

МАТЕМАТИЧНИЙ АПАРАТ ПІДТРИМКИ ВИБОРУ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ ТА СЕРЕДОВИЩА ПРОГРАМУВАННЯ ДЛЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У даній статті розроблено математичну модель технології проектування програмного забезпечення (ТППЗ), критерії оцінювання ТППЗ, алгоритм та продукційні правила вибору технології проектування програмного забезпечення, які надають організації можливість мотивованого та обґрунтованого вибору технології проектування для її подальшого впровадження. Також у роботі розроблено математичну модель середовища програмування, критерії оцінювання середовища програмування, алгоритм та продукційні правила вибору середовища програмування, що надають підтримку вибору оптимального середовища програмування для програмного забезпечення відповідно до вимог користувача.

Ключові слова: програмне забезпечення (ПЗ), технологія проектування програмного забезпечення (ТППЗ), середовище програмування.

T.O. HOVORUSHCHENKO
Khmelnytsky National University

THE MATHEMATICAL APPARATUS TO SUPPORT OF SELECTION OF DESIGN TECHNOLOGY AND PROGRAMMING ENVIRONMENT FOR SOFTWARE

Abstract - Currently actual task at the design stage is the construction of mathematical apparatus to support of the evaluation and selection of software design technology (SDT) and actual task at the implementation stage is the construction of mathematical apparatus to support of selection of programming environment (PE) for software.

Multicriteriality of selection of software design technology and programming environment is that each existing SDT and PE should be measured not by one criterion, but by a combination of many parameters (criteria) simultaneously. In this paper the mathematical model of software design technology, criteria of software design technology evaluation, algorithm and production rules for selection of software design technology were developed. Also in this article the mathematical model of programming environment, criteria of programming environment evaluation, algorithm and production rules of selection of programming environment were developed.

The developed mathematical apparatus to support of selection of SDT provides the possibility of the reasoned and grounded selection of SDT for its further implementation. The developed mathematical apparatus to support of selection of PE supports the selection of optimal PE according to user requirements.

Keywords: software, software design technology (SDT), programming environment (PE).

Вступ

Розроблення програмного забезпечення (ПЗ) – це діяльність, яка вимагає детального вивчення предметної області та повного розуміння цілей розроблюваного продукту [1].

Факти, що визначають специфіку програмного забезпечення (ПЗ) як продукту: замовник не розуміє складності процесу розроблення ПЗ і впливу зміни вимог до ПЗ на процес розроблення; зміна та поява нових вимог в процесі розроблення ПЗ є неминучими; ітеративність процесу розроблення ПЗ, яка ускладнює його; рівень новизни та складності ПЗ є дуже високим; технології швидко змінюються, оновлюються та застарівають.

Всі специфічні факти, рівні, важливі знання, заходи, операції, прийоми щодо керування розробленням ПЗ включає в себе технологія проектування програмного забезпечення.

Технологія проектування (розроблення) програмного забезпечення (ТППЗ) – це впорядкована сукупність взаємозв'язаних технологічних процесів в межах життєвого циклу ПЗ. ТППЗ представляє собою інженерний підхід до розроблення програмних засобів, який охоплює методологію програмування, проблеми забезпечення надійності програм, оцінки робочих характеристик та якості проектів. ТППЗ визначає професійну культуру фахівців, яка забезпечує заданий рівень продуктивності праці і якості одержуваної в результаті програмної продукції [2, 3].

Основною вимогою, яка висувається до сучасних ТППЗ, є їхня відповідність стандартам та нормативним документам, пов'язаним з процесами життєвого циклу ПЗ та оцінкою технологічної зрілості організацій-розробників [4–6].

Під впровадженням ТППЗ розуміють всі дії – від оцінювання початкових потреб до повномасштабного використання ТППЗ у різних підрозділах організації [2, 3]. Процес впровадження складається з наступних етапів [2, 3]: 1) визначення потреб у ТППЗ, характеристик об'єкту впровадження та проектів створення ПЗ; 2) визначення вимог, які висуваються до ТППЗ (аналіз характеристик об'єкту впровадження та проектів, обґрунтування вимог до ТППЗ, визначення пріоритетів вимог); 3) оцінювання варіантів ТППЗ – попереднє експертне оцінювання, яке полягає у аналізі доступних ТППЗ на предмет відповідності вимогам, та деталізоване оцінювання, яке полягає у формуванні детального опису кожної ТППЗ-претендента; 4) визначення потреб у ТППЗ, характеристик об'єкту впровадження та проектів; 5) вибір ТППЗ на основі порівняльного аналізу технологій та з врахуванням експертної оцінки; 6) адаптація ТППЗ до умов застосування шляхом формування конкретної робочої конфігурації ТППЗ, адаптованої до умов об'єкту впровадження.

