

## **ПОБУДОВА ТА АНАЛІЗ МОДЕЛІ ЯКОСТІ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

*У статті розглянуті питання встановлення множини показників якості архітектури програмної системи на основі міжнародних стандартів якості програмного забезпечення. Проведено аналіз можливості вираження нестандартизованих критеріїв якості архітектури в термінах стандарту ISO25010. Для виділення найсуттєвіших критеріїв якості для оцінювання архітектури розроблюваної системи запропоновано використовувати експертну технологію на основі методу парних порівнянь стосовно реалізації кожного з атрибутів якості системи відповідними критеріями архітектури.*

*Ключові слова: архітектура програмної системи, якість архітектури програмної системи, критерії якості архітектури програмної системи, модель якості архітектури програмної системи.*

I.O. BODNARCHUK

Ternopil Ivan Pul'uj National Technical University

### **CREATION AND ANALYSIS OF QUALITY MODEL FOR SOFTWARE ARCHITECTURE**

*Abstract – There are examined questions of definition of the set of quality criteria for software system architecture on the base of international standards of software quality in the paper. The analysis of literature, which refers to evaluation of architecture quality, is carried out. On the base of the analysis main disadvantages of these methods are eliminated. The analysis of possibility to express nonstandardized indices of architecture quality in terms of standard ISO 25010 is done. Since this standard regulates the set of quality indices of ready-to-use software product the problem to interpret this set of characteristics for architecture of software system as intermediate product during software design is stated. As technic for such interpretation the method of logical analysis is offered.*

*As result for each subcharacteristic of software product quality described the way of its application for the evaluation of architecture quality. In addition to standardized indices of quality in the paper characteristics of architecture, which considerably influence on the quality of architecture but not mentioned in the standards, are examined as well. These characteristics are cohesion and coupling, and conceptual integrity too.*

*The technics for evaluation of these characteristics are proposed. As following step in the paper the method for selection of established set of software quality indices for architecture those, which are most important for designed software system is offered. To do this the expert method of pairwise comparisons (simple algorithm of selection) and QFD method. First one is offered as a tool for defining of priorities of quality indices for software system, which will be realized in the architecture decisions. With QFD method the weight coefficients of quality indices for the architecture will be calculated, and after comparison of these weights with defined threshold value the most significant are selected for architecture quality of designed software system.*

*Key words: architecture of software system, quality of software system architecture, quality criteria of software system architecture, model of quality for software system architecture.*

### **Вступ**

Вимоги до темпів створення сучасних програмних систем (ПС) ніяким чином не зменшують вимог замовників до якості створюваних програмних продуктів. Тому розробники змушені використовувати формальні методи та системні підходи при створенні програмних систем.

Проектування архітектури, як один з перших етапів життєвого циклу програмної системи, має велику важливість, оскільки зроблені на цьому кроці рішення, окрім функціональності системи, задають також і такі характеристики, як зручність у використанні, надійність, захищеність та ін. Тобто, якість проєктованого програмного продукту визначається якістю архітектурних рішень, і якість архітектури – це теж її властивість. Тому потрібно мати визначення терміну архітектури ПС, яке буде відображати це поняття таким чином, щоби до нього можна було застосувати запропоновану у статті модель якості архітектури.

В різних літературних джерелах зустрічаються різні означення архітектури ПС. Зокрема, у [1] архітектура означена як система конструкторських рішень, котрі в процесі подальшої розробки системи змінюються найменше. При цьому в [1] наголошено, що одна і та ж архітектура може бути описана різними способами, кожен з яких виражає бачення проєктованої системи певної групи учасників розробки – замовників архітекторів, менеджерів тощо. Очевидно, що ці описи відображатимуть пріоритети кожної з груп, що неминуче спричинить конфлікти інтересів. Роль архітектора у цьому випадку – досягнути таких компромісів, щоби рівень задоволення кожної вимоги був максимальний. Дане означення добре відображає роль архітектури для подальшого процесу створення ПС, але тут важко говорити про якісні характеристики конструкторських рішень.

У двох інших роботах [2, 3] архітектура означається, як зовнішнє представлення елементів системи, їх властивостей та зв'язків між ними. Також у [2] вказано, що архітектура системи може бути виражена по різному в залежності від групи зацікавлених осіб через різні моделі архітектури. По аналогії до [1] робиться акцент на незмінність певних елементів розробки, на рішення, котрі відображають концепцію роботи майбутньої системи у своїй предметній області. У даному означенні мова йде вже про певні елементи та

зв'язки між ними. Це дає змогу формувати якісні характеристики архітектури на основі характеристик якості елементів та зв'язків.

Е. Брауде у своїй роботі [4] розглядає архітектуру як проект системи на найвищому рівні абстрагування. Всі подальші кроки, на його думку, – це детальне проектування. Але і тут можна побачити наголос на незмінності прийнятих рішень, як основи для всіх наступних етапів створення ПС. Це означення теж є проблемним для створення для нього моделі якості, оскільки є дуже загальним через високий рівень абстрагування при представленні ПС.

В загальному можна сказати, що ці означення не заперечують одне одного, а доповнюють. Тобто архітектура ПС є репрезентацією цієї системи на найвищому рівні абстрагування з вказанням основних елементів цієї системи та зв'язків між ними. Причому архітектура кожної системи може бути представлена з різних точок зору з використанням різних моделей.

Означення архітектури ПС, яке акумулювало в собі всі перераховані вище особливості, наведено у стандарті ISO 42010 [5], який регламентує методика опису архітектури. Згідно цього стандарту архітектура ПС – це фундаментальні концепції властивостей системи, втілені у елементах, зв'язках і принципах дизайну і розвитку. Таке означення не вказує перелік концепцій, а тому може передбачати також врахування якості при проектуванні архітектури.

