

М.В. ЄВЛАНОВ

УДОСКОНАЛЕНИЙ МЕТОД СИНТЕЗУ ВАРІАНТІВ ОПИСУ АРХІТЕКТУРИ СТВОРЮВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Проаналізовано існуючі підходи до аналізу вимог до інформаційної системи і виявлення дубльованих вимог. Пропонується удосконалення методу синтезу варіантів опису архітектури створюваної інформаційної системи. Наведені результати апробації дозволяють стверджувати, що запропоноване удосконалення скорочує обчислювальні витрати на реалізацію методу.

1. Постановка проблеми виявлення дубльованих функціональних вимог до створюваної інформаційної системи

В даний час аналіз вимог до створюваної інформаційної системи (ІС) є одним з типових процесів життєвого циклу системи [1]. В рамках даного процесу рекомендується піддавати аналізу повну сукупність виявлених вимог. При цьому в аналіз вимог рекомендується включати ідентифікацію суперечливих, пропущених, неповних, неоднозначних, нелогічних вимог або вимог, що не перевіряються, а також розстановку пріоритетів виконання цих вимог. В ході аналізу також слід вирішувати проблеми, що виникають у зв'язку з визначенням вимог. Ці проблеми пов'язані з вимогами, які не можуть бути реалізовані або які недоцільно реалізовувати [1].

Під вимогами, які недоцільно реалізовувати, в першу чергу, слід розуміти вимоги, що дублюють одна одну. Виявлення таких вимог набуває особливого значення при використанні сервісного підходу до створення ІС [2]. Відповідно до даного підходу, кожна функціональна вимога описує конкретну ІТ-послугу. Тому дублювання функціональних вимог призводить до необхідності витратити час і ресурси на дослідження і розробку однієї і тієї ж ІТ-послуги як множини різних ІТ-послуг. Наслідком цього є необхідність перепроектування ІС з метою виправлення допущених помилок і зниження ефекту від експлуатації ІС через дублювання ІТ-послуг.

Однак рекомендації з аналізу вимог до системи в переважній більшості випадків носять загальний характер [1]. У більшості сучасних засобів формування і управління вимогами функції аналізу вимог не реалізовані [3, 4]. Тому проблема розробки і удосконалення методів виявлення вимог до ІС, які дублюють один одного, є однією з найважливіших проблем, що виникають в ході формування та аналізу вимог до ІС, а також в ході проектування архітектури створюваної ІС.

2. Аналіз сучасних досліджень методів і способів аналізу вимог до інформаційної системи

Більшість існуючих методів і способів, запропонованих до теперішнього часу для аналізу вимог, описані в [5]. Однак ці методи, як показує практика, не дозволяють виявляти і усувати більшість помилок, пов'язаних з формуванням вимог до ІС. В даний час безпосередній вплив помилкового визначення вимог до ІС на результативність надання ІС і її оновлень Споживачеві ІТ-послуг вважається визнаним фактом [6]. Тому сучасні дослідження в галузі інженерії вимог спрямовані, в тому числі, на пошук методів і способів, що дозволяють усунути подібний вплив.

Найбільш перспективним напрямком даних досліджень є створення і розвиток формальних моделей і методів аналізу вимог до системи. Наприклад, в [7] процес встановлення вимог до даних визначається як система управління зі зворотним зв'язком з безперервною оптимізацією моделей поведінки користувачів. Таке представлення дозволяє застосувати для опису процесів формування та аналізу вимог положення сучасної кібернетики. В [8] розглядаються можливості опису і моделювання поведінки користувачів створюваної системи з застосуванням апарату теорії категорій, на основі якого за допомогою графічних методів створено спеціальну декларативну мову. В [9] запропонований алгебраїчний підхід до аналізу імовірнісних моделей поведінки програмного забезпечення. Але найперспективні-

ішим підходом до формального опису вимог, а також моделей і методів їх аналізу є використання онтологій, знання-орієнтованих моделей і методів. Огляд застосування онтологій в інженерії вимог наведено в [10]. Зокрема, там зазначається наступне [10]:

- а) існують емпіричні докази переваг використання онтологій в інженерії вимог спеціально для зменшення двозначності, неузгодженості та неповноти вимог;
- б) процес інженерії вимог в більшості досліджень розглядається тільки частково;
- в) єдиного стилю моделювання процесів інженерії вимог, заснованого на онтологіях, в даний час не існує;
- г) більшість досліджень в даній області пов'язано тільки з функціональними вимогами;
- д) жодна з онтологій інженерії вимог в даний час не має широкого поширення у спільноті фахівців в даній області.

Аналогічні висновки наведені і в роботі [11], присвяченій дослідженням питань видобутку та управління знаннями для формування специфікацій вимог.

