циклу. Дано опис інструментального засобу, який реалізує запропонований метод.*

*В статье проанализированы проблемы, связанные с обеспечением качества программных систем. Детально рассмотрены технологии разработки требований качества и их коммуникации на этапы жизненного цикла ПО, отмечены их недостатки. Приведены результаты дальнейшего развития метода разработки требований качества на основе использования моделей качества стандарта SQUARE, который позволяет строить модели требований в формализованном унифицированном виде. Разработан алгоритм коммуникации требований на этапы жизненного цикла. Дано описание инструментального средства, которое реализует предложенный метод.*

*In the article the problems connected to support of quality of program systems are analysed. Technologies of development of requirements to quality and their communications on stages of software life cycle are in details considered, their lacks are marked. Results of the further evolution of a method of design of requirements to quality are resulted on the basis of use of models of quality of standard SQUARE which allows to build models of requirements in the formalized unified kind. The algorithm of the communications of requirements of stages of life cycle is developed. The description of tool resource which realizes the suggested method is given.*

**Ключові слова:** програмна система, вимоги до якості програмної системи, життєвий цикл, інструментальний засіб

### Вступ

У зв'язку з розширенням функціональності і зростанням вартості програмних систем значно зросли вимоги до їх якості, для забезпечення яких необхідно постійно вдосконалювати технології їх проектування.

Не дивлячись на значні досягнення у впровадженні формальних методів, та базованих на них інструментальних засобів проектування, значний об'єм робіт виконується «вручну» з використанням евристичних підходів [1]. Це значно збільшує вартість робіт, та підвищує ризик відхилення проектів через їх низьку якість та перевищення необхідних ресурсів. Так, за статистикою, в США тільки близько 53% програмних проектів завершується вдало, а з них лише в 42% забезпечуються всі вимоги, в тому числі і вимоги до якості [2].

Найменш формалізованими та забезпеченими засобами автоматизації є такі процеси життєвого циклу (ЖЦ), як розробка специфікацій вимог, проектування архітектури, запропонованого методу і принципи побудови інструментального засобу, який її реалізує.

та оцінювання якості ПС [1]. Особливо критичним у цьому плані є процес розробки специфікацій вимог, неякісне виконання якого породжує помилкову реалізацію всіх наступних етапів ЖЦ.

В даний час на практиці використовуються ряд технологій розробки та управління вимогами, які використовують рекомендації стандарту IEEE 830 [3], методів шаблонів, діаграм та інші. Однак, якщо застосування цих технологій до розробки функціональних вимог часто виявляється достатньо ефективним, то при їх використанні для побудови вимог якості виникають суттєві проблеми. В роботі [4] проведено порівняльний аналіз технологій розробки вимог, та запропоновано підхід до розробки вимог якості, який базується на концепції моделей якості стандарту ISO/IEC 25010.

Дана стаття містить результати подальшої формалізації та обґрунтування

### Аналіз сучасного стану проблеми

При розробці вимог до ПС реалізуються наступні процеси [1]:

- визначення та аналіз вимог;

- представлення вимог;
- трасування вимог;
- розробка процедур і критеріїв приймання результатів.

І хоча даний етап ЖЦ ПС є надзвичайно важливим, якість його виконання залишається достатньо низькою. Про це свідчить те, що 60% відхилених замовником програмних проектів не відповідали вимогам [5]. Одною з основних причин цього є недостатнє використання формальних методів для представлення вимог, що ускладнює трасування при їх відстеженні на етапах ЖЦ, а також трансформацію їх проміжних представлень при внесенні змін у вихідні вимоги [1]. Це також ускладнює впровадження засобів автоматизації цих процесів. Важливість застосування формалізованих методів підтверджується ще й тим, що біля 30% всіх помилок у вихідних вимогах виявлялись на етапі їх формалізації [5].