На основі цього аналізу означень можна сформулювати визначення терміну архітектури ПС, яке використовуватиметься у цій статті.

Архітектура при цьому буде розумітись, як набір компонентів, які реалізують логіку роботи програми та зв'язки між цими компонентами, що забезпечують їх взаємодію та конфігурацію компонентів. Архітектура ПС забезпечує абстрактну модель високого рівня для представлення структури і ключових властивостей ПС і створює передумови забезпечення якості ПС.

Проектування архітектури ПС як елемент життєвого циклу цієї системи регламентується стандартом ISO 12207 [6]. Це стандарт передбачає, що якість програмної системи базується на реалізації процесів "забезпечення якості" та "управління якістю" на всіх етапах життєвого циклу (ЖЦ) програмної системи.

Забезпечення якості гарантується використанням відповідних норм і стандартів на етапах ЖЦ.

Управління якістю означає контроль (моніторинг) якості проміжних продуктів на кожному етапі життєвого циклу ПС: якість вимог до проєктованої системи, якість її архітектури, яка на наступному кроці реалізовуватиметься у якості модулів, і далі – якість всього програмного продукту.

Отже, оптимальне рішення з вибору архітектури ПС на основі моделі якості дозволяє уникнути суттєвих затрат при подальшому проектуванні системи, а тому розробка методики створення архітектури з врахуванням показників якості є важливою задачею.

### Постановка і обґрунтування проблеми

Необхідність враховувати показники якості при розробці архітектури ПС обґрунтовується твердженням про те, що спроектована архітектура повинна забезпечити реалізацію вимог якості до ПС. Тобто, показники якості архітектури повинні забезпечувати вимоги якості, які ставляться до готової системи. Крім того, як згадувалось вище, згідно ISO 12207 [6] процес управління якістю має бути впроваджений на всіх етапах ЖЦ ПС. А, отже, і на етапі збору вимог до системи та при проектуванні архітектури.

Збір вимог до ПС – задача, яка розглядалась досить часто, і у цьому напрямку є певні результати, наприклад, роботи [7, 8, 9]. Але що стосується розробки архітектури, то ця галузь досліджена на сьогодні недостатньо. Як наслідок – відсутність уніфікованих підходів до задач вибору архітектури на основі показників якості.

На теперішній час для опису та оцінювання якості готової програмної системи використовуються стандартизовані характеристики і модель якості ПС, описана у стандарті ISO 25010 [10]. Відповідно до цього стандарту така модель якості може бути подана у вигляді виразу:

$$Q_{prod} = \{H_i^{prod}, S_{ik}^{prod}, A_{ik}^{prod}, C_{ik}^{prod}, M_{ik}^{prod}\}. \quad (1)$$

Тут  $H_i^{prod}$  –  $i$ -та характеристика якості програмного продукту;

$S_{ik}^{prod}$  –  $k$ -та підхарактеристика  $i$ -ї характеристики якості;

$A_{ik}^{prod}, C_{ik}^{prod}, M_{ik}^{prod}$  – відповідно  $k$ -й атрибут, обмеження атрибуту та метрика атрибуту для  $i$ -ї підхарактеристики якості продукту.

Зазначимо, що атрибути, обмеження та метрики для підхарактеристик якості означені у стандартах ISO 25021 [11] та ISO 25023 [12]. Варто відмітити також, що таку ж модель якості можна отримати для програмного продукту на основі старішого стандарту ISO 9126 [13], який на сьогодні також є діючим поряд з сімейством стандартів ISO 250xx.

Для архітектури ПС такої стандартизованої моделі не існує. Про це прямо вказано у стандарті ISO 42010 [5]. Тому в процесі створення архітектури та вирішення задачі її оцінювання на основі показників якості використовуються різні набори характеристик, що спричиняє не завжди однозначне їх розуміння та оцінювання.

Зокрема, у одній з найґрунтовніших робіт з проектування архітектури ПС [3] розглядаються три

групи показників якості:

- атрибути якості системи;
- комерційні атрибути;
- атрибути якості архітектури (концептуальна цілісність).

Процес проектування архітектури виконується з врахуванням тільки атрибутів якості системи, а вплив вибраного архітектурного рішення на якість ПС досліджується логічним аналізом через реалізацію "сценаріїв атрибутів якості". Ці атрибути якості, котрі реалізуються через сценарії, показані у таблиці 1.

Таблиця 1

**Атрибути якості ПС, які реалізуються архітектурним рішенням, згідно [3].**

<b>Атрибут якості</b>	<b>Кількісні характеристики</b>
Готовність	Тривалість відновлення; час перебування системи у працездатному стані; готовність; час до забезпечення повної / часткової працездатності.
Модифікованість	Кількість модифікованих елементів; грошова оцінка модифікації елементів; ступінь задіяння інших функцій та атрибутів якості.
Продуктивність	Затримка; граничний термін; пропускна здатність; нестійкість; коефіцієнт невдач; втрата даних.
Безпека	Часові/трудова/інші ресурси, необхідні для обходу системи захисту; імовірність виявлення атаки; імовірність ідентифікації особи, що проводить атаку; процент сервісів, готовність котрих може бути збережена при DOS-атаці; ступінь пошкодження даних/служб та прецедентів відмов в обслуговуванні легальних користувачів.
Зручність тестування	Процент виконання виконуваних операторів; імовірність відмови в разі несправності; тривалість тестування; Довжина найбільшої зі всіх виявлених за результатами тестування послідовності залежностей.
Практичність	Тривалість виконання задачі, кількість помилок, кількість розв'язаних проблем, ступінь задоволеності користувача, підвищення освітнього рівня користувача, процентне відношення успішно завершених операцій до їх загального числа, об'єм втраченого часу/даних.

Можна встановити відповідність між показниками з таблиці 1 та показниками стандарту [10] на рівні логічного аналізу. Така відповідність представлена у таблиці 2.