Не менша увага приділяється і дослідженням прикладних аспектів застосування онтологій в інженерії вимог. Так робота [12] присвячена застосуванню заснованого на онтологіях інструментального засобу для автоматичного аналізу послідовності, правильності та повноти вимог до ІТ-продукту (проблеми 3Cs). В [13] розглянуті окремі аспекти аналізу вимог до мультиагентних систем.

Результати проведеного аналізу дозволяють зробити висновок про те, що дослідження в галузі розробки моделей і методів аналізу вимог, заснованих на знання-орієнтованих моделях вимог до ІС, є найбільш перспективними. Однак конкретні результати, що дозволяють успішно реалізувати методи аналізу вимог у вигляді функцій інструментальних засобів формування і управління вимогами до ІС, поки ще не досягнуті.

3. Постановка задачі удосконалення методу синтезу варіантів описів архітектури створюваної інформаційної системи

Головною особливістю формального опису задачі виявлення дублюючих одна одну вимог до ІС є представлення описів цих вимог не в числовому, а в категорійному вигляді. Через це застосування традиційних методів, які розглядають вирішення цієї задачі як окремих випадків вирішення задачі кластеризації описів вимог, стає надзвичайно складним. З метою подолання цих труднощів в даний час застосовуються два основні підходи:

- а) перетворення категорійних описів вимог до ІС в числові;
- б) розробка методів вирішення задачі кластеризації, заснованих на порівнянні окремих елементів категорійних описів об'єктів аналізу.

Перший підхід передбачає опис кожного окремого атрибута і кожної окремої сутності, присутніх в описах функціональних вимог до ІС, числовими ідентифікаторами. Однак такий підхід вимагає значного збільшення витрат пам'яті в ході використання методу пошуку дублюючих одна одну вимог до середніх і великих ІС.

Методи вирішення задачі кластеризації, що реалізуються в рамках другого підходу, припускають попарне порівняння двох категорійних описів об'єктів аналізу з виділенням однакових описів окремих елементів таких описів. При цьому в загальний кластер поміщаються категорійні описи, для яких кількість дублюючих один одного елементів, а також кількість випадків дублювання елементами один одного прагне до максимуму. Прикладом реалізації такого підходу є розглянутий в [14] алгоритм вирішення задачі кластеризації категорійних даних CLOPE. Даний алгоритм був покладений автором в основу розглянутого в [15, 16] методу синтезу варіантів описів архітектури створюваної ІС, який складається з наступних етапів.

Етап 1. Сформувані вихідний варіант опису архітектури створюваної ІС $Arch_{base}$.

Крок 1.1. Визначити кількість загальносистемних представлень функціональних вимог до ІС на рівні знань $n = |K_i^{fIS}|$.

Крок 1.2. Сформувані множину описів ІТ-послуг IT_{acm} шляхом виконання операції $IT_{acm_j} = K_i^{fIS}$.

Крок 1.3. Для множини $\{K_i^{fIS}\}$ виконати операцію $Arch_{base} = \bigcup_{i=c+1}^e K_i^{fIS}$.

Етап 2. Встановити значення коефіцієнта відштовхування r і розрахувати значення

$$\text{Pr ofit}(IT_{acm}, r) = \sum_{j=1}^k \frac{S(IT_{acm_j})}{W(IT_{acm_j})^r} \times |IT_{acm_j}| \bigg/ \sum_{j=1}^k |IT_{acm_j}|.$$

функції виграшу

Етап 3. Провести синтез оптимальних і/або прийнятних варіантів опису архітектури створюваної ІС.

Крок 3.1. Прийняти $\text{Pr ofit}_{max} = \text{Pr ofit}(IT_{acm}, r)$, $i = 1$, $j = 1$, $k = |IT_{acm}|$.

Крок 3.2. Вибрати представлення K_i^{fIS} .

Крок 3.3. Якщо $K_i^{fIS} \in IT_{acm_j}$, то виключити K_i^{fIS} з IT_{acm_j} . В іншому випадку перейти до кроку 3.13.

Крок 3.4. Прийняти $z = 0$, $m = 0$.

Крок 3.5. Якщо $z = j$, то прийняти $z = z + 1$.

Крок 3.6. Якщо $z > k$, то перейти до кроку 3.12.

