

УДК 004.415.5.

**О.В.Тегельман**

# Розробка вимог до СКБД на базі стандарту ISO 9126

**Національний  
авіаційний університет**

**Кафедра комп'ютерних  
інформаційних технологій**

**Науковий керівник –  
Харченко О.Г., д.т.н.,  
професор**

*Розглянуто питання розробки вимог якості до систем керування базами даних(СКБД). Пропонується використовувати підхід, що базується на моделі якості стандарту ISO 9126. У відповідності до цього підходу, вимоги користувачів до якості СКБД, що сформовані в термінах моделі якості у використанні, проектуються на вимоги зовнішньої якості для яких ставляться у відповідність вимоги внутрішньої якості.*

*Рассмотрен вопрос разработки требований качества к системам управления базами данных(СКБД). Предлагается использовать подход, который базируется на модели качества стандарта ISO 9126. В соответствии с этим подходом, требования пользователей к качеству СКБД, которые сформированы в терминах модели качества в использовании, проектируются по требованиям внешнего качества для которых относятся в соответствии требования внутреннего качества.*

*The question of development of requirements of quality is considered to control system by the bases of data(СКБД). It is suggested to take approach, which is based on the model of quality of standard of ISO 9126. In accordance with this approach, requirements of users to quality of СКБД, that the internals formed in terms of model are in the use, designed on the requirements of external quality for which belong in accordance of requirement of internal quality.*

**Ключеві слова:** СКБД, якість програмних систем, формування вимог якості, база знань, моделі якості

## Вступ

Забезпечення якості програмних систем при їх проектуванні є надзвичайно важливим в даний час, коли на ринку програмних систем (ПС) спостерігається жорстка конкуренція. Це особливо важливо для ПС масового використання, до яких відносяться і системи керування базами даних (СКБД).

Для підтвердження заявлених характеристик якості проводяться процедури атестації та сертифікації. Однак використовувані технології створення ПС орієнтовані і в основному на задоволення функціональних вимог. І хоча на даний час вже розроблено третє сімейство стандартів якості ПС [1], їх впровадження в практику йде дуже повільно. Основними причинами такої ситуації є недостатнє використання формальних методів, та відсутність ефективних інструментальних засобів автоматизації процесів забезпечення якості.

В архітектурі процесів життєвого циклу (ЖЦ) ПС представлено два процеси, які стосуються якості, це

- Підтримуючий процес «забезпечення гарантії якості».

- Організаційний процес «управління якістю».[2]

Перший з процесів реалізується шляхом впровадження стандартів якості і відповідних процедур в практику розробки ПС, а другий процес – «управління якістю» полягає в моніторингу якості проміжного та кінцевого продукту на стадіях ЖЦ ПС. Для цього необхідно представити вимоги замовника до якості ПС і виконати процедури комунікації (трасування) цих вимог на стадії ЖЦ.

Для розробки таких процедур необхідно розробити формалізовані та стандартизовані моделі представлення вимог та запропонувати деякі формальні математичні алгоритми комунікації вимог на етапах ЖЦ.

Ці задачі можна вирішити на основі стандарту з якості [1] в якому визначено три типи моделей якості: у використанні (експлуатаційна), зовнішньої та внутрішньої. Представивши вимоги якості за цими моделями й віднісши їх до відповідних фаз ЖЦ можна реалізувати процес моніторингу якості.

## Аналіз технологій формування вимог

Для формування вимог до ПС можна використовувати стандарт [3], в якому задана стру-

ктура і зміст розділів специфікації вимог, які поділяються на два типи: загальні вимоги, або С-вимоги, та детальні вимоги, або D-вимоги. Вимоги замовника подаються, як загальний опис компонентів та функцій ПС. В детальних вимогах передбачено декларування вимог до декількох властивостей ПС, які можна віднести до характеристик якості. В загальному випадку, структура та зміст розділів специфікації вимог, згідно зі стандартом IEEE\ANSI 830, є ієрархічним слабоформалізованим деревом.