Слід відзначити, що для формалізованого представлення функціональних вимог розроблені та широко використовуються в багатьох проектах методи, базовані на представленні їх засобами мов MSC, SDL, UML і подальшої їх трансформації в форми, до яких можуть застосовуватись автоматизовані засоби їх комунікації. Однак загальні формальні процедури, які використовуються при цьому, виявляються малоефективними при роботі із значними за обсягом специфікаціями. Для подолання цих проблем була розроблена технологія заснована на використанні базових протоколів і теорії взаємодії агентів і середовищ [5].

Базові протоколи є ефективними для представлення функціональних вимог, які стосуються алгоритмів перетворень, оскільки вони представляють вимоги у вигляді трійок Хоара:  $\alpha \rightarrow \langle p \rangle \beta$ , де  $p$  - процес,  $\alpha$  і  $\beta$  – логічні формули, які визначають передумову і постумову. Але, оскільки вимоги якості формулюються в довільному вигляді, вони не можуть бути формалізованими до вигляду трійок Хоара, і тому застосування цього досить ефективного і потужного підходу виявляється неможливим.

Тому для представлення вимог якості авторами в ряді робіт [4, 6] були використані моделі якості ПС стандарту ISO/IEC 9126 та його останньої модифікації SQUARE. Розглянемо коротко засоби автоматизації, які використовуються при формулюванні та відстежуванні вимог.

На практиці в даний час використовують ряд технологій розробки ПЗ. Найбільш широко використовуються такі:

- Microsoft Solutions Framework (MSF);
- Rational Unified Process (RUP);
- Custom Development Method Oracle (CDMO).

Ці технології базуються на двох основних підходах: структурному та об'єктно-орієнтованому (ООП).

MSF є платформи-незалежною технологією, яка орієнтована на розробку ПЗ і розвиток інформаційної інфраструктури [6]. Засоби цієї технології підтримують розподілені обчислення та застосування технології "клієнт-сервер". Для проектування вимог MSF використовує метод шаблонів та UML діаграм. Засобами графічного моделювання та документування вимог до ПЗ є Microsoft Visio і Microsoft Office. Для управління проектом та контролю загального процесу створення ПЗ використовують засіб Microsoft Project.

Технологія RUP базується на принципах ООП до розробки ПС та уніфікованої мови моделювання UML. Підтримку цієї технології забезпечує група Rational фірми IBM. Для розробки вимог за ООП використовують діаграми класів та діаграми Use case, які відображають функціональність майбутньої ПС.

Засобом, що підтримує моделювання діаграм цього типу є Rational Rose. Інший засіб керування вимогами та їх документування – середовище Rational Requisite Pro. Основне призначення цього засобу полягає у наданні можливості відслідковування та зручного внесення змін у вимоги. При цьому вимоги представляють у текстовому вигляді з відображенням діаграм, змодельованих у Rational Rose.

Об'єктно-орієнтовний підхід до розробки вимог втілений також в автоматизованому засобі DOORS компанією Telelogic. Принцип проектування вимог заснований на структурі шаблону, що рекомендований у стандарті [3].

В основі технології CDMO лежить метод ORACLE CASE\* METHOD, який базується на визначенні об'єктів (сутностей) та зв'язків між ними, що фактично є структурним підходом. Технологія CDMO підтримується інструментальними засобами компанії Oracle і використовується під час створення автоматизованих інформаційних систем на основі реляційних баз даних. Для розробки та керування вимогами використовується засіб Oracle Designer, який дає змогу моделювати діаграми "сутність-зв'язок". Однак, за його допомогою можна лише визначати основні сутності всередині системи та зв'язки між ними, але неможливо описати систему загалом та процеси, які в ній протікають. В Oracle Designer

відсутня можливість представлення вимог якості та обмежень, що суттєво звужує область його застосування.

Засобами автоматизації, які підтримують принципи структурного підходу до проектування ПС, є також програмні продукти ARIS Toolset, ERwin Datamodeler, Process Modeler (BPwin).