Крок 3.7. Включити K_i^{fIS} в IT_{acm_z} .

$$\text{Pr ofit}(IT_{acm}, r) = \sum_{j=1}^k \frac{S(IT_{acm_j})}{W(IT_{acm_j})^r} \times |IT_{acm_j}| \bigg/ \sum_{j=1}^k |IT_{acm_j}|.$$

Крок 3.8. Розрахувати значення

Крок 3.9. Якщо $\text{Pr ofit}(IT_{acm}, r) > \text{Pr ofit}_{max}$, то зафіксувати варіант опису архітектури створюваної ІС як множину IT_{acm} , скориговану з урахуванням виконання Кроку 3.3 і Кроку 3.7, прийняти $m = 1$ і перейти до кроку 3.12.

Крок 3.10. Якщо $\text{Pr ofit}(IT_{acm}, r) \in [\text{Pr ofit}_{max} - \varepsilon; \text{Pr ofit}_{max}]$, то зафіксувати варіант опису архітектури створюваної ІС як множину IT_{acm} , скориговану з урахуванням виконання Кроку 3.3 і Кроку 3.7.

Крок 3.11. Прийняти $z = z + 1$. Якщо $z \leq k$, то перейти до кроку 3.5.

Крок 3.12. Якщо $IT_{acm_j} = \emptyset$ і $m = 1$, то виключити IT_{acm_j} з множини IT_{acm} і прийняти $k = k - 1$.

Крок 3.13. Прийняти $j = j + 1$. Якщо $j \leq k$, то перейти до кроку 3.3.

Крок 3.14. Прийняти $i = i + 1$. Якщо $i \leq n$, то перейти до кроку 3.2.

Крок 3.15. Якщо $m = 1$, то прийняти $i = 1$ і $j = 1$, після чого перейти до Кроку 3.2. В іншому випадку завершити виконання етапу методу.

Етап 4. Виключити з розгляду всі зафіксовані на Етапі 3 варіанти опису архітектури створюваної ІС IT_{acm} , для яких не виконується умова $\text{Pr ofit}(IT_{acm}, r) \in [\text{Pr ofit}_{max} - \varepsilon; \text{Pr ofit}_{max}]$. Завершити застосування методу.

В даному методі ε позначає величину допустимої похибки. У загальному випадку рекомендується вважати $\varepsilon = 0,1 \times \text{Pr ofit}_{max}$ [15, 16].

Головним недоліком даного методу, з точки зору його реалізації, є необхідність розгляду тих варіантів об'єднання загальносистемних представлень функціональних вимог до ІС на рівні знань, які вже розглядалися раніше. Це призводить до збільшення кількості ітерацій виконання Етапу 3 даного методу і, отже, до збільшення обчислювальних витрат на реалізацію методу.

Тому метою даного дослідження є удосконалення методу синтезу варіантів описів архітектури створюваної ІС, спрямоване на скорочення кількості ітерацій пошуку дублюючих одне одного загальносистемних представлень функціональних вимог до ІС на рівні знань K_i^{fIS} . Для досягнення даної мети пропонується вирішити такі задачі дослідження:

- розробка варіанту виконання етапів методу синтезу варіантів описів архітектури створюваної ІС, що дозволяє скоротити кількість ітерацій при його використанні;
- практична апробація результатів дослідження.

4. Виклад основного матеріалу дослідження

Основним джерелом великої кількості ітерацій пошуку дублюючих одне одного загально-системних представлень функціональних вимог до ІС на рівні знань K_i^{fIS} є необхідність попарного порівняння кожного представлення з усіма іншими представленнями. Це призводить до того, що в ході виконання Кроків 3.2 - 3.7 доводиться виконувати операції вилучення з кластера Π_{acm_j} представлення K_i^{fIS} і включення цього представлення в кластер Π_{acm_z} навіть в тому випадку, якщо подібне поєднання представлень розглядалося раніше. Наприклад, якщо в ході першої ітерації виконання Етапу 3 аналізувалося включення K_1^{fIS} , що належить кластеру Π_{acm_1} , в кластер Π_{acm_2} шляхом об'єднання з K_2^{fIS} , то в ході другої ітерації методу (якщо не спрацює умова зупинки виконання Етапу 3) слід проаналізувати включення представлення K_2^{fIS} , що належить кластеру Π_{acm_2} , в кластер Π_{acm_1} шляхом об'єднання з K_1^{fIS} . Таке повторне включення не призведе до збільшення значення функції $Pr\ ofit(\Pi_{acm}, r)$, а виконання необхідних для такого включення операцій призводить до необхідності здійснення в загальному випадку $n(n-1)$ ітерацій виконання Етапу 3.