Аналізуючи таблицю 2, можна побачити, що вибрані в [3] атрибути відносяться до різних моделей якості: задоволеність та ефективність є показниками якості у використанні, а решту атрибутів відносяться до якості програмного продукту. Очевидно, що таке змішування вносить незручності при застосуванні до оцінювання архітектури. Для характеристики самої архітектури, а не реалізації певної характеристики ПС через архітектурне рішення, автори в [3] пропонують використовувати концептуальну цілісність, коректність, завершеність, можливість побудови. Кількісні характеристики цих атрибутів не описані.

Якщо проаналізувати іншу працю [4] з області розробки архітектури ПС автора Е. Брауде, то для опису її якості виділяються такі характеристики: розширюваність, зміна (модифікованість), простота, ефективність-швидкість та ефективність-зберігання. Для вибору архітектури автор пропонує використовувати зважені відносні оцінки вибраних альтернатив. Крім того, автор часто оперує поняттями зв'язності та зчеплення компонентів архітектури для визначення придатності проектованої системи до модифікованості (висока зв'язність всередині модуля та низьке зчеплення між модулями).

Таблиця 2

**Відповідність між атрибутами з таблиці 1 та стандартними атрибутами згідно [10]**

<b>Атрибут якості згідно [3]</b>	<b>Стандартна характеристика (підхарактеристика) якості</b>
Готовність	Відмовостійкість, поновлюваність
Модифікованість	Модернізованість
Продуктивність	Ефективність
Безпека	Захищеність
Зручність тестування	Тестованість
Практичність	Задоволеність

По аналогії до попереднього прикладу всі ці характеристики можуть бути виражені у термінах стандарту [10]. Це показано у таблиці 3.

Таблиця 3 також відображає використання для якості архітектури характеристик, що відносяться до різних моделей якості та призначених для використання різними категоріями учасників процесу проектування.

Тому, оскільки задача стандартизації якості архітектури ПС на сьогодні невирішена, є потреба хоча б в уніфікації показників якості архітектури на основі існуючих стандартів.

**Відповідність характеристик якості архітектури згідно [4] характеристикам згідно стандарту [10]**

Атрибут якості згідно [4]	Стандартна характеристика (підхарактеристика) якості
Розширюваність	Здатність до модернізації
Зміна (модифікованість)	Здатність до модернізації
Простота	Придатність до використання
Ефективність-швидкість	Ефективність
Ефективність-зберігання	Функціональність
Зв'язність	Стабільність
Зчеплення	Стабільність

На основі виконаного аналізу літературних джерел робимо висновок, що для уніфікації характеристик якості архітектури можна застосовувати артефакти стандарту ISO 25010. Але не всі характеристики ПС є характеристиками якості архітектури. І навпаки – для розробки архітектури на основі показників якості є необхідність у застосуванні характеристик, котрі не згадані у стандартах [10,13].

Отже, в кінцевому результаті, загальна задача може бути сформульована наступним чином. Потрібно розробити метод отримання критеріїв якості архітектури. Для цього має бути встановлено множину всіх критеріїв, які можуть використовуватись для оцінювання якості архітектури. Тому потрібно інтерпретувати стандартизовані характеристики якості програмної системи стосовно характеристик якості архітектури. Також слід до загальної моделі якості архітектури включити характеристики, котрі у стандарті ISO 25010 не вказані, але є важливими для розробки архітектури.

Таким чином, модель якості архітектури ПС може бути записана наступним чином:

$$Q_A = \{H_i^A, S_{ik}^A, A_{ik}^A, C_{ik}^A, M_{ik}^A\}, \quad (2)$$

де  $H^A = H^{std} \cup H^{nstd}$  – характеристики якості архітектури, як об'єднання інтерпретованих стосовно архітектури стандартизованих  $H^{std}$  характеристик якості програмного продукту та нестандартизованих  $H^{nstd}$  характеристик якості архітектури;

$S_{ik}^A$  – підхарактеристики характеристик якості архітектури;

$A_{ik}^A, C_{ik}^A, M_{ik}^A$  – відповідно атрибути, обмеження та метрики підхарактеристик якості архітектури, які визначаються предметною областю, для якої розробляється програмна система.

Далі, з використанням експертних методів встановлюємо найсуттєвіші для даної предметної області критерії якості архітектури зі всієї множини характеристик якості (2). На цій основі можна на виконувати порівняльне оцінювання альтернативних архітектурних рішень проєктованої системи з метою вибору того, що максимально задовольняє поставленим до системи вимогам.

### Модель якості для архітектури ПС

Як було сказано вище, для проєктування архітектури на основі показників якості крім стандартизованих характеристик, інтерпретованих стосовно архітектури, є потреба у використанні тих, котрі у стандартах з якості ПС не описані.

Тому спочатку виконаємо інтерпретацію характеристик та підхарактеристик стандарту ISO 25010 стосовно якості архітектури ПС, а потім проаналізуємо характеристики, неописані у цьому стандарті, але теж необхідні для побудови моделі якості архітектури ПС.

### Інтерпретація стандартизованих характеристик якості програмного продукту стосовно якості архітектури

Для якості програмної системи можна записати, що

$$Q_{prod} = \Theta(Q_A) + X, \quad (3)$$

де  $\Theta$  – певний оператор, який виконує відображення усіх характеристик якості архітектури на якість програмного продукту;

$X$  – складова якості ПС, що залежить від способів реалізації окремих модулів (елементів) системи, використовуваних алгоритмів роботи тощо.

Таким чином, для інтерпретації стандартизованих характеристик якості програмного продукту стосовно його архітектури можна виконати обернену операцію, тобто:

$$Q_A = \Theta^{-1}(Q_{prod}). \quad (4)$$

В даній статті в якості цього оберненого оператора пропонується застосовувати асоціативний логічний аналіз. Подібний підхід було зроблено у [14], але стосовно інших задач програмної інженерії.