Виходячи з результатів проведеного аналізу можна стверджувати, що тільки технології MSF та DOORS мають засоби формалізованого представлення вимог якості, які використовують метод шаблонів. Однак структура і класифікація рубрик шаблонів не уніфіковані, а тому при їх використанні можуть виникати неоднозначності трактувань, що звужує область застосування цих CASE-технологій.

Для перевірки відповідності готового програмного продукту заявленим у специфікації вимогам фірми-розробника використовують автоматизовані засоби тестування. Але засоби автоматизації технологічних операцій у процесі оцінювання якості ПЗ відсутні.

У роботах [4, 6] міститься обґрунтування та основні положення технології побудови специфікацій вимог та оцінювання якості ПС, яка базується на використанні концепції моделі якості ISO/IEC 9126. Ця технологія забезпечує формалізацію процесів відображення потреб замовника на специфікації вимог в уніфікованому стандартизованому вигляді і використання побудованих специфікацій при оцінюванні якості. Дана робота присвячена розробці CASE-засобу, що автоматизує процеси технології побудови специфікацій вимог та оцінювання якості ПС.

#### Формалізоване представлення та комунікація вимог якості на етапах ЖЦ ПС

Для розробки технології формування вимог якості ПС, які б можна було ефективно реалізувати при розробці інструментального засобу, необхідно побудувати моделі вимог якості з уніфікованою структурою і в стандартизованих термінах, та побудувати процедури їх комунікації.

В стандартах визначено три типи вимог якості, а саме: вимоги якості у використанні, вимоги зовнішньої якості та внутрішньої якості, які пов'язані з відповідними моделями якості [7].

Вимоги якості у використанні виражають потреби користувача бізнес-системи, які задаються в довільній формі і можуть бути записані у вигляді множини:

$$П = \{P_i, C_{ik}\}, \quad i = \overline{1, N}, \quad K = \overline{1, M_i}, \quad (1)$$

де  $P_i$  – і-та потреба користувача,  $C_{ik}$  – обмеження на і-ту потребу.

Для перетворення потреб (1) у вимоги якості у використанні відобразимо їх на структуру моделі якості [8], для чого для кожної потреби  $P_i$  підбираємо найбільш прийнятну характеристику якості  $H_i^u$ . Після цього, на основі аналізу бізнес-вимог та специфіки предметної області, для кожної характеристики  $H_i^u$  вибираємо один, або декілька атрибутів  $A_{ik}^u$ , які є вимірюваними і відображають ступінь задоволення вимоги якості по і-й характеристиці, а також відповідні метрики  $M_{ik}^u$ . В результаті отримуємо модель вимог якості у використанні:

$$V_{use} = \{H_i^u, A_{ik}^u, \overline{C}_{ik}^u, M_{ik}^u\}, \quad i \in N_u, \quad K = \overline{1, S_i} \quad (2)$$

Вимоги (2) являються вимогами якості до кінцевого продукту і використовуються при оцінюванні його якості та фінальному тестуванні.

Модель вимог зовнішньої якості буде мати подібну до (2) структуру, з тою лише різницею, що в ній для кожної характеристики  $H_i^e$  визначена множина підхарактеристик  $\Pi_{ij}^e, j = \overline{1, K_i}$  [8]. Враховуючи це, запишемо модель вимог зовнішньої якості в наступному вигляді:

$$V_{ex} = \{H_i^e, \Pi_{ik}^e, A_{ik}^e, C_{ik}^e, M_{ik}^e\}, \quad i \in N_e, \quad K = \overline{1, R_i}, \quad (3)$$

тут  $H_i^e$  – характеристики,  $\Pi_{ik}^e$  – підхарактеристики зовнішньої якості,  $A_{ik}^e, C_{ik}^e, M_{ik}^e$  – відповідно атрибути, обмеження та метрики.