Сьогодні процеси оцінювання та вибору ТППЗ залишаються незабезпеченими математичним підґрунтям. Тому *актуальною задачею на етапі проектування* наразі є побудова математичного апарату для підтримки процесів оцінювання та вибору ТППЗ.

Після етапу проектування перед будь-якою компанією, що займається розробленням програмного забезпечення, постає задача вибору середовища програмування. Очевидно, що цей вибір зводиться до багатокритеріальної задачі і далеко не очевидний. Багатокритеріальність вибору середовища програмування полягає у тому, що кожне існуюче середовище програмування слід оцінювати не за одним критерієм, а за сукупністю багатьох показників (критеріїв), що розглядаються одночасно.

Безліч інформації про середовища програмування програміст може знайти в мережі Інтернет, але ця інформація не є структурованою. Надзвичайно багато часу програміст витрачає на пошук оптимального рішення щодо вибору середовища програмування, читаючи форуми та шукаючи порад більш досвідчених користувачів, але такі поради не можуть бути достатньо об'єктивними, адже, як відомо, вибір середовища програмування здебільшого є справою звички. Програмісти зазвичай використовують для розроблення ПЗ ті середовища, з якими вони вже давно працюють та до особливостей (переваг та недоліків) яких звикли. Освоєння нового середовища програмування вимагає значних витрат часу та коштів, причому ціна на сучасні засоби розроблення програмного забезпечення може сягати 1000\$ за одну ліцензію. Зрозуміло, що, якщо необхідно забезпечити середовищем програмування штат програмістів, припустимо, невеликої компанії з 10-30 працівників, то витрати стануть досить значними. Важливим є питання, чи є реальною така висока ціна для існуючих середовищ програмування. Але значно важливішим є питання, чи є реальною висока ціна для середовища програмування в контексті поставленого завдання. Можливо, для виконання поставлених перед програмістом задач буде достатньо більш дешевого або взагалі безкоштовного середовища. Отже, наразі *актуальною задачею на етапі реалізації* є побудова математичного апарату підтримки вибору середовища програмування для програмного забезпечення.

Постановка задачі. З результатів аналізу сучасного стану галузі слідує, що перспективними напрямками досліджень на етапі проектування є: 1) побудова математичної моделі технології проектування ПЗ; 2) побудова критеріїв оцінювання та продукційних правил вибору технології проектування ПЗ. А перспективними напрямками досліджень на етапі реалізації є: 1) створення математичної моделі середовища програмування; 2) побудова критеріїв оцінювання та продукційних правил вибору середовища програмування.

Математична модель технології проектування програмного забезпечення

Технологія проектування програмного забезпечення – це комплекс організаційних заходів, операцій та прийомів, спрямованих на розроблення програмних продуктів високої якості в рамках відведеного бюджету і в термін. Аналіз ТППЗ [2, 3, 7] дав можливість визначити основні складові ТППЗ: 1) технологічний процес; 2) технологічна операція; 3) робочий продукт; 4) роль; 5) керівництво; 6) інструментальний (CASE-) засіб.

Технологічна операція – основна одиниця роботи, виконувана певною роллю, яка: передбачає чітко визначену відповідальність ролі; дає чітко визначений результат (набір робочих продуктів), який базується на певних початкових даних (іншому наборі робочих продуктів); представляє собою одиницю роботи з жорстко визначеними межами, які встановлюються при плануванні проекту. Технологічний процес – це сукупність взаємозв'язаних технологічних операцій. Робочий продукт – це інформаційна або матеріальна сутність, яка створюється, модифікується або використовується в деякій технологічній операції (модель, документ, код, тест і т.і.). Робочий продукт визначає область відповідальності ролі і є об'єктом керування конфігурацією. Роль – визначення поведінки та обов'язків окремої особи або групи осіб в середовищі організації-розробника ПЗ, які здійснюють діяльність в межах деякого технологічного процесу та відповідають за певні робочі продукти. Керівництво – це практичне керівництво по виконанню однієї технологічної операції. Керівництва містять методичні матеріали, інструкції, нормативи, стандарти та критерії оцінювання якості робочих продуктів. Інструментальний засіб (CASE-засіб) – це програмний засіб, який забезпечує автоматизовану підтримку діяльності в межах технології [2, 3].