Тепер можна проаналізувати характеристики якості програмного продукту на предмет їх застосування для оцінювання якості архітектури цього продукту. Тут слід наголосити, що оцінювання якості архітектури проводиться як порівняння певного архітектурного рішення з іншими, що реалізують ту ж функціональність системи. Тому нам не потрібно оперувати абсолютними значеннями характеристик якості, а достатньо лише буде вказати, кращий чи гірший даний показник для поточного архітектурного рішення по

відношенню до такого самого показника, реалізованого іншим архітектурним дизайном.

Для характеристик якості у використанні такого аналізу виконувати не будемо на тій підставі, що вони можуть бути оцінені вже для готового програмного продукту його користувачем, про що сказано в п. 4.1 стандарту [10].

Результати аналізу викладені у таблиці 4.

Таблиця 4

**Відповідність між показниками якості програмного продукту та якості архітектури.**

Підхарактеристика якості програмного продукту згідно ISO 25010	Інтерпретація підхарактеристики стосовно якості архітектури
1	2
<b>Функціональна придатність (ФП) (Functional suitability)</b>	
"Функціональна завершеність (ФЗ)" (Functional completeness – FCom)	На рівні архітектури ФЗ може вимірюватись множиною значень "так" чи "ні" відповідно до того, реалізована певна функціональна вимога хоч у одному елементі архітектури чи не реалізована. $FCom = \frac{M}{N}$ , де $N$ – кількість функцій для реалізації, $M$ – реалізовано функцій ( $M \leq N$ )
"Функціональна коректність (ФК)" (Functional correctness – FCor)	Характеризує міру забезпечення достовірних результатів з необхідним рівнем точності. На рівні архітектури цей параметр може бути лише оцінений відповідно до шкали "так" чи "ні" для кожного елемента архітектури, що реалізує певну функцію з необхідним рівнем точності. $FCor = \frac{M}{N}$ , де $N$ – кількість функцій для реалізації, $M$ – реалізовано функцій ( $M \leq N$ )
"Функціональна відповідність (ФВ)" (Functional appropriateness – FA)	Не інтерпретується для архітектури. Пов'язана зі способом виконання операцій. Мова йде скоріш про особливості реалізації інтерфейсу користувача, ніж про архітектурні рішення.
<b>Ефективність роботи (ЕР) (Performance efficiency)</b>	
"Поведінка в часі (ПЧ)" (Time behavior – TB)	ПЧ описує ступінь відповідності часу відгуку системи чи часу обробки даних значенням, описаним у вимогах до системи. На рівні архітектури цей параметр може бути оцінений лише через відносні значення часу проходження даних через всі елементи архітектури $\sum_i E_i^{TB}$ та через зв'язки між елементами $\sum_j C_j^{TB}$ , через які необхідно пройти даним для завершення виконання певної функції: $TB = \sum_i E_i^{TB} + \sum_j C_j^{TB}$ .
"Використання ресурсів (ВР)" (Resource utilization – RU)	ВР описує кількість різномісних ресурсів обчислювальної системи, необхідних для функціонування проектованої ПС (за винятком потреби у людських ресурсах). На рівні архітектури від проектувальника вимагається знання особливостей вимог до ресурсів тих чи інших архітектур при інших рівних умовах, наприклад, вимог до об'єму та/або часу використання оперативної пам'яті для рішення на основі Java™ та для аналогічного рішення на основі .NET
"Ємність (Є)" (Capacity – C)	Означає максимальне граничне значення параметру системи. Це може бути, наприклад, кількість записів у таблиці, кількість одночасно підключених користувачів тощо. На рівні архітектури ця підхарактеристика ефективності роботи реалізується саме архітектурними елементами та застосуванням тих чи інших рішень. Може бути оцінена лише відносно аналогічного рішення у іншій архітектурі.
<b>Сумісність (Compatibility)</b>	
"Співіснування (СІ)" (Co-existence – CE)	Не інтерпретується для архітектури. СІ описує можливість функціонування проектованої ПС в одному програмному/апаратному середовищі з іншими системами без шкоди для них. На рівні архітектури цю характеристику оцінювати не потрібно, оскільки вона реалізується на етапі фізичного створення системи з врахуванням обраного середовища, де планується розгорнути систему.