Для виключення необхідності виконання Кроків 3.2-3.7, які не поліпшують значення функції $Pr\ ofit(\Pi_{acm}, r)$, пропонується формально описати варіант архітектури створюваної ІС матрицею вигляду

$$Arch = \begin{bmatrix} t_{11}(\Pi_{acm_1}) & \dots & t_{1j}(\Pi_{acm_1}) & \dots & t_{1n}(\Pi_{acm_1}) \\ \dots & \dots & \dots & \dots & \dots \\ t_{i1}(\Pi_{acm_i}) & \dots & t_{ij}(\Pi_{acm_i}) & \dots & t_{in}(\Pi_{acm_i}) \\ \dots & \dots & \dots & \dots & \dots \\ t_{n1}(\Pi_{acm_n}) & \dots & t_{nj}(\Pi_{acm_n}) & \dots & t_{nn}(\Pi_{acm_n}) \end{bmatrix}, \quad (1)$$

де $t_{ij}(\Pi_{acm_i})$ - кількість входжень загальносистемних представлень функціональних вимог до ІС на рівні знань в опис ІТ-послуги (функції) створюваної ІС.

Значення $t_{ij}(\Pi_{acm_i})$ в ході формування матриці $Arch$ визначаються наступним чином:

а) $t_{ij}(\Pi_{acm_i}) = 1$, якщо для пропонованого варіанту включення $\{K_i^{fIS}\}$ в кластер Π_{acm_i} значення $Pr\ ofit(\Pi_{acm}, r)$ не розраховане;

б) якщо $t_{ij}(\Pi_{acm_i}) = 1$, то $t_{ji}(\Pi_{acm_j}) = 0$.

Тоді Етап 1 методу синтезу варіантів описів архітектури створюваної ІС матиме такий вигляд.

Етап 1. Сформувані вихідний варіант опису архітектури створюваної ІС $Arch_{base}$.

Крок 1.1. Визначити кількість загальносистемних представлень функціональних вимог до ІС на рівні знань $n = |K_i^{fIS}|$.

Крок 1.2. Сформувані множини описів ІТ-послуг Π_{acm} шляхом виконання операції $\Pi_{acm_j} = K_i^{fIS}$.

Крок 1.3. Для множини $\{K_i^{fIS}\}$ сформувані матрицю $Arch_{base}$ вигляду

$$\text{Arch}_{\text{base}} = \begin{bmatrix} t_{11}(\text{IT}_{\text{acm}_1}) = 1 & \dots & t_{1j}(\text{IT}_{\text{acm}_1}) = 1 & \dots & t_{1n}(\text{IT}_{\text{acm}_1}) = 1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & t_{ij}(\text{IT}_{\text{acm}_i}) = 1 & \dots & t_{in}(\text{IT}_{\text{acm}_i}) = 1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \dots & t_{nn}(\text{IT}_{\text{acm}_n}) = 1 \end{bmatrix}. \quad (2)$$

Результатом виконання Етапу 1 методу буде матриця $\text{Arch}_{\text{base}}$, що задає базовий варіант опису архітектури створюваної ІС. Базовим вважається варіант, в якому кожна сформульована функціональна вимога правовласників реалізується окремою функцією створюваної ІС [15, 16]. Пропонований формальний опис архітектури дозволяє виключити з розгляду всі повторні включення описів вимог правовласників, що зменшує вдвічі кількість ітерацій виконання Етапу 3 методу синтезу варіантів описів архітектури створюваної ІС.

Для виключення з подальшого розгляду варіантів включення загальносистемних представлень функціональних вимог на рівні знань в опису ІТ-послуг створюваної ІС, які зменшують значення функції $\text{Pr ofit}(\text{IT}_{\text{acm}}, r)$, значення $t_{ij}(\text{IT}_{\text{acm}_i})$ пропонується визначити наступним чином:

а) якщо для пропонованого варіанту включення $\{K_i^{\text{fIS}}\}$ в кластер IT_{acm_j} виконується умова $\text{Pr ofit}(\text{IT}_{\text{acm}}, r) < [\text{Pr ofit}_{\text{max}} - \varepsilon; \text{Pr ofit}_{\text{max}}]$, то прийняти $t_{ij}(\text{IT}_{\text{acm}_i}) = 0$;

б) якщо для пропонованого варіанту включення $\{K_i^{\text{fIS}}\}$ в кластер IT_{acm_j} виконується умова $\text{Pr ofit}(\text{IT}_{\text{acm}}, r) \in [\text{Pr ofit}_{\text{max}} - \varepsilon; \text{Pr ofit}_{\text{max}}]$, то прийняти $t_{ij}(\text{IT}_{\text{acm}_i}) = 1$;

в) якщо для пропонованого варіанту включення $\{K_i^{\text{fIS}}\}$ в кластер IT_{acm_j} виконується умова $\text{Pr ofit}(\text{IT}_{\text{acm}}, r) > \text{Pr ofit}_{\text{max}}$, то прийняти

$$t_{jj}(\text{IT}_{\text{acm}_j}) = t_{jj}(\text{IT}_{\text{acm}_j}) + t_{ii}(\text{IT}_{\text{acm}_i}); \quad (3)$$

$$t_{ij}(\text{IT}_{\text{acm}_i}) = 0, \quad t_{ji}(\text{IT}_{\text{acm}_j}) = 0 \quad \text{для } j = 1, \dots, n. \quad (4)$$

Тоді Етап 3 методу синтезу варіантів описів архітектури створюваної ІС матиме такий вигляд.