Враховуючи вищевикладене, побудуємо *математичну модель технології проектування ПЗ*:

$$TSD = \{TO, TP, WP, R, G, CT, TOR, WPTOR, GTO\}, \quad (1)$$

де $TO = \{to_1, \dots, to_f\}$ – множина технологічних операцій to_b (f – кількість технологічних операцій ТППЗ);

$TP = \{tp_1, \dots, tp_k\}$ – множина технологічних процесів tp_h (k – кількість технологічних процесів ТППЗ), кожен з яких представляє собою підмножину $\{to_1, \dots, to_m\}$ взаємозв'язаних технологічних операцій to_{ij} (m – кількість технологічних операцій, які складають даний технологічний процес), тобто $TP = \{\{to_1, \dots, to_o\}_1, \dots, \{to_1, \dots, to_e\}_k\}$, де o – кількість технологічних операцій, які складають технологічний процес tp_1 , e – кількість технологічних операцій, які складають технологічний процес tp_k ;

$WP = \{wp_1, \dots, wp_x\}$ – множина робочих продуктів wp_c (x – кількість робочих продуктів технології);

$R = \{r_1, \dots, r_{lp}\}$ – множина ролей r_y , які містить ТППЗ (lp – кількість ролей ТППЗ);

$G = \{g_1, \dots, g_f\}$ – множина практичних керівництв g_q (f – кількість технологічних операцій і відповідно керівництв технології);

$CT = \{ct_1, \dots, ct_w\}$ – множина інструментальних (CASE-) засобів ct_a (w – кількість інструментальних засобів технології проектування);

$TOR = \{<to_1, r_1>, \dots, <to_n, r_n>\}$ – множина пар to_{ji} і r_{ji} , де to_{ji} – технологічна операція, яка входить до технології проектування ПЗ і виконується роллю r_{ji} (n – кількість взаємозв'язаних операцій та ролей технології);

$WPTOR = \{<wp_1, to_1, r_1>, \dots, <wp_s, to_s, r_s>\}$ – множина трійок wp_d, to_d, r_d , де wp_d – робочий продукт, який створюється, модифікується або використовується в технологічній операції to_d і визначає область відповідальності ролі r_d (s – кількість взаємозв'язаних робочих продуктів, технологічних операцій та ролей технології);

$GTO = \{<g_1, to_1>, \dots, <g_f, to_f>\}$ – множина пар g_u, to_u , де g_u – практичне керівництво по виконанню технологічної операції to_u (f – кількість технологічних операцій і відповідно керівництв технології).

Критерії оцінювання технології проектування програмного забезпечення

Метою процесу оцінювання ТППЗ є визначення функціональності та якості ТППЗ для наступного вибору. Оцінювання виконується відповідно до конкретних критеріїв, його результати містять як об'єктивні, так і суб'єктивні дані по кожній ТППЗ. Типовий процес оцінювання може використовувати набір критеріїв різних типів. Кожний критерій повинен бути обраний та адаптований експертом з врахуванням особливостей конкретного процесу. Початковими даними для оцінювання є набір параметрів (техніко-економічних характеристик) ТППЗ [2, 3]: 1) функціональні характеристики, орієнтовані на процеси життєвого циклу ПЗ (керування проектом, керування вимогами, керування конфігурацією та змінами, аналіз та проектування ПЗ і т.і.); 2) функціональні характеристики застосування (середовище функціонування, сумісність з іншими ТППЗ, відповідність технологічним стандартам); 3) характеристики якості (надійність, зручність використання, ефективність, супроводжуваність, можливість переносу); 4) загальні характеристики (витрати на технологію, ліцензійна політика, оціночний ефект від впровадження ТППЗ, потрібна для впровадження інфраструктура, доступність та якість навчання, сертифікація постачальника, підтримка постачальника). На основі даного набору параметрів аналізуються та класифікуються існуючі ТППЗ.

Враховуючи вищенаведені початкові дані для оцінювання ТППЗ, очевидними є наступні *критерії оцінювання ТППЗ*:

1) трудомісткість створення ПЗ – кількість людиномісяців, які прогнозовано будуть витрачені на створення ПЗ з використанням ТППЗ (цей критерій потрібно мінімізувати);

2) продуктивність – обсяг роботи (кількість рядків коду), який доводиться на одиницю трудомісткості (людиномісяць) при використанні даної ТППЗ (критерій вимагає максимізації);

3) кількість дефектів у створюваному ПЗ при використанні даної ТППЗ (критерій вимагає мінімізації);

4) рівень повернення інвестицій – даний критерій вимагає максимізації і обчислюється за формулою:

$$\text{Повернення}_\text{інвестицій} = \frac{(\text{Прибуток} - \text{Витрати})}{\text{Витрати}}, \quad (2)$$

де *Прибуток* – прибуток від використання ПЗ, *Витрати* – витрати на створення та супровід ПЗ;

5) витрати на супровід ПЗ – даний критерій вимагає мінімізації і обчислюється за формулою:

$$\text{Витрати}_\text{на}_\text{супровід} = \frac{\text{Вартість}_\text{супроводу}}{\text{Сукупні}_\text{витрати}}, \quad (3)$$

де *Вартість_супроводу* – це очікувана вартість етапу супроводу ПЗ при використанні даної ТППЗ, *Сукупні_витрати* – це витрати на впровадження даної ТППЗ в організації;

6) час впровадження ТППЗ – прогнозований часовий інтервал від початку впровадження ТППЗ до повного впровадження та використання ТППЗ всіма учасниками процесу розроблення, який вимагає мінімізації;

7) витрати на впровадження ТППЗ – очікувана сумарна вартість придбання, вивчення та супроводу ТППЗ (цей критерій потрібно мінімізувати);

8) термін окупності витрат на впровадження ТППЗ – часовий інтервал від початку впровадження ТППЗ до повної окупності витрат на її впровадження (критерій вимагає мінімізації).