При використанні даної технології виникає ряд проблем, пов'язаних із представленням вимог. Основними з недоліків є відсутність повного набору характеристик якості та формалізованих означень характеристик. Це робить неможливим повне та адекватне відображення потреб в ПС на вимоги, а також призводить до неоднозначного трактування термінів різними учасниками процесу розробки.

Крім того, у стандарті відсутні рекомендації щодо вибору атрибутів та метрик з допомогою яких можна було б оцінити міру задоволення вимог. Враховуючи вище перелічені недоліки, бачимо, що технологія, базована на IEEE\ANSI 830, не задовольняє потреби розробки ПС.

При проектуванні вимог до ПС ряд фірм-розробників використовують шаблони та спеціалізовану мову, що дозволяє формалізувати потреби до ПС та розділити описову структуру та дані при їх заповненні [4]. Перевагою даного підходу, відносно технології, базованої на рекомендаціях IEEE 830, є те, що методика використання шаблонів та відповідної мови дозволяє суттєво формалізувати формулювання та представлення вимог, а це, в свою чергу, робить її більш гнучкою та прийнятною. Особливо важливо це при внесенні змін у вимоги. Використовуючи шаблони при формулюванні вимог, необхідно визначити та зарезервувати перелік слів або фраз, які б однозначно трактувались як стороною замовника, так і стороною розробника. Якщо це зроблено, тобто узгоджено мову для формулювання вимог, тоді спосіб, а відповідно й засіб трактування вимог, стає досить гнучким. Для прикладу, якщо зарезервувати слово «повинен» та його інтерпретації, то спосіб формулювання вимог буде мати наступний вигляд: «Система <повинна> опис можливостей системи».

В даному випадку використовується спеціальна мова, що представляє собою набір зарезервованих та узгоджених фраз. Аналізуючи вище наведений приклад, можна виділити певну структуру та порядок слів при формулюван-

ні вимог. Структурований порядок слів та спеціальна мова формують шаблон, що застосовується в проекті при розробці вимог до ПС. Вимоги з обмеження також можна представити у вигляді шаблонів.

До недоліків вище наведеного підходу можна віднести те, що він є не стандартизованим, тобто кожен розробник на свій розсуд конструє шаблони і вводить суб'єктивну класифікацію даних. суттєвим недоліком є складність відображення вимог якості до ПС в структурі шаблону та його неоднозначність. При використанні даної технології досить важко оцінити ступінь задоволення вимог якості ПС при їх атестації та сертифікації, оскільки відсутні загальні рекомендації щодо вибору метрик та атрибутів.

#### Розробка вимог якості СКБД

Для оцінки якості системного програмного забезпечення, до якого відносяться і СКБД, використовуються наступні технології:

- Інтегральні тести SPEC (Standard Performance Evolution Corporation);
- Тести TCP (Transaction Processing performance Council). [5],[6]

Тести SPEC дозволяються оцінити об'єм обчислень, або кількість ітерацій за фіксований час, та швидкість обчислень, або час виконання однієї ітерації тесту.

Тести TCP є більш спеціалізованими і вони призначені для оцінки продуктивності системи ітеративної обробки транзакцій (OLTP) при виконанні запитів до баз даних (БД) та формуванні звітів.

Однак для кінцевого користувач СКБД крім продуктивності важливими є також такі показники якості, як надійність, зручність у використанні, практичність та інші. Тому, необхідно розробити технологію, яка б дозволяла відображати вимоги якості кінцевого споживача на характеристики якості СКБД, які можна контролювати на етапах ЖЦ ПС при проектуванні, а також при сертифікації і виборі конкретного варіанта СКБД на ринку.

Одним з шляхів вирішення цієї задачі є використання рекомендацій стандартів з якості ПС ISO\IEC 25010, в якому описані моделі якості ПС та стандарту ISO\IEC 25030, в якому сформовані положення по розробці вимог якості.