Вимоги в формі (3) використовуються на етапі проектування архітектури ПС для розробки тестів, тестуванні, та оцінювання якості робочого продукту цього етапу, що відповідає рекомендаціям стандарту ISO 12207.

На етапі проектування програмних модулів (написання коду) вимоги якості записуються в термінах моделі внутрішньої якості для кожного модуля ПС, враховуючи його функціональність. По аналогії з (3) модель вимог внутрішньої якості для j-го модуля буде мати наступний вигляд:

$$V_{int}^j = \{H_i^j, \Pi_{ik}^j, A_{ik}^j, M_{ik}^j\}, \quad i \in N_j, \quad K = \overline{1, M^j}, \quad j = \overline{1, S} \quad (4)$$

Метрики для вимірювання атрибутів в (2), (3), (4) підбираються з відповідної частини стандарту [8].

Далі необхідно побудувати процедури комунікації вимог, тобто відображення вимог (2) на структуру (3), і далі, отримані вимоги в формі (3) – на модель вимог (4). Для цього можна використати алгоритм пошуку асоціацій на ієрархічній структурі [9], який був використаний

при визначенні вимог якості у використанні (2). Однак дослідження показали, що більш ефективними є експертні технології з використанням методу Сааті [10], або з використанням технології SQFD (Software Quality Function Deployment) [11].

Приведемо алгоритм методу SQFD стосовно задачі комунікації вимог. Суть алгоритму полягає в побудові “будинку якості” (рис. 1), результатом якого є матриця кореляцій  $\{S_{ij}\}$ ,  $i \in N_u, j \in N_e$  між  $i$ -ю характеристикою вимог якості у використанні  $H_i^u$  і  $j$ -ю підхарактеристикою моделі вимог зовнішньої якості (3).

Вимоги користувача визначені і записані в формі (2) виписуються зліва, а вгорі виписуються всі підхарактеристики зовнішньої моделі якості, які на думку аналітиків і експертів можуть впливати на реалізацію користувацьких вимог. Справа проставляються пріоритети користувацьких вимог  $P_{ri}^u$  визначені користувачами і аналітиками та нормовані до деякого інтервалу  $[a, b]$ . Значення пріоритетів можуть обчислюватись по алгоритмам визначення вагових множників при побудові згортки в задачі багатокритеріальної оптимізації [10].

Значення коефіцієнтів кореляції  $S_{ij}$  між різними користувацькими вимогами і підхарактеристиками моделі вимог зовнішньої якості визначають експерти і аналітики.

Після заповнення матриці кореляцій  $\{S_{ij}\}$  обчислюються пріоритети вимог зовнішньої якості по формулі

$$P_{rj}^e = \sum_{i \in N_u} P_{ri}^u \cdot S_{ij} \quad (6)$$

Після цього визначається граничне значення пріоритетів  $P_{ri}^e$ . Вимоги зовнішньої якості включаються лише ті підхарактеристики моделі вимог (3), для яких значення пріоритету перевищує граничне

$$P_{rj}^e > P_r^e \quad (7)$$

Після визначення коефіцієнтів кореляції і пріоритетів (6) записуються вимоги зовнішньої якості по моделі (3) в яку включаються підхарактеристики для яких пріоритет задовольняє умові (7).