Етап 3. Провести синтез оптимальних і/або прийнятних варіантів опису архітектури створюваної ІС.

Крок 3.1. Прийняти $\text{Pr ofit}_{\text{max}} = \text{Pr ofit}(\text{IT}_{\text{acm}}, r)$, $i = 1$, $j = i + 1$.

Крок 3.2. Якщо $t_{ij}(\text{IT}_{\text{acm}_i}) = 1$, то виключити K_i^{fIS} з IT_{acm_i} і включити K_i^{fIS} в IT_{acm_j} . В іншому випадку перейти до кроку 3.8.

Крок 3.3. Для отриманого варіанту опису архітектури розрахувати значення $\text{Pr ofit}(\text{IT}_{\text{acm}}, r)$.

Крок 3.4. Якщо $\text{Pr ofit}(\text{IT}_{\text{acm}}, r) > \text{Pr ofit}_{\text{max}}$, то прийняти $t_{jj}(\text{IT}_{\text{acm}_j}) = t_{jj}(\text{IT}_{\text{acm}_j}) + t_{ii}(\text{IT}_{\text{acm}_i})$, $t_{ij}(\text{IT}_{\text{acm}_i}) = 0$, $t_{ji}(\text{IT}_{\text{acm}_j}) = 0$ для $j = 1, \dots, n$, $K_j^{\text{fIS}} = K_j^{\text{fIS}} \cup K_i^{\text{fIS}}$ і перейти до кроку 3.1.

Крок 3.5. Якщо $\text{Pr ofit}(\text{IT}_{\text{acm}}, r) \in [\text{Pr ofit}_{\text{max}} - \varepsilon; \text{Pr ofit}_{\text{max}}]$, то прийняти $t_{ij}(\text{IT}_{\text{acm}_i}) = 1$.

Крок 3.6. Якщо $\text{Pr ofit}(\text{IT}_{\text{acm}}, r) < [\text{Pr ofit}_{\text{max}} - \varepsilon; \text{Pr ofit}_{\text{max}}]$, то прийняти $t_{ij}(\text{IT}_{\text{acm}_i}) = 0$.

Крок 3.7. Прийняти $j = j + 1$. Якщо $j \leq n$, то перейти до кроку 3.2.

Крок 3.8. Прийняти $i = i + 1$, $j = i + 1$. Якщо $i < n$, то перейти до кроку 3.2. В іншому випадку завершити виконання етапу методу.

З урахуванням модифікації Етапу 3 Етап 4 методу синтезу варіантів описів архітектури створюваної ІС матиме такий вигляд.

Етап 4. Виключити з розгляду всі зафіксовані на Етапі 3 варіанти опису архітектури створюваної ІС IT_{acm} , для яких не виконується умова $Pr ofit(IT_{acm}, r) \in [Pr ofit_{max} - \varepsilon; Pr ofit_{max}]$.

Крок 4.1. Сформувати варіант опису архітектури створюваної ІС, включаючи в нього ті IT_{acm_j} , в яких $t_{ij}(IT_{acm_i}) \geq 1$

Крок 4.2. Для кожного $t_{ij}(IT_{acm_i}) = 1$ матриці Arch сформувати варіант опису архітектури створюваної ІС, прийнявши на час формування $K_i^{fIS} = K_i^{fIS} \cup K_j^{fIS}$, $IT_{acm_j} = \emptyset$ і $t_{jj}(IT_{acm_j}) = 0$.
Завершити застосування методу.

5. Апробація результатів удосконалення методу синтезу варіантів опису архітектури створюваної інформаційної системи

Апробацію результатів удосконалення методу синтезу варіантів опису архітектури створюваної ІС пропонується провести на прикладі функціонального модуля безпеки праці. До цього модуля правовласниками було висунуто п'ять вимог, описаних в [17].

Результат виконання Етапів 1 і 2 методу синтезу варіантів опису архітектури створюваної ІС до удосконалення наведено в табл. 1.