Первинні вимоги якості до СКБД формулює кінцевий споживач в довільному текстовому вигляді, але для перетворення їх до формалізованого вигляду нами запропоновано наступний шаблон.

$$R_{user}=\{A1, A2, A3\}, \quad (1)$$

де  $A_1$  – функція виконувана СКБД,  
 $A_2$  – характеристика якості виконуваної функції,

$A_3$  – метрика вимірювання  $A_2$ .

Конкретний варіант  $A_1$  вибирається з переліку, який зберігається в базі даних.

На наступному кроці вимоги (1) перетворюються у вимоги якості у використанні, які записуються у вигляді:

$$R_{inuse} = \{H_i, A_i, M_i\}, \quad (2)$$

де  $H_i$  – характеристика якості,

$A_i$  – атрибут якості,

$M_i$  – метрика для вимірювання якості

Для цього пропонується використовувати базу знань (рис. 1), в якій зберігається інформацію про міцність зв'язків (кореляцію) між характеристиками й атрибутами якості у використанні (2) та вимогами кінцевого користувача (1). Матриця кореляції зв'язків  $\{K_{ij}\}$  між  $A_i$  з (1) та  $H_j$  з (2) визначається експертним шляхом або з використанням алгоритмів асоціативної класифікації.

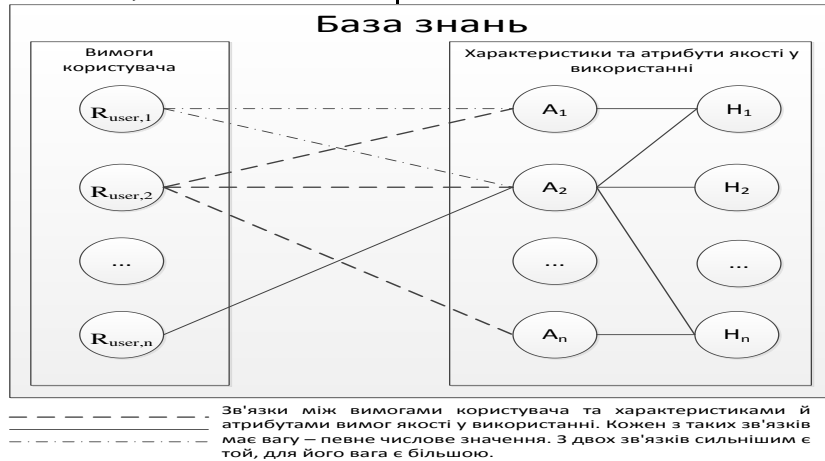


Рис. 1. Схематичне представлення бази знань та інформації, що в ній зберігається

Нижче приведено приклад проектування вимог користувача на вимоги якості у використанні.

Нехай кінцевий користувач висунув наступні вимоги:

- 1) Швидкість відклику на запит повинна становити від 1 до 10 секунд;
- 2) Після збоїв в електроживленні відновлення повинно тривати не більше 10 хвилин;

3) Об'єм даних, який може зберігатись повинен становити не менше 10 Гб з можливістю розширення.

Для зручності їх використання сформуємо таблицю з вимогами (таблиця 1), яка буде відповідати запропонованому шаблону представлення вимоги користувачів (2).

Таблиця 1 Вимоги користувачів у структурованому представленні

Умове позначення	Функція	Характеристика	Метрика
R1	Обробка даних	Продуктивність	Секунди
R2	Відновлення	Безпека	Хвилини
R3	Збереження даних	Задоволеність	Гб

Після того, як вимоги користувачів були приведені у вигляд, який є зручним для роботи, відбувається проектування цих вимог на вимоги якості у використанні за допомогою інформації, яка міститься в базі знань або з викорис-

танням експертних методик. Отримані вимоги якості у використанні повинні відповідати шаблону (1). Кінцевий результат приведено в таблиці 2.