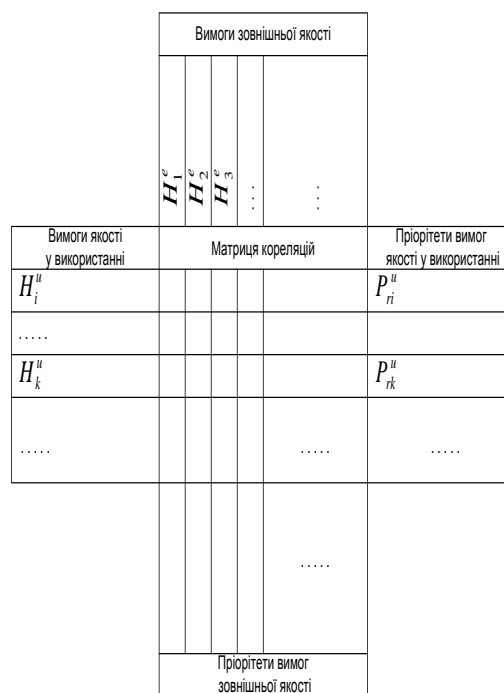


Рис. 1. Модель методу SQFD

На наступному кроці алгоритму будується «будинок якості» для визначення коефіцієнтів кореляції між підхарактеристиками зовнішньої якості, які були включені в вимоги, і підхарактеристиками внутрішньої якості.

Процедура подібна описаній вище реалізується для кожного програмного модуля.

### Структура та функціональність інструментального засобу розробки вимог до ПЗ та оцінювання його якості

При розробці цього програмного продукту були використані результати роботи [12]. Враховуючи принципи побудови вимог моделей якості і дані, які будуть зберігатись в репозиторії, розроблено функціональну схему інструментального засобу (рис. 2):

Потреби замовника, що описані в деякій структурі, вводяться в базу даних. Після цього оператором в режимі діалогу проводиться перетворення потреб у вимоги якості у використанні в формі (2). Для цього призначаються атрибути, на основі аналізу означень характеристик якості, та до кожного атрибута виконується підбір метрики, які зберігаються в БД. Сформовані користувацькі вимоги якості вводяться в БД вимог. Перетворення вимог якості у використанні вимоги зовнішньої якості виконується за алгоритмом SQFD або за алгоритмом класифікацій шляхом пошуку асоціацій.

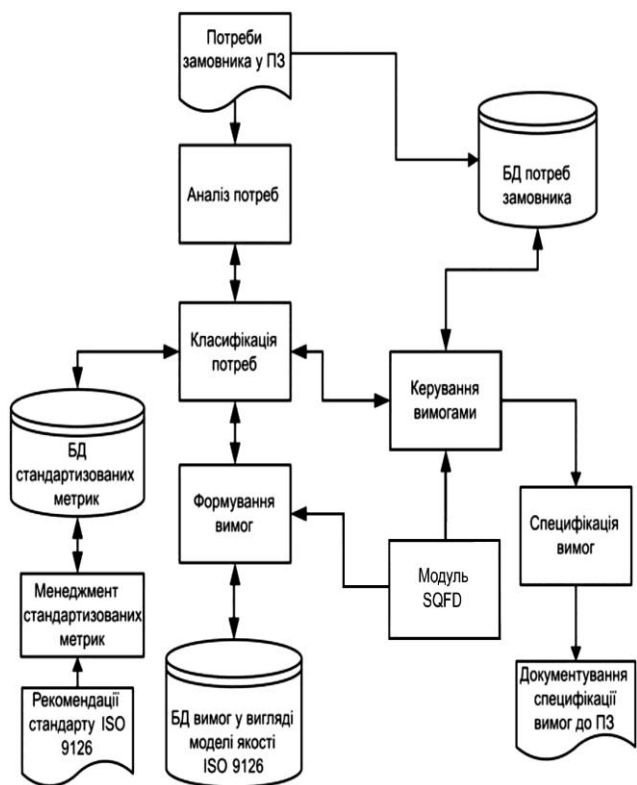


Рис. 2. Функціональна схема програмного комплексу розробки вимог якості

Оцінювання якості готового програмного продукту відбувається шляхом використання тих же репозиторіїв, але з урахуванням особливостей цього процесу. Зокрема, для кількісного відображення міри якості програмного продукту потрібно обчислити інтегральний показник якості. У роботі [4] наведено методи обчислень локальних, частинних та глобального показників якості ПЗ. Враховуючи аспекти технології оцінювання якості, у інструментальному засобі розроблено відповідну підсистему, функціональна схема якої наведена на рис.3.