1	2
"Здатність до взаємодії (ЗВ)" (Interoperability – IO)	ЗВ на рівні архітектури може бути оцінена як підхарактеристика зі значеннями "так" або "ні" і реалізуватись наявністю відповідних елементів. Наприклад, якщо ПС вимагає сумісності з CORBA, то в архітектурі мають бути наявні відповідні елементи для сумісності з цією технологією.
<b>Придатність до використання (Usability)</b>	
"Прийнятна впізнаваність(ПВ)" (Appropriateness recognizeability– AR)	Не інтерпретується для архітектури. Реалізується через інтерфейс користувача, а тому незалежна від архітектурних рішень
"Придатність до вивчення (ПН)" (Learnability – LA)	ПН реалізується через наявність (Так / ні) елементів у архітектурі системи, які дозволяють користувачеві самостійно без документації взнати про призначення та спосіб використання програми (система допомоги, підказки, тощо).
"Придатність до роботи (ПР)" (Operability – OA)	Не інтерпретується для архітектури. Реалізується через інтерфейс користувача, а тому незалежна від архітектури.
"Захист від помилок користувача (ЗПК)" (User error protection –UEP)	ЗПК на рівні архітектури реалізується шляхом введення до її складу елементів, що не дозволяють користувачеві виконати неправильні дії (ввід некоректних даних, використання хибних елементів інтерфейсу тощо). Наявність таких елементів – "так", відсутність – "ні".
<b>Надійність (Reliability)</b>	
"Зрілість (Зр)" (Maturity – Mt)	Зрілість системи означає її здатність виконувати свої функції в нормальних умовах. На рівні архітектури цю підхарактеристику трактуватимемо як сумарний вклад зрілостей всіх елементів $\sum_i E_i^{Mt}$ та зв'язків між ними $\sum_j C_j^{Mt}$ у архітектурі <sup>12</sup> : $Mt = \sum_i E_i^{Mt} + \sum_j C_j^{Mt}$
"Наявність (Нв)" (Availability – Av)	Наявність означає можливість використання системи при потребі. У стандарті [10] вказано, що дана підхарактеристика є комбінацією зрілості, відмовостійкості та відновлюваності, а тому при забезпеченні цих характеристик і сама наявність буде забезпечена.
"Відмовостійкість (ВС)" (Faulttolerance – FT)	Відмовостійкість описується як здатність системи функціонувати, незважаючи на наявність апаратних чи програмних помилок. На рівні архітектури ця підхарактеристика може бути реалізована через використання відповідних елементів для коректної обробки помилкових ситуацій чи через надлишковість у системі і характеризується за шкалою із значеннями "так" чи "ні".
"Придатність до відновлення (ПВд)" (Recoverability – RA)	Придатність до відновлення означає здатність системи відновити дані та свій стан після настання збою у роботі за встановлений період часу. На рівні системи може вимірюватись необхідним часом на відновлення свого стану. На рівні архітектури ця підхарактеристика реалізується наявністю відповідних архітектурних елементів, що забезпечують відновлення системи після збою. "Так" чи "ні" для кожного елемента архітектури, де можлива втрата даних при збоях.
<b>Безпека (Security)</b>	
"Конфіденційність (Кн)" (Confidentiality – Conf)	Міра того, що система надає доступ до даних лише тим користувачам, для яких ці дані доступні згідно авторизації. На рівні архітектури забезпечується наявністю у ній певних елементів та вимірюється згідно значень "так" та "ні".
"Цілісність (Цл)" (Integrity – Int)	Здатність системи запобігати неавторизованому доступу до програми та/або даних. На рівні архітектури забезпечується наявністю у ній певних елементів та вимірюється значеннями "так" та "ні".
"Незворотність (НЗв)" (Non-repudiation – NR)	Незворотність описує властивість системи зберігати всі зміни та події, що настали в системі без ризику втратити їх пізніше внаслідок інших дій. За змістом це нагадує принцип тривалості (durability) для транзакцій. Реалізується на етапі детального проектування системи, але на рівні архітектури може бути представлений у вигляді елементів, що забезпечують дану характеристику і вимірюється згідно значень "так" та "ні".
"Придатність до обліку (ЗО)" (Accountability – AA)	Підхарактеристика забезпечує механізм журналювання роботи системи і на рівні архітектури реалізується наявністю чи відсутністю елементів з відповідними функціями та оцінюється відповідно до значень так" чи "ні".

<sup>12</sup> Характеристики зрілості для елементів архітектури та зв'язків між ними мають бути відомі. У випадку використання патернів проектування ця умова виконується [16].

1	2
"Достовірність (Дст)" (Authenticity – Auth)	Достовірність є підхарактеристикою того, що кожен елемент чи система в цілому може бути розпізнана, що це саме вона, а не щось інше. Підхарактеристика є досить важливою для побудови систем з віддаленим доступом чи розподілених систем. На рівні архітектури може по аналогії до інших підхарактеристик безпеки представляється наявністю відповідних елементів для її забезпечення і вимірюється значеннями "так" чи "ні".
<b>Здатність до обслуговування (Maintainability)</b>	
"Модульність (Мд)" (Modularity – Md);	Спосіб побудови системи, коли зміна будь-якого її елементу вимагає мінімуму змін інших елементів. В [4] ця підхарактеристика має назву "Зчеплення". Для архітектури ця підхарактеристика може бути визначена, як кількість входів елементів та кількість виходів (fan-in та fan-out атрибути відповідно) і бути оцінена для найбільш залежного елементу архітектури згідно виразу: $Md = \max \left( fan - in \left( E_i^{Md} \right) + fan - out \left( E_i^{Md} \right) \right), \forall i = \overline{1, n}.$ Тут $E_i^{Md}$ – i-й елемент архітектури; n – кількість елементів у архітектурі.
"Повторне використання (ПВк)" (Reusability – RU);	Не інтерпретується для архітектури. Повторне використання означає придатність системи чи її компоненту до застосування в інших системах. Оскільки архітектуру програмного продукту пропонується компонувати з патернів проектування, то ця концепція закладена в сам процес розробки архітектури, тому розглядати його для оцінки архітектури не будемо.
"Придатність до аналізу (ПрА)" (Analysability – AnA);	Придатність до аналізу означає ефективність процесу оцінювання того, як вплине на систему зміна певних її елементів, чи діагностування причин збоїв у роботі або можливість ідентифікації частин, які потребують заміни. Згідно стандарту [10] ця підхарактеристика визначається на етапі тестування та супроводу системи, а на рівні архітектури вона може бути оцінена опосередковано через складність, тобто кількість елементів та зв'язків між ними.
"Придатність до модифікації (ПрМ)" (Modifyability – MD);	Придатність до модифікації означає можливість зміни у системі її елементів без зниження загального показника якості. Може на рівні архітектури визначатись при порівнянні різних архітектурних рішень для реалізації однієї і тієї ж архітектури.
"Придатність до тестування (ПрТ)" (Testability – TA)	Придатність до тестування означає можливість перевірки значень певних параметрів системи, чи відповідають вони вимогам. Для архітектури ці підхарактеристики, як і придатність до аналізу, також можуть бути оцінені через складність системи.
<b>Переносимість" (Portability)</b>	
"Адаптованість (Ад)" (Adaptability – Ad)	Адаптованість для системи означає продуктивність та ефективність процесу пристосування системи чи її елементів до програмно-апаратного середовища. На рівні архітектури може бути представлена наявністю відповідних елементів, що реалізують таку можливість і оцінюватись згідно значень "так" або "ні".
"Здатність до встановлення – ЗВс" (Installability – IA)	Здатність до встановлення означає ефективність процесів установки/видалення програмного продукту в системі і на рівні архітектури може бути реалізований у вигляді відповідних елементів, що полегшують виконання цих процедур. Оцінювання – "так", "ні".
"Здатність до заміни – ЗЗ" (Replaceability – RpA)	Для того, щоби система відповідала вимогам цієї підхарактеристики, має бути змога замінити існуючу системою іншою аналогічного призначення. У стандарті [10] сказано, що ця підхарактеристика містить атрибути як адаптованості, так і здатності до встановлення, але виділена окремо через свою важливість (відображає можливість оновлення продукту до нових версій та знижує ризик замкнутості системи через використання, наприклад, стандартних форматів файлів). Через це оцінювання адаптованості та здатності до встановлення автоматично означає оцінювання здатності до заміни.