Отже, група експертів повинна оцінити кожен із розглядуваних технологій ПЗ, враховуючи її складові частини (формула (1)), за всіма вищенаведеними критеріями. В результаті такої оцінки буде одержано наступну множину параметрів для i -ї розглядуваної ТППЗ:

$$EVAL = \{l, p, nd, ri, mc, it, ic, re\}, \quad (4)$$

де l – прогнозована трудомісткість створення ПЗ з використанням i -ї ТППЗ;
 p – продуктивність при використанні i -ї ТППЗ;
 nd – прогнозована кількість дефектів у створюваному ПЗ при використанні i -ї ТППЗ;
 ri – очікуваний рівень повернення інвестицій при використанні i -ї ТППЗ;
 mc – витрати на супровід ПЗ, яке розроблялось з використанням i -ї ТППЗ;
 it – очікуваний час впровадження i -ї ТППЗ;
 ic – очікувані витрати на впровадження i -ї ТППЗ;
 re – термін окупності витрат на впровадження i -ї ТППЗ.

Тоді, якщо експертами оцінюються $jtsd$ технологій проектування програмного забезпечення, то отримаємо $jtsd$ множин $EVAL$ (формула (4)), тобто $EVTSD = \{EVAL_1, \dots, EVAL_{jtsd}\}$, де $EVTSD$ – множина підмножин оцінок для всіх розглядуваних ТППЗ.

Множину підмножин $EVTSD$ доцільніше представити у вигляді наступної матриці (з врахуванням формули (4)):

$$EVTSD = \begin{bmatrix} l_1 & p_1 & nd_1 & ri_1 & mc_1 & it_1 & ic_1 & re_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ l_{jtsd} & p_{jtsd} & nd_{jtsd} & ri_{jtsd} & mc_{jtsd} & it_{jtsd} & ic_{jtsd} & re_{jtsd} \end{bmatrix}, \quad (5)$$

де в першому рядку наведено оцінки для технології проектування ПЗ №1, в другому рядку – оцінки для ТППЗ №2, і т.д., і в останньому рядку – оцінки для останньої технології проектування № $jtsd$.

Процес оцінювання тісно пов'язаний із процесом вибору ТППЗ. За результатами оцінювання цілі вибору та/або критерії вибору, а також їхні вагові коефіцієнти можуть вимагати модифікації. В таких випадках може знадобитись повторне оцінювання.

Алгоритм та продукційні правила вибору технології проектування програмного забезпечення

Алгоритм процесу вибору ТППЗ включає в себе наступні дії [2, 3]: 1) формулювання задач вибору, включаючи цілі, припущення та обмеження; 2) виконання всіх необхідних дій по вибору, включаючи визначення критеріїв, визначення технологій-кандидатів, збір необхідних даних та застосування критеріїв до результатів оцінювання для визначення засобів з найкращими показниками; 3) виконання необхідної кількості ітерацій з метою вибору (або відхилення) технології, яка має подібні до інших показники.

На основі формули (5) та опису критеріїв оцінювання ТППЗ, розробимо алгоритм вибору ТППЗ, оптимальної для організації:

Крок 1. Знайти мінімальні елементи 1-го (l), 3-го (nd), 5-го (mc), 6-го (it), 7-го (ic) та 8-го (re) стовпців матриці $EVTSD$ і зберегти номери їхніх рядків у змінні $\min_1, \min_3, \min_5, \min_6, \min_7, \min_8$ відповідно.

Крок 2. Знайти максимальні елементи 2-го (p) та 4-го (ri) стовпців матриці $EVTSD$ і зберегти номери їхніх рядків у змінні $\max_2, \max_4$ відповідно.

Крок 3. За допомогою продукційних правил визначити найкращу технологію проектування ПЗ за кожним критерієм, призначити 1 бал технології-«переможцю» за кожним критерієм, порахувати суму балів для кожної ТППЗ і зберегти суму балів технології № ki у елемент № ki множини ST , де $ST = \{st_1, \dots, st_{jtsd}\}$ – множина кількостей балів для кожної технології проектування, причому st_1 – кількість балів, набраних технологією №1, st_{nt} – кількість балів, набраних технологією № $jtsd$, $jtsd$ – кількість розглядуваних ТППЗ.

Крок 4. Знайти максимальний елемент множини $ST = \{st_1, \dots, st_{jtsd}\}$ і зберегти його номер у змінній not .