Процес оцінювання якості ПЗ починається із звернення до репозиторію вимог, які сформульовані у вигляді характеристик, підхарактеристик, атрибутів та метрик моделей якості. Для кожного атрибута якості ставимо у відповідність пряму або непряму метрику і задаємо елементарну функцію для його оцінювання.

Для визначення ваги і впливу кожного атрибута запропоновано використати матриці попарних порівнянь, які дозволяє генерувати інструментальний засіб. Побудовані шаблони матриць попарних порівнянь оцінюють експерти. Відношення впливу між атрибутами опрацьовують методами статистичної обробки. В результаті отримуємо вагові коефіцієнти кожного атрибута. Аналогічно до атрибутів

будують матриці попарних порівнянь для підхарактеристик, що дозволяє визначити кількісні частинні показники якості по відповідних характеристиках та підхарактеристиках моделей якості. Усі вагові множники, визначені експертами, записують у репозиторій вагових множників, а результати оцінювання всіх рівнів моделі якості – у репозиторій кількісного оцінювання якості. На основі даних, які відображають кількісну міру якості, формують специфікацію процесу оцінювання та розробляють рекомендації щодо покращення якості програмного продукту.

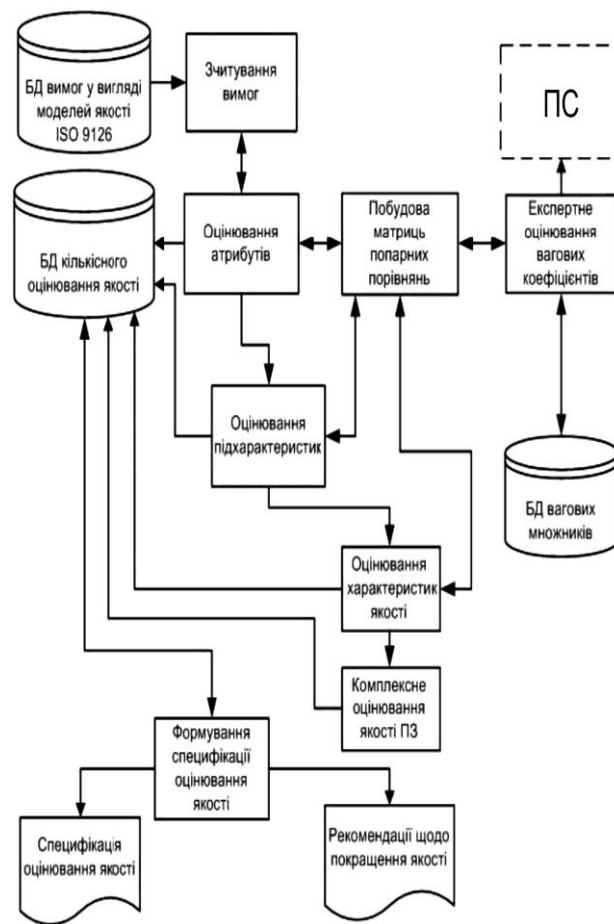


Рис. 3. Функціональна схема підсистеми оцінювання якості

Розроблений інструментальний засіб є web-орієнтованим і його можна використовувати як web-сервіс для підтримки процесів розробки вимог і оцінювання якості ПЗ. У сервісі забезпечено можливість віддаленого формування потреб замовника, специфікації вимог до ПЗ і визначення ступеня задоволення та відповідності вимог і потреб.