### Нестандартизовані характеристики якості для моделі якості архітектури

Серед характеристик якості власне архітектури, а не програмної системи, з аналізу літературних джерел можна віднести зв'язність і зчеплення [4] та концептуальна цілісність [17, 2].

Характеристики зв'язності та зчеплення характеризують здатність архітектури до внесення в неї змін, що є досить частою операцією внаслідок змін вимог до проектованої системи. Ці характеристики застосовуються до архітектури, представленої у вигляді декомпозиції задачі, розв'язок котрої реалізує система, на окремі взаємопов'язані модулі [4]. Висока зв'язність всередині кожного модуля разом із

низьким зчепленням між модулями означає хороше архітектурне рішення. Цими характеристиками можна користуватись на початковому етапі розробки архітектури, коли приймається рішення про загальну архітектурну модель (клієнт/сервер, паралельні процеси, багат шарова модель тощо). Кожна з таких архітектурних моделей краще підходить для різних предметних областей, і, відповідно, показники зчеплення і зв'язності для кожної предметної області будуть різними для однієї і тієї ж моделі. Оскільки при проектуванні архітектури стоїть задача вибрати одне з декількох рішень, то значення цих показників можуть бути оцінені одне відносно одного для різних архітектурних моделей без прив'язки до абсолютних значень.

Іншим важливим показником якості архітектури ПС, в першу чергу у [17], атакою[2, 3], згадується концептуальна цілісність. Поняття концептуальної цілісності для архітектури може бути сформульоване, як єдина система підходів до конструкторських рішень, узгоджених по визначальному набору показників якості. Цей набір служить для оцінки рівня концептуальної цілісності, оскільки сам по собі цей показник є комплексним і загальним, а, отже, не піддається легкому оцінюванню напряму.

В [2] вказано, що визначальний набір показників якості для концептуальної цілісності становлять наступні характеристики: здатність до змін, продуктивність, ємність, екологічність, модульність, придатність до побудови, технологічність і безпечність. Кожен з цих визначальних показників характеризується своїм внеском у концептуальну цілісність, тобто коефіцієнтом важливості у реалізації концептуальності. Ці коефіцієнти можуть бути визначені експертним шляхом та будуть незмінними для всіх альтернативних архітектурних рішень проектованої ПС. В якості експертного методу може бути використаний алгоритм простого вибору, описаний у [18].

В результаті концептуальна цілісність може бути оцінена, як сума добутків згаданих вище коефіцієнтів важливості характеристик у реалізації концептуальної цілісності та ступеня реалізації цієї характеристики даним архітектурним рішенням:

$$CI^k = \sum_i p_i \cdot c_i^k, \tag{5}$$

де  $CI^k$  – показник концептуальної цілісності для  $k$ -го архітектурного рішення;

$p_i$  – коефіцієнт важливості  $i$ -го показника у реалізації концептуальності;

$c_i^k$  – ступінь реалізації показника у  $k$ -му архітектурному рішенні.

Максимальне значення оцінки  $CI$  відповідає найкращому архітектурному рішенням з точки зору концептуальної цілісності. Ступінь реалізації показників концептуальної цілісності  $c_i^k$  для кожного архітектурного рішення встановлюється експертним шляхом. Для забезпечення коректності обчислень як  $p_i$ , так і  $c_i^k$  мають бути нормовані до певного значення, найчастіше до одиниці.

Отже, маємо в результаті загальну модель якості архітектури ПС, яка містить більшість характеристик і підхарактеристик якості програмного продукту, описаних у ISO 25010, а також характеристики якості архітектури, які у цьому стандарті не описані, але є важливими при порівняльному оцінюванні архітектурних рішень для проектованої ПС.

**Отримання показників якості для проектованої програмної системи**

В роботах [19,20] питання вибору характеристик якості архітектури розглядалось стосовно предметної області web-застосувань. У цих роботах характеристики якості архітектури вибрано на основі аналізу предметної області з характеристик стандарту ISO-9126. Потім їх було розбито на категорії, а кожна характеристика описувалась атрибутами, перелік яких визначається предметною областю.

У цій статті пропонується загальна методика отримання характеристик якості архітектури на основі загальної моделі її якості, оскільки для різних предметних областей різні характеристики якості мають різні значення важливості та різний вплив на реалізацію характеристик якості ПС. Як згадувалось вище у цій статті, реалізовані у архітектурі показники її якості визначають якість програмної системи загалом. Тому потрібно мати технологію комунікації характеристик якості архітектури та характеристик якості програмної системи.

В якості такої технології пропонується застосовувати метод на основі алгоритму парних порівнянь (простого алгоритму вибору) [18] та методу QFD.