Враховуючи стандартний алгоритм процесу вибору ТППЗ, запропоновані критерії оцінювання ТППЗ та розроблений допоміжний алгоритм вибору оптимальної ТППЗ, побудуємо продукційні правила для вибору технології проектування СПЗ:

якщо змінна $\min_1 = ti$ (тобто за критерієм «трудомісткість створення ПЗ» найкращою є технологія за номером ti), то технологія за номером ti одержує 1 бал: $st_{ti} = st_{ti} + 1$ (st_{ti} – кількість балів ti -ї технології, де $st_{ti} \in ST$);

якщо змінна $\max_2 = ti$ (тобто за критерієм «продуктивність» найкращою є технологія за номером ti), то технологія за номером ti одержує 1 бал: $st_{ti} = st_{ti} + 1$;

якщо змінна $\min_3 = ti$ (тобто за критерієм «кількість дефектів у створюваному ПЗ» найкращою є технологія за номером ti), то технологія за номером ti одержує 1 бал: $st_{ti} = st_{ti} + 1$;

якщо змінна $\max_4 = ti$ (тобто за критерієм «рівень повернення інвестицій» найкращою є технологія

за номером ti), то технологія за номером ti одержує 1 бал: $st_{ti} = st_{ti} + 1$;

якщо змінна $\min_5 = ti$ (тобто за критерієм «витрати на супровід ПЗ» найкращою є технологія за номером ti), то технологія за номером ti одержує 1 бал: $st_{ti} = st_{ti} + 1$;

якщо змінна $\min_6 = ti$ (тобто за критерієм «час впровадження ТППЗ» найкращою є технологія за номером ti), то технологія за номером ti одержує 1 бал: $st_{ti} = st_{ti} + 1$;

якщо змінна $\min_7 = ti$ (тобто за критерієм «витрати на впровадження ТППЗ» найкращою є технологія за номером ti), то технологія за номером ti одержує 1 бал: $st_{ti} = st_{ti} + 1$;

якщо змінна $\min_8 = ti$ (тобто за критерієм «термін окупності витрат від впровадження ТППЗ» найкращою є технологія за номером ti), то технологія за номером ti одержує 1 бал: $st_{ti} = st_{ti} + 1$;

якщо змінна $not = toi$, то технологія за номером toi є найбільш оптимальною для впровадження в даній організації.

Розроблені математична модель технології проектування програмного забезпечення, критерії оцінювання ТППЗ, алгоритм та продукційні правила вибору технології проектування програмного забезпечення надають софтверній організації можливість мотивованого та обґрунтованого вибору технології проектування для її подальшого впровадження.

Математична модель середовища програмування

Середовище програмування служить для розроблення програм і є орієнтованим на конкретну мову або декілька мов програмування [8]. Загальний аналіз сучасних середовищ програмування [9] дав можливість визначити основні характеристики середовища програмування: 1) доступні у середовищі мови програмування; 2) операційні системи, з якими сумісні розроблені у середовищі прикладні програми; 3) ціна; 4) інтерфейс; 5) наявність компонентів для проведення тестування.

Виходячи з аналізу сучасних середовищ програмування [8] та характеристик середовища програмування, побудуємо математичну модель середовища програмування:

$$PE = \{APL, OSC, ECost, Elnt, ACT\}, \quad (6)$$

де $APL = \{apl_1, \dots, apl_{Qapl}\}$ – множина доступних у середовищі мов програмування ($Qapl$ – кількість доступних мов програмування);

$OSC = \{osc_1, \dots, osc_{Qosc}\}$ – множина операційних систем, з якими сумісні розроблені у середовищі прикладні програми ($Qosc$ – кількість операційних систем, з якими сумісні розроблені прикладні програми);

$ECost$ – експертна оцінка цінового рівня середовища програмування (наприклад, за 5-бальною шкалою (оцінка «5» позначає найнижчий ціновий рівень): «1» – для Microsoft Visual Studio, «5» – для безкоштовного Sharp Developer і т.і.);

$Elnt$ – експертна оцінка інтерфейсу (наприклад, за 5-бальною шкалою (оцінка «5» позначає найкращий інтерфейс): «5» – для Borland Delphi, «4» – для Sharp Developer і т.і.);

$ACT = \{act_1, \dots, act_{Qact}\}$ – множина компонентів середовища, призначених для проведення тестування ($Qact$ – кількість компонентів середовища для проведення тестування).