Таблиця 1

Номер ІТ-послуги	Загальносистемні представлення, що описують ІТ-послугу	Значення $S(IT_{acm_j})$ для ІТ-послуги	Значення $W(IT_{acm_j})$ для ІТ-послуги
1	K_1^{fIS}	7	7
2	K_2^{fIS}	12	12
3	K_3^{fIS}	6	6
4	K_4^{fIS}	8	8
5	K_5^{fIS}	28	28
Результат розрахунку функції $Pr ofit(IT_{acm}, r)$			
$ IT_{acm_j} = 5$	$r = 2$	$Pr ofit(IT_{acm}, r) = 0,110714$	

В ході виконання Етапу 3 методу синтезу варіантів опису архітектури створюваної ІС до удосконалення було зроблено 23 ітерації. При цьому на ітерації 11 було знайдено нове значення $Pr ofit_{max}$, через що виникла необхідність знову проводити попарне порівняння більшості раніше розглянутих варіантів.

Результати виконання Етапу 4 методу синтезу варіантів опису архітектури створюваної ІС до удосконалення наведені в табл. 2. Аналіз цих результатів показує, що варіанти опису архітектури, отримані в ході ітерацій 17 і 22, практично не розрізняються між собою і є дублюючими одна одну.

В ході виконання Етапу 1 і Етапу 2 вдосконаленого методу синтезу варіантів опису архітектури створюваної ІС було сформовано матричний опис базової архітектури функціонального модуля, який має вигляд

$$Arch_{base} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

Таблиця 2

Номер ІТ-послуги	Загальносистемні представлення, що описують ІТ-послугу	Значення $S(IT_{actj})$ для ІТ-послуги	Значення $W(IT_{actj})$ для ІТ-послуги
1	K_1^{fIS}	7	7
2	K_2^{fIS}	12	12
3	-	-	-
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	K_5^{fIS}	28	28
Результат розрахунку функції $Pr ofit(IT_{act}, r)$ на ітерації 11			
$ IT_{actj} = 5$	$r = 2, \varepsilon = 0,0110714$	$Pr ofit(IT_{act}, r) = 0,139881$ (зафіксований в результаті виконання Кроку 3.9, $m = 1$)	
1	K_1^{fIS}	7	7
2	-	-	-
-	-	-	-
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	$\{K_2^{fIS}, K_5^{fIS}\}$	40	31
Результат розрахунку функції $Pr ofit(IT_{act}, r)$ на ітерації 17			
$ IT_{actj} = 4$	$r = 2, \varepsilon = 0,0139881$	$Pr ofit(IT_{act}, r) = 0,132721$ (зафіксований в результаті виконання Кроку 3.10)	
1	K_1^{fIS}	7	7
2	$\{K_2^{fIS}, K_5^{fIS}\}$	40	31
-	-	-	-
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	-	-	-
Результат розрахунку функції $Pr ofit(IT_{act}, r)$ на ітерації 22			
$ IT_{actj} = 4$	$r = 2, \varepsilon = 0,0139881$	$Pr ofit(IT_{act}, r) = 0,132721$ (зафіксований в результаті виконання Кроку 3.10)	

Результат розрахунку значення функції $Pr ofit(IT_{act}, r)$ для даного варіанту архітектури збігається з результатами розрахунку, наведеними в табл. 1.

В ході виконання Етапу 3 вже на ітерації 2 був знайдений прийнятний варіант опису архітектури функціонального модуля, отриманий в результаті злиття першої і третьої функціональних вимог. Оскільки значення функції $Pr ofit(IT_{act}, r)$ не перевищувало максимальне, виконання Етапу 3 було продовжено. Матричний опис архітектури створюваного функціонального модуля після виконання Кроку 3.5 ітерації 2 має вигляд:

$$Arch = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

В ході виконання ітерації 7 Етапу 3 вдосконаленого методу був знайдений прийнятний варіант опису архітектури функціонального модуля, отриманий в результаті злиття другої і п'ятої функціональних вимог. Оскільки значення функції $Profit(IT_{acm}, r)$ не перевищувало максимальне, виконання Етапу 3 було продовжено. Матричний опис архітектури створюваного функціонального модуля після виконання Кроку 3.5 ітерації 7 має вигляд

$$Arch = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

В ході виконання ітерації 8 Етапу 3 вдосконаленого методу було знайдено нове максимальне значення функції $Profit(IT_{acm}, r)$, отримане в результаті злиття другої і п'ятої функціональних вимог. Після цього був зроблений повернення до Кроку 3.1. Матричний опис архітектури створюваного функціонального модуля після виконання Кроку 3.4 ітерації 8 має вигляд:

$$Arch = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

Для завершення Етапу 3 знадобилося ще 3 ітерації. При цьому було підтверджено знайдений раніше прийнятний варіант опису архітектури, отриманий в результаті злиття другої і п'ятої функціональних вимог [17].