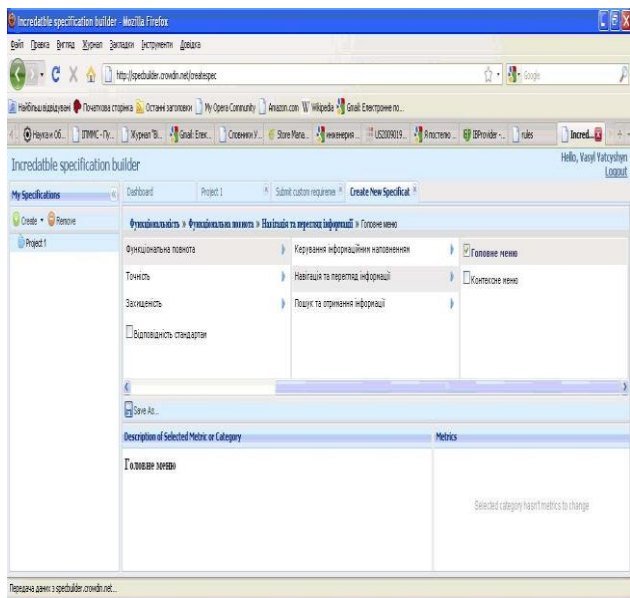


Рис. 4. Шаблон формування потреб замовника

Структурно CASE-засіб складається з двох частин: клієнтської і серверної. Клієнтська частина надає замовнику можливість задання потреб у вигляді шаблону моделей якості. Шаблон представляє собою набір стандартизованих характеристик, підхарактеристик, атрибутів та метрик. Замовник може вибирати властивості, які необхідно реалізувати (рис. 4).

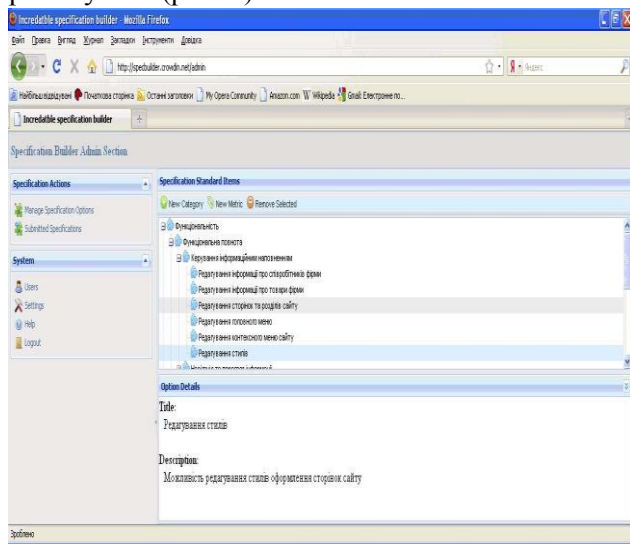


Рис. 5. Серверна частина

У інструментальному засобі реалізовано також можливість формування потреб замовника у формальному текстовому вигляді.

Процедура адаптації потреб, за даних таким чином, виконується на сервері шляхом їх класифікації за відповідними атрибутами, підхарактеристиками і характеристиками.

Серверна частина реалізує функції менеджера роботи з репозиторіями вимог,

потреб, вагових множників. На рис. 5 зображено інтерфейс серверної частини.

Адміністратор має можливість контролю, керування та наповнення усіх репозиторіїв відповідними даними.

У серверній частині забезпечено можливість розподілу прав доступу, налаштування зовнішнього інтерфейсу, спроектовано систему допомоги з користування інструментальним засобом.

Під час розроблення засобу автоматизації, орієнтованого на підтримку етапу аналізу вимог та оцінювання якості використано відкриті компоненти та засоби розробки. Зокрема, клієнтську частину реалізовано за допомогою компонентів JavaScript (Ext JS). Серверну частину розроблено за допомогою PHP 1 (Zend Framework) та MySQL 4.4 (InnoDB). Web-сервіс можна використовувати за адресою в мережі Internet: <http://specbuilder.crowdin.net/createspec>.

## Висновки

Одним із шляхів забезпечення необхідного рівня якості ПС є реалізація процесів контролю якості проміжних продуктів на всіх етапах життєвого циклу, як це передбачено стандартом ISO 12207. Труднощі при впровадженні цього в технології проектування ПС полягають у відсутності формалізованих конструктивних моделей вимог якості та процедур їх комунікації.