Таблиця 2 Вимоги якості у використанні

Умове позначення	Вимога користувача	Характеристика	Атрибут	Метрика
A1	R2	Безпека	Відновлення після збоїв	Мінімальний час для відновлення, хвилини
A2	R1	Продуктивність	Відгук на запит	Мінімальний час відгуку, секунда
A3	R3	Задоволеність вимог користувача	Об'єм даних, що можуть бути збережені	Максимальний об'єм даних, Гб

Наступним іде етап побудови моделі та визначення вимог зовнішньої якості

### Побудова моделі зовнішньої якості та визначення вимог до СКБД

Побудова моделі зовнішньої якості для СКБД розпочинається з етапу відбору з числа наявних в стандарті ISO 25010 характеристик та підхарактеристик з метою побудови модифікованої моделі якості. Модифікована модель зовнішньої якості до СКБД приведена на рис. 2.

Для цього використовується методологія QFD, на основі якої будується матриця кореляції між атрибутами якості у використанні та підхарактеристиками зовнішньої якості [7].

Після того, як модель якості побудовано можна визначити атрибути для підхарактеристик, які є найбільш суттєвими для конкретно взятого випадку. В таблиці 3 приведено характеристики, підхарактеристики та визначені для них атрибути у відповідності до (2).

Таблиця 3 Характеристики, підхарактеристики та атрибути зовнішньої якості

Характеристика	Підхарактеристика	Атрибут	Умовне позначення
Надійність	Можливість до відновлення	Наявність механізму резервного копіювання даних	B1
		Відновлення в результаті поломки апаратного забезпечення	B2
Ефективність	Використання ресурсів	Механізм розподіленого збереження даних	B3
		Архівування даних	B4
Функціональність	Захищеність	Наявність механізму керування користувачами та групами	B5
		Наявність механізму шифрування даних	B6

Для визначення міцності зв'язку можна використовувати різні варіанти шкал (десятибальна, з використанням символів + та - тощо). В даній роботі пропонується використовувати символну шкалу з наступною градацією:

«--» - дуже слабкий зв'язок;  
«-» - слабкий зв'язок;  
«+» - міцний зв'язок;  
«++» - дуже міцний зв'язок.

Таблиця 4 Матриця кореляції

	B1	B2	B3	B4	B5	B6
A1	++	++	+	++	--	--
A2	--	--	-	-	+	+
A3	+	-	++	++	--	+

Для визначення найбільш суттєвих атрибутів зовнішньої якості можна скористатись наперед визначеним правилом – наприклад, атрибут повинен мати від 3 і більше знаків +. У відповідності до побудованої матриці маємо наступні атрибути зовнішньої якості:

B1 – Наявність механізму резервного копіювання;

B2 – Відновлення роботи після поломки апаратного забезпечення;

B3 – Механізм розподіленого збереження даних;

B4 – Архівування даних.

Після того, як визначено атрибути зовнішньої якості, їх потрібно спроектувати на атрибути внутрішньої якості. Для цього використовується той самий підхід, що і для зв'язку між якістю у використанні та зовнішньою якістю: побудова моделі внутрішньої якості до СКБД, визначення атрибутів для внутрішньої якості, побудова матриці кореляції з використанням методу QFD та визначення атрибутів, які мають найвищий ступінь зв'язку з атрибутами моделі зовнішньої якості.