Метод QFD передбачає побудову так званого "будинку якості", який умовно показаний на рис. 1. Тут справа записуються характеристики якості програмної системи  $H_j^{ПС}$ , а зверху – характеристики якості архітектури  $H_i^A$ . Через використання простого алгоритму вибору для кожної характеристики якості програмного продукту  $H_j^{ПС}$  визначаються її пріоритети  $p_j^{ПС}$ .

Далі у клітинках таблиці експерти розставляють значення, які відображають ступінь впливу кожної характеристики якості архітектури на кожну характеристику

	...	$H_i^A$	...	
		⋮		
$H_j^{ПС}$	...	$a_{ij}$	...	$p_j^{ПС}$
		⋮		
		$w_i^A$		

Рис. 1. "Дім якості" для вибору характеристик якості архітектури

якості ПС  $a_{ij}$ . Останнім кроком буде обрахунок коефіцієнту важливості (ваги) кожної характеристики



якості архітектури для даної предметної області згідно формули:

$$w_i^A = \sum_j a_{ij} \cdot p_j^{PC} \quad (6)$$

Якщо наступним кроком задати нижню порогову межу  $w_{пор.}$  для розрахованих значень ваг характеристик якості архітектури, то можна таким чином "відсікти" малозначимі характеристики:

$$\{w_i^s\} \in \{w_i^{PC}\} > w_{пор.} \quad (7)$$

де  $\{w_i^s\}$  – множина критеріїв якості архітектури, на основі яких проводитиметься її порівняльне оцінювання.

Цей крок на наступних стадіях проектування програмної системи дозволить значно скоротити часові, людські та матеріальні ресурси, оскільки не доведеться затрачати сили на реалізацію всіх характеристик якості архітектури, а лише на найбільш значимі в контексті розроблюваної програмної системи.

Встановлена таким чином множина критеріїв якості архітектури дасть можливість отримати інтегральну оцінку кожної з альтернативних архітектур на основі обрахованих ваг критеріїв якості та знаходження найкращого варіанту архітектурного рішення.

### Висновок

На основі аналізу основних підходів до проектування архітектури ПС з врахуванням показників якості, показано, що в більшості з них використовують неуніфіковані і нестандартизовані показники. Причому в основному це показники якості ПС загалом, а не архітектури. Часто ці показники відносяться до різних етапів життєвого циклу проектованої системи.

Залежність між цими групами показників досліджується методами логічного аналізу, а не формальними методами. Це приводить до суб'єктивних факторів та неоднозначності при тлумаченні показників, нечіткості, підміні понять. Тому запропоновано використовувати стандарт ISO 25010 як базовий для формулювання показників якості архітектури. Але оскільки цей стандарт описує якість програмного продукту, а не його архітектури, то для оцінювання нестандартизованих показників якості архітектури пропонується виражати їх в термінах стандарту. Крім того, модель якості архітектури доповнена такими характеристиками, як зв'язність та зчеплення, а також концептуальна цілісність та запропоновано методику їх оцінювання для вибору архітектури ПС на основі показників якості.

Оскільки для оцінювання якості архітектури може бути запропоновано значну кількість показників, то пропонується обмежити цю множину через експертне оцінювання важливості кожного критерію якості архітектури відносно реалізації ним певного атрибуту якості ПС з наступним вибиранням тих критеріїв, для котрих обраховані вагові коефіцієнти перевищують встановлене порогове значення. В якості експертної технології пропонується застосовувати метод парних порівнянь та метод QFD.

### Література

1. Fowler, Martin. Patterns of Enterprise Application Architecture [Text] / Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford.: Addison-Wesley, 2002. – 533 p.
2. Спинеллис, Диомидис. Идеальная архитектура. Ведущие специалисты о красоте программных архитектур [Текст]. / Диомидис Спинеллис, Георгиос Гусиос. Санкт-Петербург – Москва: Издательство "Символ+", 2010. – 528 с.: ил.
3. Bass, L. Software architecture in practice : 2<sup>nd</sup> edition [Text] / Len Bass, Paul Clements, Rick Kazman. – Boston, MA: Addison-Wesley Professional, 2003. – 528 p.
4. Брауде Э. Технология разработки программного обеспечения [Текст] / Эрик Брауде. – СПб.: Питер, 2004. – 655 с.: ил.
5. ISO/IEC/IEEE 42010:2011 Systems and software engineering – Architecture description.
6. ISO/IEC 12207:2008. Systems and software engineering – Software lifecycle processes. – 123 p.
7. ДСТУ ISO 9001 – 2001. Системи управління якістю. Вимоги. – Чинний від 27.06.2001. – К. : Держстандарт України, 2001. – 23 с.
8. Коваль Г. Моделирование вимог до якості програмних систем оброблення даних. [Текст] / Г. Коваль, Г. Мороз // Проблеми програмування. – № 2-3. – 2006. – С. 237 – 244.
9. Райчев І. Інструментальний засіб розробки та комунікації вимог якості до програмних систем [Текст] / І. Райчев, О. Харченко. // Збірник наукових праць ІПМЕ НАНУ. – 2007. – Вип. 39. – С. 184-193.
10. ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.
11. ISO/IEC TR 25021:2007 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality measure elements.
12. ISO/IEC CD 25023 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality.
13. 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.