Використання моделей якості стандарту ISO25010 дозволило побудувати моделі вимог для трьох основних етапів проектування та розробити процедури, які реалізують зв'язки між ними. Розроблений на основі запропонованих моделей та процедур інструментальний засіб дозволяє автоматизувати ці процеси і таким чином підвищити ефективність технології проектування ПС. Слід також відзначити, що оскільки модель вимог містить атрибути і метрики характеристик якості, до яких формулюються вимоги, то засіб може використовуватись також при оцінюванні задоволення вимог.

## Література

1. Брауде Е. Технология разработки программного обеспечения / Е. Брауде. – СПб.: Изд-во "Питер", 2004. – 655 с.
2. Андон Ф.И. Основы инженерии качества программных систем / Ф.И.Андон, Г.И.Коваль., Т.М.Коротун, Е.М.Лаврищева, В.Ю.Суслов. –К.: Академперіодика, 2007. –672 с.
3. IEEE Std 830-1993, IEEE Recommended Practice for Software Requirements Specifications (ANSI).
4. Харченко О. Розроблення та керування вимогами до програмного забезпечення на основі

моделі якості / О. Харченко, В. Яцишин // Вісник ГДТУ. – 2009. – Том 14, №1. – С. 201-207.

5. A. Letichevsky., Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications A. Letichevsky, J. Kapitonova, A. Letichevsky Jr., V. Volkov, S. Baranov, V. Kotlyarov//ISSRE 2004, WITUL, Rennes, 4 November 2005.- pp 112 – 42.

6. Райчев І.Е. Принципи побудови моделі якості у використанні програмних систем /І.Е. Райчев, О.Г. Харченко // Збірник наукових праць інституту проблем моделювання в енергетиці. – 2007. – Вип. 39. – С. 31-38.

7. ISO/IEC 25030 Software engineering-Software product Quality Requirement and Evaluation (SQuARE), Quality Requirements, 2007.

8. ISO/IEC 9126 (1-4). Software engineering –

Product quality – Part 1: Quality model, Part 2:External metrics, Part 3:Internal metrics, Part 4:Quality in use metrics, 2001-2004/

9. Барсегян А.А. Технологии анализа данных / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. – СПб. : Изд-во " БХВ-Петербург", 2008. – 384 с.

10. Черноуцкий И.Г. Методы принятия решений / Черноуцкий И.Г.-БХВ-Петербург, 2005-408 С

11. Haag S, Raja M.K., SchKade h.h. Quality Function Deployment. Usage in Software Development//Comm. of the ACM.-V39.-N1.-1996/

12. Яцишин В.В. , CASE-технологія розроблення вимог до програмного продукту, та оцінювання його якості. / В.В. Яцишин, О.Г. Харченко // Науковий вісник НЛТУ України-Вин.20.2-2010.-Ст.277-285

### Відомості про авторів:



**Харченко Олександр Григорович** – кандидат технічних наук, професор кафедри комп'ютерних інформаційних технологій факультету комп'ютерних наук Національного авіаційного університету. Сфера наукових інтересів – технології розробки ПЗ, технології проектування ІС.



**Яцишин Василь Володимирович** – асистент кафедри комп'ютерної інженерії Тернопільського національного технічного університету ім. Івана Пулюя. Напрямки наукових досліджень – оцінювання якості ПС, технології проектування web-застосунків.

E-mail: [Yatcyshyn\\_v@ukr.net](mailto:Yatcyshyn_v@ukr.net)



**Райчев Ігор Едуардович** – кандидат технічних наук, доцент кафедри комп'ютерних інформаційних технологій факультету комп'ютерних наук Національного авіаційного університету. Коло наукових інтересів – технології оцінювання якості ПС, технології проектування ІС.

E-mail: [raichev@i.com.ua](mailto:raichev@i.com.ua)

Стаття надійшла до редакції 10.06.2010 р.