Критерії оцінювання середовища програмування

Аналіз середовищ програмування, виконаний за допомогою методу аналізу ієрархій у [9], дав можливість сформулювати наступні критерії оцінювання середовища програмування:

- 1) кількість доступних у середовищі мов програмування – критерій потребує максимізації;
- 2) кількість операційних систем, з якими сумісні розроблені у середовищі прикладні програми, критерій потребує максимізації;
- 3) ціна середовища програмування – критерій потребує мінімізації;
- 4) експертна оцінка інтерфейсу користувача середовища програмування – критерій потребує максимізації;
- 5) кількість компонентів середовища, призначених для проведення тестування – критерій потребує максимізації;
- 6) наявність довідника (за 2-бальною шкалою, де оцінка «1» позначає наявність довідника, а оцінка «0» – його відсутність) – критерій потребує значення «1»;
- 7) експертна оцінка простоти використання середовища (за 5-бальною шкалою, де оцінка «5» позначає найпростіше для використання середовище, а оцінка «1» – найважче для використання середовища) – критерій потребує максимізації;
- 8) експертна оцінка читабельності коду, розробленого у середовищі (за 5-бальною шкалою, де оцінка «5» позначає найбільш читабельний код, а оцінка «1» – найменш читабельний код, розроблений у середовищі);
- 9) середня продуктивність програміста у середовищі – вимірюється кількістю часу, витраченого на одиницю продукції. Так, для розроблення ПЗ продуктивністю вважатимемо кількість часу, витрачену на 1 рядок коду, тобто продуктивність розраховується за формулою:

$$\text{Продуктивність} = \frac{\text{Кількість_рядків}}{\text{Кількість_часу}}, \quad (7)$$

де Кількість_рядків – кількість рядків раніше розробленого аналогічного програмного засобу, який розроблявся у даному середовищі, Кількість_часу – це кількість часу (в годинах або хвилинах), витраченого на розроблення розглядуваного програмного засобу в даному середовищі. Звісно, що більш доцільно розглядати не один розроблений раніше програмний засіб, а множину таких засобів і визначати середню продуктивність програміста у середовищі. Критерій потребує максимізації;

10) наявність відлагоджувача у середовищі програмування (за 2-бальною шкалою, де оцінка «1» позначає наявність довідника, а оцінка «0» – його відсутність) – критерій потребує значення «1».

Кожне наявне середовище оцінюється експертами за вищенаведеними критеріями, в результаті чого буде одержано наступну множину параметрів для j -го середовища програмування:

$$\text{EVALPE} = \{Qapl, Qosc, Cost, EInt, Qact, Ha, ESu, ERa, APt, Da\}, \quad (8)$$

де $Cost$ – ціна j -го середовища програмування;

Ha – наявність довідника у j -му середовищі програмування;

ESu – експертна оцінка простоти використання j -го середовища;

ERa – експертна оцінка читабельності коду, розробленого у j -му середовищі;

APt – середня продуктивність програміста у j -му середовищі;

Da – наявність відлагоджувача у j -му середовищі програмування.

Тоді, якщо експертами оцінюються qep середовищ програмування, то отримаємо qep множин EVALPE (формула (8)), тобто $\text{EVPE} = \{\text{EVALPE}_1, \dots, \text{EVALPE}_{qep}\}$, де EVPE – множина підмножин оцінок для всіх розглядуваних середовищ програмування.

Множину підмножин EVPE доцільніше представити у вигляді наступної матриці (з врахуванням формули (8)):

$$\text{EVPE} = \begin{bmatrix} Qapl_1 & Qosc_1 & Cost_1 & EInt_1 & Qact_1 & Ha_1 & ESu_1 & ERa_1 & APt_1 & Da_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Qapl_{qep} & Qosc_{qep} & Cost_{qep} & EInt_{qep} & Qact_{qep} & Ha_{qep} & ESu_{qep} & ERa_{qep} & APt_{qep} & Da_{qep} \end{bmatrix}, \quad (9)$$

де в першому рядку наведено оцінки для середовища програмування №1, в другому рядку – оцінки для середовища програмування №2, і т.д., і в останньому рядку – оцінки для останнього розглядуваного середовища програмування № qep .

Алгоритм та продукційні правила вибору середовища програмування для програмного забезпечення

На основі формули (9) та опису критеріїв оцінювання середовища програмування, розробимо алгоритм вибору оптимального середовища програмування.

Крок 1. Знайти мінімальний елемент 3-го ($Cost$) стовпця матриці EVPE і зберегти номер його рядка у змінній $\min_3$.

Крок 2. Знайти максимальні елементи 1-го ($Qapl$), 2-го ($Qosc$), 4-го ($EInt$), 5-го ($Qact$), 7-го (ESu), 8-го (ERa), 9-го (APt) стовпців матриці EVPE і зберегти номери їхніх рядків у змінні $\max_1$, $\max_2$, $\max_4$, $\max_5$, $\max_7$, $\max_8$, $\max_9$ відповідно.

Крок 3. Знайти елементи, які дорівнюють 1, у 6-му (Ha) та 10-му (Da) стовпцях матриці EVPE і зберегти номери їхніх рядків у змінні one_6 , one_{10} відповідно.

Крок 4. За допомогою продукційних правил визначити найбільш оптимальне середовище програмування для ПЗ за кожним критерієм, призначити 1 бал середовищу-«переможцю» за кожним критерієм, порахувати суму балів для кожного середовища і зберегти суму балів середовища № pei у елемент № pei множини EP , де $EP = \{ep_1, \dots, ep_{qep}\}$ – множина кількостей балів для кожного середовища програмування, причому ep_1 – кількість балів, набраних середовищем №1, ep_{qep} – кількість балів, набраних середовищем № qep , qep – кількість розглядуваних середовищ програмування.

Крок 5. Знайти максимальний елемент множини $EP = \{ep_1, \dots, ep_{qep}\}$ і зберегти його номер у змінній per .

Враховуючи стандартний алгоритм процесу вибору середовища програмування, запропоновані критерії оцінювання середовищ програмування та розроблений алгоритм вибору оптимального середовища, побудуємо продукційні правила для вибору оптимального середовища програмування:

1) якщо змінна $\max_1 = ter$ (тобто за критерієм «кількість доступних у середовищі мов програмування» найкращим є середовище за номером ter), то середовище за номером ter одержує 1 бал: $sep_{ter} = sep_{ter} + 1$ (sep_{ter} – кількість балів ter -го середовища, де $sep_{ter} \in EP$);

2) якщо змінна $\max_2 = tep$ (тобто за критерієм «кількість операційних систем, з якими сумісні розроблені у середовищі прикладні програми» найкращим є середовище за номером tep), то середовище за номером tep одержує 1 бал: $sep_{tep} = sep_{tep} + 1$;

3) якщо змінна $\min_3 = tep$ (тобто за критерієм «ціна середовища програмування» найкращим є середовище за номером tep), то середовище за номером tep одержує 1 бал: $sep_{tep} = sep_{tep} + 1$;

4) якщо змінна $\max_4 = tep$ (тобто за критерієм «експертна оцінка інтерфейсу користувача середовища програмування» найкращим є середовище за номером tep), то середовище за номером tep одержує 1 бал: $sep_{tep} = sep_{tep} + 1$;

5) якщо змінна $\max_5 = tep$ (тобто за критерієм «кількість компонентів середовища, призначених для проведення тестування» найкращим є середовище за номером tep), то середовище за номером tep одержує 1 бал: $sep_{tep} = sep_{tep} + 1$;

6) якщо змінна $one_6 = tep$ (тобто за критерієм «наявність довідника» найкращим є середовище за номером tep), то середовище за номером tep одержує 1 бал: $sep_{tep} = sep_{tep} + 1$;

7) якщо змінна $\max_7 = tep$ (тобто за критерієм «експертна оцінка простоти використання середовища» найкращим є середовище за номером tep), то середовище за номером tep одержує 1 бал: $sep_{tep} = sep_{tep} + 1$;

8) якщо змінна $\max_8 = tep$ (тобто за критерієм «експертна оцінка читабельності коду, розробленого у середовищі» найкращим є середовище за номером tep), то середовище за номером tep одержує 1 бал: $sep_{tep} = sep_{tep} + 1$;

9) якщо змінна $\max_9 = tep$ (тобто за критерієм «середня продуктивність програміста у середовищі» найкращим є середовище за номером tep), то середовище за номером tep одержує 1 бал: $sep_{tep} = sep_{tep} + 1$;

10) якщо змінна $one_{10} = tep$ (тобто за критерієм «наявність відлагоджувача у середовищі програмування» найкращим є середовище за номером tep), то середовище за номером tep одержує 1 бал: $sep_{tep} = sep_{tep} + 1$;

11) якщо змінна $nep = spe$, то середовище за номером spe є найбільш оптимальним.

Розроблені математична модель середовища програмування, критерії оцінювання середовища програмування, алгоритм та продукційні правила вибору середовища програмування надають підтримку вибору оптимального середовища програмування для програмного забезпечення відповідно до вимог користувача.

Висновки

У статті розроблено математичну модель технології проектування програмного забезпечення та критерії оцінювання ТППЗ, які дають можливість експертам оцінити кожну розглядану технологію проектування програмного забезпечення більш точно, з врахуванням всіх її складових частин. Запропоновані автором алгоритм та продукційні правила вибору технології проектування програмного забезпечення надають організації можливість мотивованого та обґрунтованого вибору технології проектування для її подальшого впровадження.

Також у статті розроблено математичну модель середовища програмування та критерії оцінювання середовища програмування, які дають можливість оцінити середовище програмування з точки зору вимог користувача. Запропоновані автором критерії та продукційні правила вибору середовища програмування для програмного забезпечення підтримуватимуть організацію у виборі середовища програмування для програмного забезпечення.

Перспективними напрямками подальших досліджень є: 1) побудова методу вибору технології проектування програмного забезпечення на основі розроблених математичної моделі ТППЗ, критеріїв оцінювання ТППЗ, а також алгоритму та продукційних правил вибору ТППЗ; 2) побудова методу та алгоритму вибору середовища програмування для програмного забезпечення; 3) проектування та реалізація системи підтримки прийняття рішень для вибору технології проектування та середовища програмування для програмного забезпечення.

Література

1. Мищенко В.О. CASE-оценка критических программных систем : в 3-х т. / Мищенко В.О., Поморова О.В., Говорущенко Т.А. ; Под ред. Харченко В.С. – Харьков : НАУ «ХАИ», 2012. – Т. 1. – 201 с.
2. Чернев Д.А. Технология разработки программного обеспечения / Чернев Д.А. – Ташкент, 2004 –

224 с.

3. J.Greenfield, K.Short et al. Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools – John Wiley & Sons, 2004 – 666 p. // [Electronic resource] – Access mode: <http://iclass.iuea.ac.ug/intranet/Ebooks/INFORMATION%20TECHNOLOGY/SOFTWARE%20ENGINEERING/%5BWiley%5D%20Software%20Factories%20-%20Assembling%20Applications%20with%20Patterns,%20Models,.pdf>
4. ISO/IEC 12207:2008. Systems and software engineering – Software life cycle processes
5. ISO 9000:2005 Quality management systems – Fundamentals and vocabulary // [Electronic resource] – Access mode: http://thuvienkhcn.vinhlong.gov.vn/tailieukhcn/data/TieuChuantuanvan/ISO/ISO_9000_2005.pdf
6. Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber. Capability Maturity Model fro Software - Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University // [Electronic resource] – Access mode: <http://www.sei.cmu.edu/reports/93tr024.pdf>
7. Говорущенко Т.О. Оцінювання та вибір технології проектування системного програмного забезпечення / Т.О. Говорущенко, Є.Б.Курас, М.В. Красовський // III Всеукраїнська НПК молодих учених та студентів «ІТСП-2014». Збірник наукових праць – Хмельницький : Триада-М, 2014
8. Учебник по программированию для начинающих [Електронний ресурс] – Режим доступа : <http://iguania.ru/uchebnik-po-osnovam-programmirovaniya-dlya-nachinauschich/glava-4-sreda-programmirovaniya.html>
9. Говорущенко Т.О. Підтримка вибору середовища програмування для системного програмного забезпечення / Т.О. Говорущенко, Р.А. Малирчук, В.В. Лаврінчук // III Всеукраїнська НПК молодих учених та студентів «ІТСП-2014». Збірник наукових праць – Хмельницький : Триада-М, 2014

References

1. Mishchenko V.O., Pomorova O.V., Hovorushchenko T.A. CASE-otsenka kriticheskikh programmnih sistem. Tom 1. Kachestvo / Pod red. Kharchenko V.S. - Kharkov: NAU "KhAI", 2012 - 201 s. [in Russian]
2. Chernov D.A. Tekhnologiya razrabotki programmnogo obespecheniya – Tashkent, 2004 – 224 s.
3. J.Greenfield, K.Short et al. Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools – John Wiley & Sons, 2004 – 666 p. // [Electronic resource] – Access mode: <http://iclass.iuea.ac.ug/intranet/Ebooks/INFORMATION%20TECHNOLOGY/SOFTWARE%20ENGINEERING/%5BWiley%5D%20Software%20Factories%20-%20Assembling%20Applications%20with%20Patterns,%20Models,.pdf>
4. ISO/IEC 12207:2008. Systems and software engineering — Software life cycle processes
5. ISO 9000:2005 Quality management systems – Fundamentals and vocabulary // [Electronic resource] – Access mode: http://thuvienkhcn.vinhlong.gov.vn/tailieukhcn/data/TieuChuantuanvan/ISO/ISO_9000_2005.pdf
6. Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber. Capability Maturity Model fro Software - Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University // [Electronic resource] – Access mode: <http://www.sei.cmu.edu/reports/93tr024.pdf>
7. T.O.Hovorushchenko, E.B.Kuras, M.V.Krasovskiy. Otsinuvannya ta vibir tekhnologii proektuvannya sistemnogo programnogo zabezpechennya // III Vseukrainska NPK molodih uchenih ta studentiv “ITSP-2014”. Zbirnik naukovih prac – Khmelniitskiy: Triada-M, 2014
8. Uchebnik po programmirovaniu dlya nachinaushchih // [Electronic resource] – Access mode: <http://iguania.ru/uchebnik-po-osnovam-programmirovaniya-dlya-nachinauschich/glava-4-sreda-programmirovaniya.html>
9. T.O.Hovorushchenko, R.A.Malyarchuk, V.V.Lavrinchuk. Pidtrymka viboru seredovishcha programuvannya dlya sistemnogo programnogo zabezpechennya // III Vseukrainska NPK molodih uchenih ta studentiv “ITSP-2014”. Zbirnik naukovih prac – Khmelniitskiy: Triada-M, 2014

Рецензія/Peer review : 18.3.2014 р.

Надрукована/Printed :9.4.2014 р.

Рецензент: Шалапко Ю. І., д.т.н., професор кафедри ІМ, ХНУ,