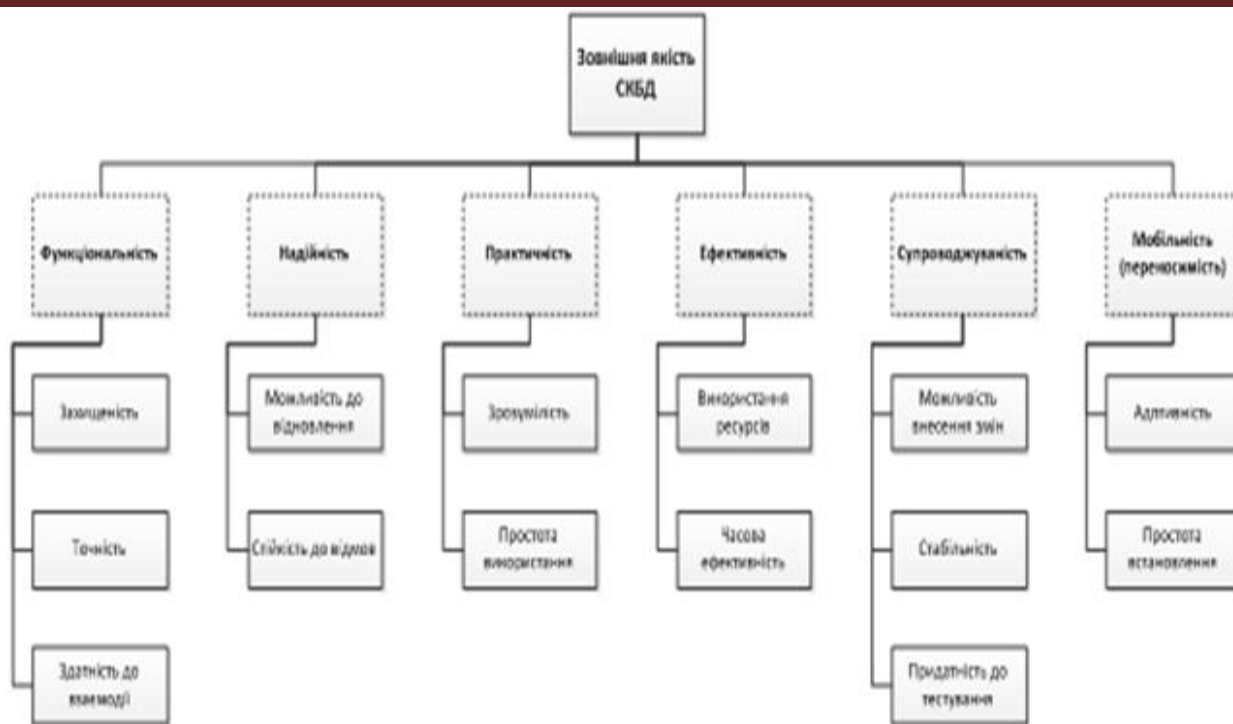


Рис. 2. Модифікована модель зовнішньої якості для СКБД

### Висновки

Використання моделей якості ПС стандарту ISO 25010, методології QFD та методів класифікації дозволило сформулювати вимоги якості до СКБД на етапах ЖЦ ПС в стандартизованому та формалізованому вигляді. Отримане представлення вимог дає змогу забезпечити моніторинг якості при проектуванні СКБД, а також при сертифікації відповідності їх заявленим характеристикам. Запропонований алгоритм послідовного перетворення вимог якості користувача до СКБД у вимоги внутрішньої та зовнішньої якості надає також певний інструмент для обґрунтування вибору СКБД.

### Використана література

1. *ISO/IEC 25010. Software engineering-Software product Quality Requirements and Evaluation - Quality model.* /2006
2. *ISO/IEC 12207. Information technology – Software life cycle processes* 2002.
3. *IEEE Std 830, IEEE Recommended Practice for Software Requirement Specification.*
4. Requirement Engineering/ Elizabeth Hull , Ken Jekson, Jeremy Dick// Springer Science + Buisness Media, 2005.-240 p.
5. Standard Performance Evolution Corporation (SPEC). <http://www.spec.org>
6. Transaction Processing Performance (TCP). <http://www.tcp.org>
7. Проектування архітектури WEB-застосування на основі моделі якості/ Харченко О.Г., Галай І.О., Боднарчук І.О., Яцишин В.В. // Інженерія програмного забезпечення №4, 2010.-26-34 с.

### Відомості про авторів



**Тегельман Олег Вікторович** – студент 4 курсу Національного авіаційного університету, спеціальність програмна інженерія. Наукові інтереси – інженерія програмного забезпечення, рефакторинг баз даних, ERP-системи, оцінка якості програмного забезпечення

E-mail: [tegelmann@gmail.com](mailto:tegelmann@gmail.com)



**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.

E-mail: [harchenko\\_a1@voliacable.com](mailto:harchenko_a1@voliacable.com)