В результаті виконання Етапу 4 були виділені два варіанти опису архітектури функціонального модуля. Матричний опис архітектури створюваного функціонального модуля після завершення Етапу 3 має вигляд

$$Arch = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

Результати застосування удосконаленого методу наведені в табл. 3.

Таким чином, застосування вдосконаленого методу синтезу варіантів опису архітектури створюваної ІС дозволило отримати результати, аналогічні результатам застосування невдосконаленого методу. При цьому для отримання результатів з застосуванням вдосконаленого методу потрібно в ході виконання Етапу 3 провести 11 ітерацій (замість 23 ітерацій для невдосконаленого методу). Результати удосконалення слід визнати ефективними.

6. Висновки з проведеного дослідження і перспективи подальших досліджень

Пропоноване удосконалення методу синтезу варіантів опису архітектури створюваної ІС дозволило скоротити обчислювальні витрати на виконання методу за рахунок виключення кількості ітерацій пошуку дублюючих одне одного загальносистемних представлень функціональних вимог до ІС на рівні знань K_i^{fIS} . В ході практичної апробації вдосконаленого методу кількість ітерацій виконання Етапу 3 як найскладнішого етапу методу скоротилася з 23 до 11. При цьому результати виконання удосконаленого методу відповідають результатам, отриманим в ході виконання даного методу до вдосконалення.

Таблиця 3

Номер IT-послуги	Загальносистемні представлення, що описують IT-послугу	Значення $S(IT_{actj})$ для IT-послуги	Значення $W(IT_{actj})$ для IT-послуги
1	K_1^{fIS}	7	7
2	K_2^{fIS}	12	12
3	-	-	-
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	K_5^{fIS}	28	28
Результат розрахунку функції $Profit(IT_{act}, r)$ на ітерації 8			
$ IT_{actj} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{act}, r) = 0,139881$ (зафіксований в результаті виконання Кроку 3.4)	
1	K_1^{fIS}	7	7
2	-	-	-
-	-	-	-
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	$\{K_2^{fIS}, K_5^{fIS}\}$	40	31
Результат розрахунку функції $Profit(IT_{act}, r)$ на ітерації 9			
$ IT_{actj} = 4$	$r = 2, \varepsilon = 0,0139881$	$Profit(IT_{act}, r) = 0,132721$ (зафіксований в результаті виконання Кроку 3.5)	

Однак необхідно вказати, що отримані результати справедливі тільки для проектів малих ІС і окремих функціональних модулів. Тому в перспективі для подальших досліджень пропонується вибрати роботи з перевірки ефективності та точності застосування вдосконаленого методу для ІТ-проектів середніх і великих ІС та їх функціональних модулів.

Список літератури: 1. ГОСТ ИСО/МЭК 15288-2005. Системная инженерия. Процессы жизненного цикла систем. Введ. 01-01-2007. М.: Стандартинформ, 2006. 57 с. 2. Левыкин В.М., Евланов М.В., Керносов М.А. Паттерны проектирования требований к информационной системе: моделирование и применение: монография. Харьков: ООО "Компанія СМІТ", 2014. 320 с. 3. Rational Requisite Pro / Сайт "IBM developerWorks". Режим доступа: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Wbcd69e09400c_4f72_9665_66f116225986/page/Rational%20RequisitePro. Заголовок с экрана. 4. IBM Rational DOORS Next Generation. An efficient requirements management tool for complex systems. Режим доступа: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=SP&infotype=PM&appname=SWGE_RA_IR_USEN&htmlfid=RADI4128USEN&attachment=RADI4128USEN.PDF. Заголовок с экрана. 5. Maguire M., Bevan N. User requirements analysis. A review of supporting methods // Proceedings of IFIP 17th World Computer Congress, Montreal, Canada, 25-30 August 2002. 2002. P.133-148. doi: 10.1007/978-0-387-35610-5_9. 6. Stowell F., Cooray S. The appreciative system, learning, and its impact on information systems design // Communications of the Association for Information Systems. 2017. V. 40. P. 93-119. 7. Liu L., Zhou Q., Liu J., Cao Z. Requirements cybernetics: Elicitation based on user behavioral data // Journal of Systems and Software. 2017. V. 124. P. 187-194. doi: 10.1016/j.jss.2015.12.030. 8. Asteasuain F., Braberman V. Declaratively building behavior by means of scenario clauses // Requirements Engineering. 2017. V. 22. I. 2. P. 239-274. doi: 10.1007/s00766-015-0242-2. 9. Yu Y.-J., Liu C. Little Model in Big Data: An Algebraic Approach to Analysing Abstract Software Behaviours // Ruan Jian Xue Bao/Journal of Software. 2017. V. 28 (6). P. 1488-1497. doi: 10.13328/j.cnki.jos.005229. 10. Dermeval D. etc. Applications of ontologies in requirements engineering: a systematic review of the literature // Requirements Engineering. 2016. V. 21. I. 4. P. 405-437. doi: 10.1007/s00766-015-0222-6. 11. Serna M.E., Bachiller S.O., Serna A.A. Knowledge meaning and management in requirements engineering // International Journal of Information Management. 2017. V. 37. I. 3. P. 155-161. doi:

10.1016/j.jinfomgt.2017.01.005. 12. *Nguyen T.H., Grundy J.C., Almorsy M.* Ontology-based automated support for goal-use case model analysis // *Software Quality Journal*. 2016. V. 24(3). P. 635-673. doi: 10.1007/s11219-015-9281-7. 13. *Lopez-Lorca A.A., Beydoun G., Valencia-Garcia R., Martinez-Bejar R.* Supporting agent oriented requirement analysis with ontologies // *International Journal of Human Computer Studies*. 2016. V. 87. I. C. P. 20-37. doi: 10.1016/j.ijhcs.2015.10.007. 14. *Yang Y., Guan H., You J.* CLOPE: A fast and Effective Clustering Algorithm for Transaction Data // *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York: ACM, 2002. P. 682-687. 15. *Євланов М.В., Васильцова Н.В., Панфьорова І.Ю.* Моделі і методи синтезу опису раціональної архітектури інформаційної системи // *Вісник наукового університету "Львівська політехніка"*. Серія "Інформаційні системи та мережі". 2015. № 829. С. 135-152. 16. *Євланов М.В.* Разработка модели и метода выбора описания рациональной архитектуры информационной системы // *Восточно-европейский журнал передовых технологий*. 2016. № 1/2(79). С. 4-12. doi: 10.15587/1729-4061.2016.60583. 17. *Євланов М.В., Сердюк Н.Н.* Формирование и анализ требований к информационно-аналитической системе управления безопасностью труда на предприятии // *Технологический аудит и резервы производства*. 2015. № 4/3(24). С. 41-45.

Надійшла до редколегії 10.01.2018

Євланов Максим Вікторович д-р техн. наук, доцент, професор кафедри ІУС ХНУРЕ. Наукові інтереси: проблеми інтелектуалізації проектування і експлуатації інформаційних управляючих систем. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 70-21-451.

УДК 004.053

DOI: 10.30837/0135-1710.2018.175.041

Н.В. ВАСИЛЬЦОВА, О.Є. НЕУМИВАКІНА, І.Ю. ПАНФЬОРОВА

МЕТОД ОЦІНЮВАННЯ ДОСВІДУ КОМАНДИ ВИКОНАВЦІВ ІТ-ПРОЕКТУ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Проаналізовано існуючі оцінки команд виконавців ІТ-проектів. Розглянуто загальний підхід до вирішення задачі оцінювання досвіду команд виконавців. Запропоновано кількісні показники, що характеризують досвід команди виконавців і засновані на формальних моделях опису архітектури інформаційної системи. Розроблено метод оцінювання досвіду команди виконавців ІТ-проекту створення інформаційної системи, що базується на запропонованих показниках.

1. Постановка проблеми оцінки команди виконавців ІТ-проекту створення інформаційної системи

Сучасні моделі і методи оцінювання витрат на виконання ІТ-проектів передбачають врахування характеристик команди виконавців запланованого проекту. Однак переліки врахованих характеристик в окремих моделях досить сильно розрізняються. Така розбіжність в поглядах на команду виконавців ІТ-проекту викликає сильні труднощі в ході застосування моделей і методів оцінювання витрат на виконання ІТ-проектів. Ці труднощі викликані неможливістю узгодження показників, що характеризують команду виконавців ІТ-проекту в різних моделях і методах.

Особливо сильно такі труднощі виявляються при спробі застосування параметричних моделей оцінювання витрат на виконання ІТ-проекту. Головними особливостями таких моделей є:

- а) опис ІТ-проекту, його результатів і ресурсів, використовуваних для його виконання, набором параметрів, що мають кількісні значення;
- б) використання для кількісної оцінки значення параметра моделі якісних (описових) характеристик ІТ-проекту, його результатів і ресурсів, використовуваних для його виконання.
- в) залучення, при необхідності, експертів для визначення кількісних значень параметрів моделі.

Остання особливість найсильніше позначається при урахуванні в параметричних моделях індивідуальних особливостей планованого ІТ-проекту, його очікуваних результатів і ресурсів. В даний час таке урахування здійснюється шляхом введення в параметричну