14. Волкова С.О. Формалізація характеристик якості програмних продуктів та їх узгодження зі специфікаціями тестування для систем критичного та комерційного застосування [Текст] / С.О. Волкова, О.В. Гнездьонова // Вісник Херсонського національного технічного університету. – Том 1. – 2008. – С. 202-208.
15. Руководство Microsoft по проектированию архитектуры приложений (2-е издание) [Текст] / – Корпорация Майкрософт, 2009. – 528 с.
16. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст] / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – М., СПб.: Питер, 2010. – 366 с.
17. Frederick P. Brooks, Jr. The Mythical Man-Month. Essays on Software Engineering. Anniversary Edition [Text] / Frederick P. Brooks, Jr. ADDISON-WESLEY, 1995. – 322 p.
18. Черноуцкий И.Г. Методы принятия решений [Текст] / Черноуцкий И.Г. – СПб. : БХВ-Петербург, 2005. – 416 с.: ил.
19. Харченко О.Г. Инструментальний засіб розробки та комунікації вимог якості до програмних систем / О.Г. Харченко, В.В. Яцишин, І.Е. Райчев // Інженерія програмного забезпечення. – № 2. – 2010. – С. 30-36.
20. Харченко О.Г. Проектування архітектури web-застосувань на основі моделі якості / О.Г. Харченко, І.О. Галай, І.О. Боднарчук, В.В. Яцишин // Інженерія програмного забезпечення. – № 4. – 2010. – С. 26-34.

#### References

1. Fowler, Martin. PatternsofEnterpriseApplicationArchitecture [Text] / MartinFowler, DavidRice, MatthewFoemmel, EdwardHieatt, RobertMee, RandyStafford.: Addison-Wesley, 2002. – 533 p.
2. Spynellys, Dyomydys. Ydeal'nayaarkhitektura. Vedushchie spetsyalisty o krasoteprogramnyh arhitektur [Tekst]. / DyomydysSpynellys, NeorhyosHusyos. Sankt-Peterburh – Moskva : Izdatel'stvo "Simvol+", 2010. – 528 p.: yl.
3. Bass, L. Softwarearchitectureinpractice : 2nd edition [Text] / LenBass, PaulClements, RickKazman. – Boston, MA : Addison-Wesley Professional, 2003. – 528 p.
4. BraudeE. Tehnolohiyarazrabotkyprohrammnohoobespecheniya [Tekst] / Erik Braude. – SPb.: Piter, 2004. – 655 p.: yl.
5. ISO/IEC/IEEE 42010:2011 Systems andsoftwareengineering – Architecturedescription.
6. ISO/IEC 12207:2008. Systemsandsoftwareengineering – Softwarelifecycleprocesses. – 123 p.
7. DSTU ISO 9001 – 2001. Systemyupravlinnyayakisty. Vymohy. – Chynnyyvid 27.06.2001. – K. : DerzhstandartUkrayiny, 2001. – 23 p.
8. Koval' H. Modelyuvannyavymohdoyakostiprohramnykhsystemobroblennyadanykh. [Tekst] / H. Koval', H. Moroz // Problemyprohramuvannya. # 2 – 3, 2006. – p. 237 – 244.
9. Raychev I. Instrumental'nyyazasibrozkbytakomunikatsiyivymohyakostidoprohramnykhsystem [Tekst] / I. Raychev, O. Kharchenko. // Zbirnyknaukovykhprats' IPME NANU. – 2007. – vyp. 39. – p. 184 – 193.
10. ISO/IEC 25010:2011 Systemsandsoftwareengineering – SystemsandsoftwareQualityRequirementsandEvaluation (SQuaRE) – Systemandsoftwarequalitymodels.
11. ISO/IEC TR 25021:2007 Softwareengineering – SoftwareproductQualityRequirementsandEvaluation (SQuaRE) – Qualitymeasureelements.
12. ISO/IEC CD 25023 Systemsandsoftwareengineering – SystemsandsoftwareQualityRequirementsandEvaluation (SQuaRE) – Measurementofsystemandsoftwareproductquality.
13. ISO/IEC 9126 (1 – 4) Softwareengineering – Productquality – Part 1: Qualitymodel, Part 2: Externalmetrics, Part 3: Internalmetrics, Part 4: Qualityinusemetrics, 2001 – 2004.
14. S.O. Volkova. Formalizatsiyakharakterystykyakostiprohramnykhproduktivtayikhuz:hodzhennyazipsyfykatsiyamytstuvannyadlyasystemkrytchnohotakomert siynohozastosuvannya [Tekst] / p.O. Volkova, O.V. Hnezd'onova // VisnykKhersons'kohonatsional'nohotekhnichnohouniversytetu, tom 1, 2008, p. 202 – 208.
15. Rukovodstvo Microsoft poproektirovaniyarhitekturyprilozheniy (2-eizdanie) [Tekst] / – Korporatsyya Microsoft, 2009. – 528 p.
16. HammaE. Priemyobyektno-oryentirovannohoproektirovaniya. Patternyproektirovaniya [Tekst] / E. Hamma, R. Khelm, R. Dzhonson, Dzh. Vlyssydes. – M., SPb. : Piter, 2010. – 366 p.
17. Frederick P. Brooks, Jr. TheMythicalMan-Month. EssaysonSoftwareEngineering. AnniversaryEdition [Text] / Frederick P. Brooks, Jr. ADDISON-WESLEY, 1995. – 322 p.
18. Chernorutskyy Y.H. Metodyprinyatiyaresheniy [Tekst] / Chernorutskiy Y.H. – SPb. : BKhV-Peterburh, 2005. – 416 p.: yl.
19. Kharchenko O.H. Instrumental'nyyazasibrozkbytakomunikatsiyivymohyakostidoprohramnykhsystem / O.H. Kharchenko, V.V. Yatsyshyn, I.E. Raychev // Inzheneriyaprohramnohozabezpechennya. # 2, 2010, p. 30 – 36.
20. Kharchenko O.H. Proektuvannyaarkhitekturyweb-zastosuvan' naosnovimodeliyakosti / O.H. Kharchenko, I.O. Halay, I.O. Bodnarchuk, V.V. Yatsyshyn // Inzheneriyaprohramnohozabezpechennya. # 4, 2010, p. 26 – 34.

Рецензія/Peer review : 2.7.2013 p. Надрукована/Printed : 17.10.2013 p.

Рецензент: