

УДК 004.032.34

Шевченко В. Л., д.т.н., професор¹ (ORCID 0000-0002-9457-7454);
Федорієнко В. А.² (ORCID 0000-0002-0921-3390)

¹ - Кафедра Програмних систем і технологій Факультету інформаційних технологій Київського Національного університету імені Тараса Шевченка;

² - Центр воєнно-стратегічних досліджень Національного університету оборони України імені Івана Черняхівського, Київ

Визначення корисного ефекту динамічної залежності працездатності спеціального програмного забезпечення від фаху програмістів

Резюме. У роботі на основі статистичних даних зібраних під час супроводження проектів інформатизації як військового, так і цивільного призначення знайдено апроксимаційні моделі залежностей функції динамічних (часових) залежностей рівня працездатності спеціального програмного забезпечення трьох типів складності на прикладі ERP-систем від фаху програмістів I, II та III класу. Розглянуто три підходи щодо визначення корисного ефекту динамічної залежності працездатності спеціального програмного забезпечення від фаху програмістів. Детальніше розкрито підхід щодо пошуку часткового (максимального) корисного ефекту, як визначеного максимуму розвитку зазначеної динамічної залежності.

Ключові слова: частковий (максимальний) корисний ефект; фаховий клас програмістів; спеціальне програмне забезпечення; програмне забезпечення ERP-систем; метод золотого перетину.

Постановка проблеми. На сьогодні у Збройних Силах України здійснюється втілення проектів інформатизації за різними напрямками. У Програмі розвитку Збройних Сил України до 2020 року та визначених цілей, закладених у Стратегічному оборонному бюлетені України [1], визначені завдання щодо створення систем бойового управління (C4ISR, стратегічна ціль 1) та управління плануванням оборонними ресурсами (DRMIS, стратегічна ціль 4). Організаційна компонента щодо створення, супроводження та впровадження програмного забезпечення є ключовою. Спостерігається неоднозначність у підходах до менеджменту організаційної компоненти для підтримки та супроводження програмних систем із збереженням властивості надійності програмної компоненти.

Вирішення проблемних питань створення інформаційних технологій з погляду обґрунтованого вибору складових компонент (організаційна, програмна, технічна та мережева) здатні підвищити ефективність проектів інформатизації. Вплив розвитку IT-сфери, допомога Україні з боку Європейських країн і США у цій галузі, змінюють ситуацію в Збройних Силах України на краще. Це призвело до появи низки науково-дослідних і дослідно-конструкторських робіт. Через це нині проведення інформатизації структурних підрозділів є першочерговим завданням.

Процес створення систем, згідно з державними та світовими стандартами

(ISO/IEC 12207: 1995, ГОСТ 19.102-77 та ін.), лише поверхнево описує кількісний і якісний склад спеціалістів відповідальних за належне створення та модифікацію програмної компоненти (програмістів), прийняття її в експлуатацію. В умовах ресурсних обмежень обґрунтована конкретизація в окресленій проблемній області дасть змогу менеджерам вчасно приймати рішення та гнучкіше реагувати на зміни під час реалізації проектів інформатизації. Як наслідок, це призведе до раціональної витрати коштів та часу при розробленні та експлуатації основних програмних елементів.

Загальновідомо, що будь-який програмно-технічний комплекс потребує уваги, щодо підтримки його у працездатному стані. Виробники технологій систем класу ERP (Enterprise Resource Planning) пропонують власне стандартне програмне забезпечення (ПЗ) із максимальним наближенням інструментарію та програмних модулів до функціональних потреб використання. До того ж, враховуються такі аспекти: специфіка області використання, рівень приведення ПЗ та підтримки його відповідності національним, галузевим, внутрішнім вимогам нормативно-правових актів. Зазначені аспекти визначають складність проекту, і, відповідно, складність програмної компоненти. Під час виконання різних проектів інформатизації за будь-якої методологією WaterFlow, Cascade, Agile, EP тощо [2, 3, 4], що описується різними стандартами управління проектами (ISO 9000, PM Book, ГОСТ 34 та ін.) передбачаються етапи розроблення, впровадження та супроводження

ПЗ у межах проекту. У процесі цих робіт здійснюється створення та модифікація програмного коду спеціального програмного забезпечення (СПЗ).

Існує протиріччя між вимогою до якісного складу персоналу організаційної компоненти (програмістів), його максимальною продуктивністю щодо створення та модифікації СПЗ та можливостями оцінювання діяльності програмістів різних кваліфікаційних класів щодо створення та модифікації СПЗ. Звідси, дослідження питань оптимізації залежності кваліфікації програмістів від працездатності програмної системи в умовах ресурсних обмежень є актуальним.

Аналіз основних досліджень і публікацій. Значний внесок у розвиток інформаційних технологій для обґрунтування теоретичних і прикладних аспектів створення та модифікації ПЗ здійснили такі вчені, як В. С. Харченко, В. Г. Литвиненко, А. В. Волков, А. О. Хаджянян, К. Бек [2, 5, 6]. Їх дослідження стосувалися якості програмного забезпечення, а саме підвищення надійності ПЗ і розроблення теорії багатоверсійності під час створення ПЗ. Проте питання визначення рівня працездатності СПЗ залежно від класифікації програмістів розкриті поверхнево. Актуальність цього дослідження підкреслюється необхідністю створення методичних підходів щодо визначення корисних ефектів продуктивності відновлення працездатності СПЗ програмістами різних фахових рівнів.

Попередні дослідження цього питання раніше були проведені у площині визначення прийомів обробки статистичних даних появи та усунення помилок у СПЗ [7], уточнення фахового поділу програмістів на три класи [8], визначення впливу ефекту часових показників завантаженості на створення СПЗ ERP-системи [9] та аналізу підходів щодо визначення ймовірнісної моделі надійності СПЗ за допомогою методів Монте-Карло та системи масового обслуговування [10].

Метою статті є розкриття підходів щодо визначення корисних ефектів динамічних залежностей рівня працездатності спеціального програмного забезпечення ERP-систем від фаху програмістів.

Виклад основного матеріалу. В ERP системах існує низка напрямів з проведення робіт щодо підтримки системи у працездатному стані. Робота з підтримки працездатності розпочинається з аналізу завантаженості системи на предмет виявлення

проблемних областей, після чого отримані дані підлягають поглибленому аналізу для одержання детальнішої інформації [11]. Однією з причин ускладнення працездатності можуть бути дії програмістів під час створення та модифікації програмного коду СПЗ. Аналіз статистики діяльності програмістів різних кваліфікаційних класів за параметрами системних журналів помилок розроблення СПЗ, транспортного обміну запитів на зміну програмного коду, оновлення ERP-системи та показниками продуктивності роботи ERP-системи, пошук апроксимаційних залежностей дасть змогу описати вплив програмістів різного фаху на працездатність СПЗ системи [9].

Виведення математичних залежностей ($\text{exp} + \text{SL}$) для різних складностей СПЗ

Виокремлення статистики. Застосування автоматизованих програмних засобів щодо виміру якості програмного забезпечення дає змогу отримати метрики якості програмного забезпечення. Відповідно, на досліджуваному проекті, з метрик СПЗ можна було виділити три основні групи за складністю програм $S=1$ – прості, $S=2$ – середні, $S=3$ – складні. Подібний поділ використовують у методологіях ведення проектів [3], а математичне формулювання наведене у роботі [5].

У програмних середовищах розробки ERP-системи усі операції проведені із СПЗ у процесі його створення та модифікації автоматично записуються у системний журнал (system log file). Досліджуючи такі журнали програм відповідної складності за визначений період часу низки циклічних процедур щодо підтримки працездатності СПЗ (написання коду, активація, генерація, валідація, верифікація, тестування, усунення помилок) за параметром – *Відповідальна особа (програміст)*, була виявлена розбіжність у швидкості відновлення працездатності СПЗ різними за фахом програмістами. Було виділено три групи програмістів за швидкістю відновлення працездатності програм, які були віднесені до трьох фахових класів: I, II та III. Відповідно, було проаналізовано журнали діяльності програмістів у періоди появи та відновлення помилок у програмах СПЗ різної складності ($S=1$, $S=2$ чи $S=3$) на проектах (великих і середніх за розмірами) впровадження ERP-систем. За кожним журналом різного за складністю СПЗ, була виокремлена статистика типових представників програмістів кожного з трьох класів, які брали участь у відновленні відповідних програм за визначений період – для $S=1$ (статистичні дані діяльності програмістів I, II та III) за період 100

діб; для $S=2$ (статистичні дані діяльності програмістів I, II та III) за період 100 діб; для $S=3$ (статистичні дані діяльності програмістів I, II та III) за період 200 діб.

Регресійний аналіз. Визначення математичної функції залежності кількості працездатних програм K від часу t можливе під час проведення регресивного пошуку

апроксимаційної залежності статистичних даних. За допомогою проведення регресійного аналізу *методом найменших квадратів* (МНК) можливо знайти релевантну математичну функцію. Характерний обрис природи явища відновлення СПЗ, що цілком відповідає природі функції, можна отримати грубо розділивши точковий графік на дві складові [7] (рис. 1).

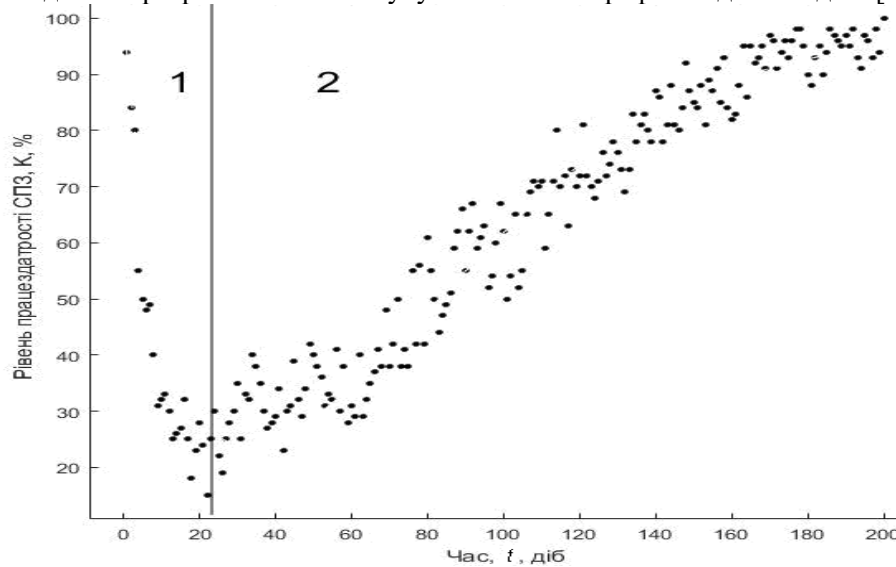


Рис. 1. Графік динамічної залежності виправлення помилок програмістом в СПЗ(складність $S=3$) програмістом II класу із декомпозицією його на дві частини

Підбір коефіцієнтів під час апроксимування дав змогу підібрати складові з розумінням природи фізики процесу. Техніка застосування МНК включає у себе поділ точкового графіку статистичних даних на дві частини для подальшого пошуку регресійної моделі залежності рівня працездатності СПЗ у часі для різного фаху програміста для кожної із частин. Такий спосіб вдало розкриває природу цієї залежності у вигляді двох регресійних функцій і має краще значення показника якості апроксимації – квадрату кореляції між значеннями відгуку виміру і передбаченими значеннями виміру у межах $R^2=0,9\pm 0,05$ [7]. Під час дослідження було проведено регресійний аналіз МНК статистичних даних результатів відновлення працездатності СПЗ зі складністю $S=1$, $S=2$ та $S=3$ кожного з класів програмістів.

З огляду на поділ на дві складові, ця функція описує явище появи та відновлення помилок у програмі суб'єктом програмування. Ця функціональна модель являє собою адитивну згортку двох динамічних (змінних у часі) функціональних моделей, що відповідають двом масивам статистичних даних, апроксимованих експоненціальною та логістичною SL залежностями (рис. 1). Ці

залежності при сумуванні дають новий тип кривої:

Кожна складність СПЗ та кожен із класів програмістів має логістичну залежність рівня працездатності СПЗ (корисного ефекту) від витрачених ресурсів (рис. 1), яка описується рівнянням в інтегральному вигляді

$$f(t) = g \cdot e^{(C \cdot t)} + \frac{s}{1 + e^{\frac{2}{T}(t-k)}}, \quad (1)$$

де $f(t) = K$ – рівень працездатності СПЗ (корисний ефект);

g - коефіцієнт експоненціальної функції;

C - постійна експоненціальної функції;

t - витрачений ресурс часу;

s - верхня асимптота;

T - постійна логістичної функції;

k - абсциса точки симетрії логістичної кривої.

Приклад відновлення СПЗ $S=3$ програмістом II класу.

Наведемо як приклад результати пошуку динамічної залежності відновлення працездатності СПЗ ($S=3$) МНК для програміста II класу. Для нього значення коефіцієнтів експоненціальної функції (1) та довірчих інтервалів (із довірчим обмеженням на рівні 95 %) становлять:

$$g = 93,08 \quad (81,99; 104,2);$$

$$C = -0,08997 \quad (-0,1066; -0,07334).$$

$R^2=0,8874$ для першої частини динамічної залежності програміста II класу.

Зважаючи на отримані значення першої частини статистичного графіку, були знайдені значення параметрів логістичної функції другої частини. Значення коефіцієнтів логістичної функції (1) та довірчих інтервалів (із довірчим обмеженням на рівні 95 %) для програміста II класу становлять:

$$\begin{aligned} k &= 81,66 \text{ (77,97; 85,35);} \\ s &= 97,99 \text{ (94,75; 101,2);} \\ T &= 68,5 \text{ (62,64; 74,36).} \end{aligned}$$

Критерій придатності наближення (коефіцієнт змішаної кореляції) для другої частини дорівнює $R^2=0,9306$.

Відповідно, рівняння динамічної залежності працездатності програм СПЗ програмістом II класу набуде вигляду загальної адитивної згортки функцій обох частин статистичного точкового графіку залежності рівня працездатності СПЗ ($S=3$) у часі, що відображає діяльність програміста II класу (представлена моделлю на рис. 2):

$$f(t) = 93,08 \cdot e^{-0,08997t} + \frac{97,99}{1 + e^{-\frac{2}{68,5}(t-81,66)}} \quad (2)$$

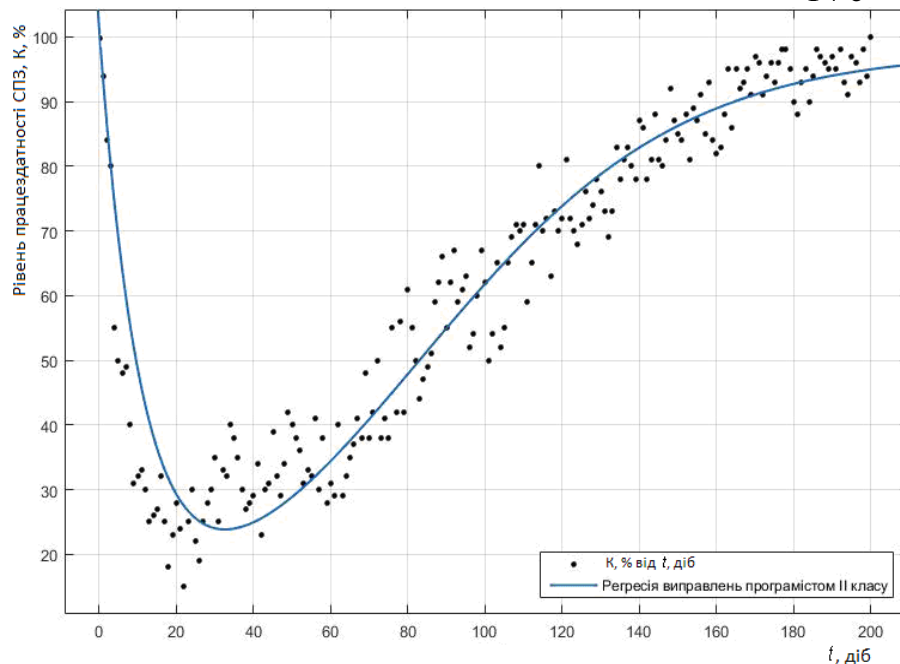


Рис. 2. Графік залежності відновлення працездатності СПЗ $Exp+SI$ (на прикладі СПЗ зі складністю $S=3$ для програміста II класу)

Значення функції (2) на рис. 2 являє собою відносну величину працездатних програм від загальної кількості програм СПЗ K узятих на початку дослідження на момент часу t на періоді дослідження 200 днів.

Відмінність між класами програмістів для однієї складності СПЗ полягає у часі реагування програмістів на помилки та їх усунення, що надходять із різною інтенсивністю. Поява помилок може бути викликана різними причинами. Наприклад, через невдале оновлення загальносистемного (стандартного) ПЗ, що призвело до непрацездатності програм СПЗ. Різні програми СПЗ мали різні режими виконання – діалоговий та фоновий. Деякі програми, що виконувалися у фоновому режимі мали заплановану періодичність запуску або виконання після настання певної запланованої події. Отже, не всі помилки могли бути

поміченими у системі відразу. Наприклад, за результатами проведеного дослідження, графоаналітичним аналізом встановлено, що час на виявлення всіх основних помилок, їх виправлення (з урахуванням внесення власних помилок), до початку стабільного росту значень відносної величини працездатних програм СПЗ ($S=3$) для програміста II класу становить 27 діб.

На рис. 2 наведено зразок однієї з дев'яти динамічних залежностей, отриманий з експерименту – точкового графіку рівня відновлення працездатності СПЗ зі складністю $S=3$ у часі для програміста II класу та її апроксимаційна модель.

Отримані унаслідок регресивного аналізу МНК коефіцієнти g, C, k, s, T кожного з трьох класів програмістів динамічної залежності відновлення працездатності СПЗ (1) для різних складностей СПЗ набудуть вигляду, наведеного у табл. 1.

Таблиця 1

Коефіцієнти динамічної залежності

	Програміст I класу	Програміст II класу	Програміст III класу
S=1	Перший рівень складності СПЗ. Апроксимована статистика МНК діяльності одного програміста за 100 днів. Джерело: дані оновлення апаратних засобів в ERP-системі ТОВ Фоззі Фуд		
<i>g</i>	96,30	98,05	95,11
<i>C</i>	-0,403	-0,263	-0,243
<i>k</i>	20,44	35,43	58,54
<i>s</i>	96,01	95,07	94,04
<i>T</i>	12,56	16,56	31,57
S=2	Другий рівень складності СПЗ. Апроксимована статистика МНК діяльності одного програміста за 100 днів. Джерело: дані переходу на новий податковий кодекс та архівація даних у системі ТОВ Фоззі Фуд		
<i>g</i>	98,03	91,12	96,10
<i>C</i>	-0,1099	0,1043	-0,1093
<i>k</i>	39,64	49,45	65,44
<i>s</i>	92,01	89,86	89,90
<i>T</i>	23,52	31,51	31,56
S=3	Третій рівень складності СПЗ. Апроксимована статистика МНК діяльності одного програміста за 200 днів. Джерело: дані оновлення загальносистемного ПЗ на ССУ АГП ЗСУ		
<i>g</i>	103,7	93,08	118,00
<i>C</i>	-0,1453	-0,08997	-0,1415
<i>k</i>	58,35	81,66	57,94
<i>s</i>	98,74	97,99	42,16
<i>T</i>	41,61	68,5	43,26

Поняття “часткового корисного ефекту”. У цій роботі під частковим корисним ефектом розуміється продуктивність роботи програміста, що виражена у відносних величинах працездатних програм СПЗ від їх загальної кількості програм на початку експерименту. Також, може розумітися, як обернена величина до зроблених помилок – чим менше помилок, тим вище ефект програміста, через що ефект досягнення необхідного рівня якості СПЗ підвищується (має пряму залежність від якості програм).

Маючи підібрані моделі для різних фахових класів програмістів та для трьох рівнів складності СПЗ можливо отримати деякі часткові корисні ефекти на основних етапах життєвого циклу програмного забезпечення при обмеженнях:

- 1) на отримання максимальної віддачі діяльності програміста – на етапі створення СПЗ;
- 2) на достатність відносної величини працездатності СПЗ – на етапі впровадження СПЗ;
- 3) на час відведений програмісту на відновлення працездатності СПЗ – на етапі супроводження (підтримки) СПЗ.

Зупинимо свою увагу на даних обмеженнях. Звідси, можна використати *три підходи* до пошуку часткового корисного ефекту динамічної залежності відновлення працездатності СПЗ програмістами кожного з класів програмістів:

1. Пошук часткового (максимального) корисного ефекту, як визначеного максимуму розвитку функції.

2. Пошук часткового корисного ефекту, за умови достатнього обмеження рівня виправлених програм (помилки у програмах).

3. Пошук часткового корисного ефекту, за умови достатнього обмеження значення часу.

У цій роботі розглянемо детально перший підхід; другий і третій – поверхнево.

Підхід 1. Процедура визначення максимуму розвитку функції, що відображає частковий корисний ефект для програмістів кожного з визначених класів за умови обмеження на отримання максимальної віддачі діяльності програміста має вид:

1.1. Визначення глобального екстремуму функції - координати точки $Q_i(q, p)$ – мінімум функціональної залежності (exp+SL);

1.2. Визначення точки часткового корисного ефекту відновлення працездатності СПЗ, як точки максимального розвитку функції на другій частині графіка SL-функції.

Зазначену процедуру проведемо на прикладі динамічної залежності відновлення працездатності СПЗ (складність S=3) програмістом I класу. Поняття корисного ефекту і зона ефекту показана геометрично на кривій SL залежності працевитрат робітників у роботі [12].

1.1. Маючи адитивну динамічну залежність відновлення працездатності СПЗ програмістами (exp+SL), зосередимо увагу на другій складовій (SL). Залежність SL є зростаючою на усьому проміжку і відображає якість відновлення працездатності програм СПЗ

програмістом кожного з фахових класів для різних складностей СПЗ. Унімодальність динамічних залежностей працездатності СПЗ дає змогу знайти точку мінімуму Q простішими методами знаходження оптимуму функцій. Для цього використовуємо *метод золотого перетину* – метод пошуку

екстремуму дійсної функції однієї змінної на заданому відрізку (рис. 3). В основі методу лежить принцип поділу відрізка в пропорціях золотого перетину. Цей метод є одним з найпростіших чисельних методів розв’язання задач оптимізації.

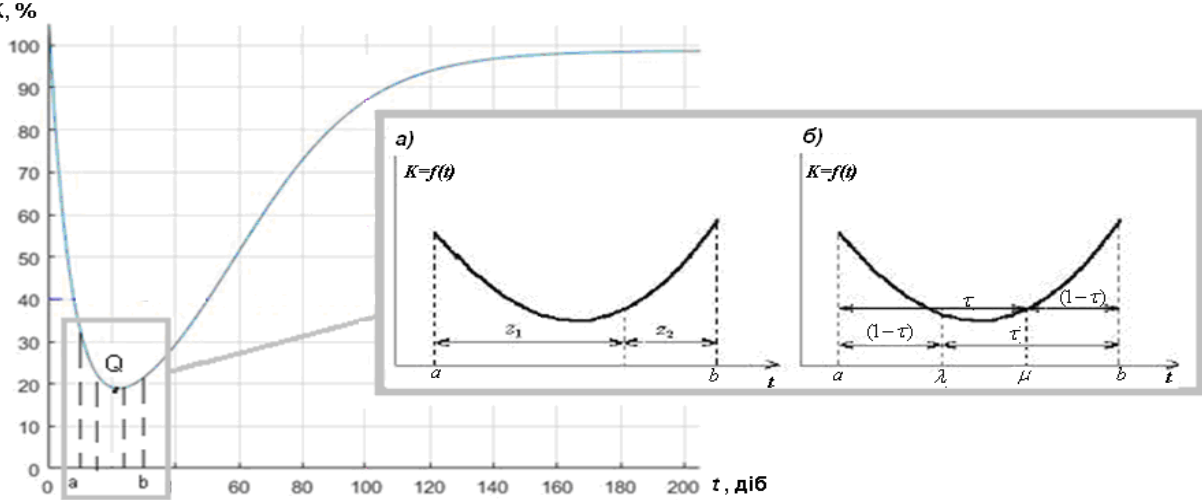


Рис. 3. Пошук точки $\min Q$ динамічної залежності відновлення працездатності СПЗ (складність $S=3$) програмістом I класу (Exp+SI)

Величина τ – пропорція “золотого перетину”, як відношення довжини більшого відрізка z_1 до довжини всього інтервалу z

дорівнюватиме відношенню довжини меншого відрізка z_2 до довжини більшого z_2 (рис. 3, а). Для цього визначимо його наближене значення:

$$\begin{cases} \frac{z_1}{z} = \frac{z_2}{z_1} \\ z_1 + z_2 = z \end{cases} \Rightarrow \left(\frac{z_2}{z_1}\right)^2 + \frac{z_2}{z_1} - 1 \Rightarrow \frac{z_2}{z_1} = \tau = \frac{-1 + \sqrt{5}}{2} \approx 0,618. \quad (3)$$

Загалом, суть *методу золотого перетину* полягає у тому, що для заданої функції $f(t) : [a, b] \rightarrow R$, $f(t) \in C([a, b])$ інтервал невизначеності (a, b) ділиться на дві нерівні частини у пропорції золотого перетину в обох напрямках від точки мінімуму Q .

Під час пошуку мінімуму функції, на кожному кроці ітеративної процедури, крім першого, обчислюється тільки одне значення функції. Однак, рекомендовано обчислювати на кожному кроці дві точки t_1 та t_2 (рис. 3, б), для того щоб не накопичувалася похибка, оскільки τ має наближене значення (рис. 3).

Якщо довжина кінцевого інтервалу невизначеності дорівнює δ , то для досягнення необхідної точності число обчислень значень функції за методом золотого перетину можна знайти за умовою

$$(0,618)^n \leq \frac{\delta}{b-a}. \quad (4)$$

Знайдемо точку мінімуму Q $[q; p]$ динамічної залежності відновлення працездатності СПЗ (складність $S=3$) програмістом I класу $f(t)$ на відрізку $[a; b]$ (відрізок визначається графоаналітичним

способом) з точністю ε і значення цільової функції в цій точці. Наприклад, на проміжку $[10; 30]$, $\varepsilon=0.1$:

$$f(t) = 103,7 \cdot e^{-0,1453t} + \frac{98,74}{1 + e^{\frac{-2}{41,61}(t-58,35)}}. \quad (5)$$

Алгоритм розв’язання цим методом складається з початкового та основного етапів.

На початковому етапі вводяться границі проміжку, на якому знаходиться мінімум. Нехай $\varepsilon > 0$ – задана точність. Припускаємо, що $a_1 = a = 10$; $b_1 = b = 30$. Визначимо дві довільні точки:

$$\lambda_1 = a_1 + (1 - \tau)(b_1 - a_1),$$

$\mu_1 = a_1 + \tau(b_1 - a_1)$, де $\tau = 0,618$, $(1 - \tau) = 0,382$. Звідси $\lambda_1 = 21,46$.

Розраховуємо $f(\lambda_1)$ і $f(\mu_1)$, відповідно $f(\lambda_1) = 18,9201$, $f(\mu_1) = 20,663$. Нехай $i=1$.

Під час основного етапу алгоритму знаходяться мінімум динамічної функціональної залежності рівня працездатності програм СПЗ:

а) якщо $(b_1 - a_1) < \varepsilon$,

то $x^* \in [a_i, b_i]$ обрахунки закінчити; інакше,

якщо $f(\lambda_1) > f(\mu_1)$, то перейти до кроку б);
 якщо $f(\lambda_1) < f(\mu_1)$, то перейти до кроку с);

б) прийmemo $a_{i+1} = \lambda_i, b_{i+1} = b_i,$
 $\lambda_{i+1} = \mu_i, \mu_{i+1} = a_{i+1} + \tau \cdot (b_{i+1} - a_{i+1});$ визначимо
 $f(\mu_{i+1})$ і перейдемо до кроку д);

с) прийmemo $a_{i+1} = a_i, b_{i+1} = \mu_i,$
 $\mu_{i+1} = \lambda_i, \lambda_{i+1} = a_{i+1} + (1 - \tau)(b_{i+1} - a_{i+1});$

визначимо $f(\lambda_{i+1})$ і перейдемо до кроку д);

д) $i = i + 1$, перейдемо до кроку а).

Виконавши розрахунки, що містять 20 ітерацій було знайдено значення функції $f(x)$, що задовольняють прийнятій точності ϵ . (Через значний обсяг, розрахунки у роботі не наведені).

Аналогічно наведеному прикладу було знайдено координати динамічної функціональної залежності рівня працездатності програм СПЗ зі складністю $S=3$ для програмістів I та III класів.

Отже, для СПЗ зі складністю $S=3$:

точка мінімуму Q_I для програміста I класу – $Q_I(22,1514; 18,8936)$;

точка мінімуму Q_{II} для програміста II класу – $Q_{II}(32,6145; 23,8402)$;

точка мінімуму Q_{III} для програміста III класу – $Q_{III}(28,1014; 10,6909)$.

Також, ця задача, для перевірки адекватності, була розв'язана складнішим методом Ньютона (Ньютона-Рафсона), який належить до групи методів однопараметричної оптимізації і є градієнтним методом другого порядку. У ньому під час пошуку екстремуму цільової функції використовувалися її перші і другі похідні. Отримані значення координат оптимуму функції (1) обома методами співпадають.

1.2. Визначення точки часткового корисного ефекту відновлення працездатності СПЗ, як точки максимального розвитку функції на другій частині графіка – SL-функції являє собою:

$$f(t)_{\text{dom}} = \left(103,7 \cdot e^{-0,1453a} + \frac{98,74}{1 + \frac{e^{-2(a-58,35)}}{41,61}} \right) + \left(-15,0676 \cdot e^{-0,1453a} + \frac{2,2829 \cdot 10^{51} \cdot e^{-2a}}{(1 + 1,156 \cdot 10^{49} \cdot e^{-2a})^2} \right) \cdot (t - a). \quad (9)$$

Перейдемо до вирішення системи рівнянь

$$\begin{cases} b - \left(-15,0676 \cdot e^{-0,1453a} + \frac{2,2829 \cdot 10^{51} \cdot e^{-2a}}{(1 + 1,156 \cdot 10^{49} \cdot e^{-2a})^2} \right) \cdot a = p, \\ b = 103,7 \cdot e^{-0,1453a} + \frac{98,74}{1 + \frac{e^{-2(a-58,35)}}{41,61}}, \end{cases} \quad (10)$$

де $p=18,8936$ (%) – значення функції точки мінімуму Q .

відкладення дотичної із значення функції точки мінімуму на осі ординат $(0; p)$ до другої частини графіку ($\exp+SL$) описаною SL-залежністю, знаходження рівняння дотичної;

знаходження точки дотичної до динамічної залежності працездатності програм СПЗ на періоді зростання виправлених програм – значення аргументу точки дотику відповідає значенню ефекту даної залежності.

Перейдемо до визначення точки дотику на прикладі динамічної залежності відновлення працездатності програм СПЗ (рівень складності $S=3$) програмістом I фахового класу, яка у загальному виді є адитивною згортокою двох функцій:

$$y(t) = \exp(t) + SL(t). \quad (6)$$

Оскільки цікавить час досягнення максимального корисного ефекту програмістом, то область допустимих значень припадатиме на проміжок $(q; 200)$ по осі абсцис представленою SL-функцією від значення аргументу точки Q до максимально дослідженого (допустимого) періоду підтримки працездатності програм СПЗ.

Розрахунок максимального корисного ефекту відновлення працездатності.

Спочатку знайдемо рівняння дотичної до графіка функції динамічної залежності, яка проходить через точку $(0; p)$. Зауважимо, що сама функціональна залежність не проходить через цю точку $(0; p)$, яка не є точкою дотику R .

Формульна модель узятого до розгляду прикладу динамічної залежності (програміст I класу, СПЗ $s=3$) має вид:

$$f(t) = 103,7 \cdot e^{-0,1453t} + \frac{98,74}{1 + \frac{e^{-2(t-58,35)}}{41,61}}. \quad (7)$$

Позначивши точку дотику як $R(a, b)$, рівняння дотичної набуде вигляду:

$$f(t)_{\text{dom}} = b + y'(a) \cdot (t - a). \quad (8)$$

Відповідно,

Знаходимо точку дотику на проміжку $(q; 200)$ на осі t – другої частини динамічної залежності відновлення працездатності СПЗ (зі складністю $S=3$) програмістом. Де, q – аргумент \min точки $Q; 200$ (днів) – граничний період дослідження експерименту.

У процесі обчислень за заданих умов (програміст I класу, складність СПЗ $S=3$) координати точки *максимального корисного ефекту* $R(a; b)$ становлять $R_I(90,3627; 81,2740)$ або, досягнення відносного рівня працездатності СПЗ програмістом I класу $b \approx 81,3$ (%) буде виконане на $a \approx 90$ (днів) розроб (підтримки) програм.

Аналогічно, були проведені обчислення для програмістів II та III класів (зі складністю СПЗ $S=3$).

Отже, для СПЗ зі складністю $S=3$:

точка *максимального корисного ефекту* R_I для програміста I класу – $R_I(90,3627; 81,2740)$;

точка *максимального корисного ефекту* R_{II} для програміста II класу – $R_{II}(133,4966; 80,3108)$;

точка *максимального корисного ефекту* R_{III} для програміста III класу – $R_{III}(93,6105; 35,3629)$.

Для візуалізації, динамічні залежності працездатності СПЗ зі складністю $S=3$ програмістів трьох фахових класів (y_I, y_{II}, y_{III}), їх мінімум (Q_I, Q_{II}, Q_{III}) та точки досягнення їх максимальних корисних ефектів (R_I, R_{II}, R_{III}) наведені на рис. 4.

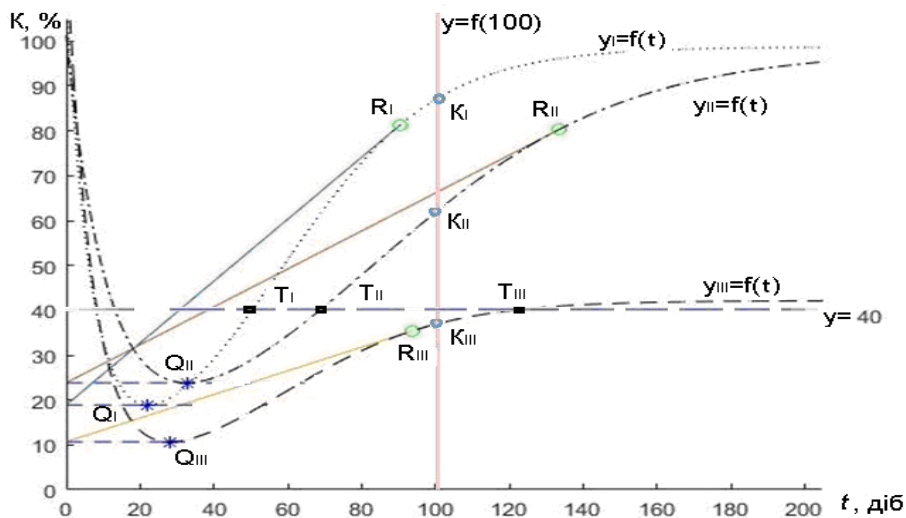


Рис. 4. Моделі динамічні залежності працездатності СПЗ (приклад, складності СПЗ $S=3$ програмістів I, II та III класів) із визначенням значень часткових корисних ефектів відновлення СПЗ (R_i, Q_i, K_i, T_i)

На рис. 4 точка дотику, R_i - точка на кривій динамічної залежності відновлення працездатності програм СПЗ, що означає найбільшу продуктивність діяльності програміста i -го класу, вказує на максимальний частковий корисний ефект, який можна очікувати від програміста. Це обумовлено подальшим переходом кривої діяльності програміста i -го класу у насичений стан у верхній частині S -подібного асимптотичного вигину, де очікується падіння продуктивності програміста. Цей підхід передбачає досягнення максимальних значень часткових корисних ефектів на кривих динамічної залежності відновлення працездатності СПЗ програмістами трьох класів. Точка R_i , є свого роду підказкою менеджерам проекту щодо ефективного використання програміста відповідно до його здібностей

Стисло розглянемо інші два підходи до пошуку часткового корисного ефекту динамічної залежності відновлення працездатності СПЗ програмістами кожного з класів програмістів.

Підхід 2. Процедура визначення часу необхідного на досягнення встановленого рівня виправлених програм (помилки у програмах) відображає частковий корисний ефект для програмістів кожного з визначених класів за умови обмеження на рівень працездатності K (%) – являє собою отримання значень часу. Виконується за допомогою пошуку точок T_I, T_{II}, T_{III} перетину кривих динамічної залежності працездатності СПЗ $y_i=f(t)$ із обмежуючою прямою (паралельній осі OX), що проходить через задане значення функції K (на осі OY) на ділянці зростання динамічної залежності (другої частини графіку, описаною SL -залежністю). Тобто, потрібно знайти корені

рівняння динамічної залежності працездатності СПЗ (1) при заданих значеннях функцій на проміжку правіше від аргументу точки мінімуму R_i . (Частковий корисний ефект виражений у часових значеннях). На рис. 4 наведено приклад обмеження $y=40\%$ працездатності СПЗ ($S=3$) для програмістів та отримані координати точок *обмежень* для СПЗ ($S=3$): T_I (50,3022; 40), T_{II} (68,6601; 40), T_{III} (121,0730; 40). Аргументи T_I , T_{II} , T_{III} – часткові корисні ефекти досягнення програмістами необхідного рівня працездатності СПЗ. Для програмістів різних класів, що працюють із СПЗ зі складністю $S=3$, досягнути достатній рівень працездатності програм ($y=40\%$) зможе: програміст I класу на 50-й день, програміст II класу на 69-й день, програміст III класу на 121-й день.

Підхід 3. Процедура визначення відносної величини працездатності СПЗ, що відображає частковий корисний ефект для програмістів кожного з визначених класів *за умови обмеження на часовий ресурс* T (діб) – являє собою отримання значень рівня виправлених програм K (%). Виконується за допомогою пошуку точок K_I , K_{II} , K_{III} перетину кривих динамічної залежності працездатності СПЗ $y_i=f(t)$ програмістів із обмежуючою прямою (паралельній осі ОУ), що проходить через заданий аргумент t (на осі ОХ). Тобто, потрібно знайти значення функції рівняння динамічної залежності працездатності СПЗ (1) при заданих значеннях аргументу. (Частковий корисний ефект виражений у значеннях відносної величини працездатності програм). На рис. 4 наведено приклад обмеження $t=100$ (діб) для програмістів та отримані координати точок *обмежень* для СПЗ ($S=3$) K_I (100; 86,9723), K_{II} (100; 61,8196), K_{III} (100; 36,8837). Значення функцій K_I , K_{II} , K_{III} – часткові корисні ефекти досягнення програмістами необхідного часу відведеного на створення та модифікацію СПЗ. Для програмістів різних класів, що працюють із СПЗ зі складністю $S=3$, під час досягнення часового обмеження на створення та модифікацію СПЗ, рівень працездатності СПЗ становитиме: для програміста I класу на 87 %, для програміста II класу на 62 %, для програміста III класу на 37 % (менше за встановлене обмеження у Підході 2 $y=40\%$).

У такий спосіб були отримані результати розрахунків за трьома підходами пошуку часткового корисного ефекту динамічної залежності працездатності СПЗ (1) для кожної складності СПЗ ($S=1$, $S=2$, $S=3$) та

фаху програмістів за класами I, II та III. (У роботі наведені результати розрахунків для прикладу із СПЗ $S=3$).

Висновки. Отже, у роботі були визначені методом найменших квадратів апроксимаційні моделі залежностей функції динамічних (часових) залежностей рівня працездатності спеціального програмного забезпечення трьох типів складності для ERP-систем від фаху програмістів I, II та III класу. Ці залежності являють собою адитивну згортку експоненціальної і логістичної SL апроксимаційних функцій.

Розкрито три підходи щодо визначення корисних ефектів. Детальніше було розкрито підхід до пошуку часткового (максимального) корисного ефекту як визначеного максимуму розвитку зазначеної динамічної залежності, що включав пошук екстремуму функцій зазначених залежностей методом золотого перетину (із перевіркою – методом Ньютона), та знаходження точок максимального корисного ефекту програмістів.

Таким чином, була удосконалена динамічна модель функціональності програмної системи, яка враховує кількість працездатних програм, кількість помилок у програмах і фаховий рівень програмістів.

Надалі доцільно визначити економічні моделі рівня працездатності програм СПЗ різної складності ERP-систем для програмістів різного фаху та розрахувати часткові корисні ефекти для команд програмістів різних класів із урахуванням їх взаємовпливу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Про рішення Ради національної безпеки і оборони України від 20 травня 2016 року “Про Стратегічний оборонний бюлетень України” [Електронний ресурс]: Указ [видано Президентом України 06 червня 2016 р. №240/2016]. – Режим доступу: <http://www.president.gov.ua/documents/2402016-20137>.
2. Войтенко О. С. Когнітивні моделі та інформаційні технології управління проектами та програмами : автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.13.22 “Управління проектами та програмами” / Войтенко Олександр Степанович – Київ, 2008. – 22 с.
3. Framework for the agile development of innovative Product-Service-Systems for existing physical rehabilitation systems / [L. Asmar, M. Rabe, C. YeeLow та ін.]. // *Procedia Manufacturing*. – 2018. – № 24. – С. 147–152. – Режим доступу до ресурсу: <https://www.sciencedirect.com/science/article/pii/S2351978918305663?via%3Dihub#!>.
4. IEEE Standard Glossary of Software Engineering Terminology [Електронний ресурс] : IEEE Std

- 610.12-1990. — [Чинний від 1990-09-28]. — NY: The Institute of Electrical and Electronics Engineers, 1990. — 84 с. — (Standards Coordinating Committee of the Computer Society of the IEEE). — Режим доступу до стандарту: <http://dis.unal.edu.co/~icasta/ggs/Documentos/Normas/610-12-1990.pdf>.
5. Ханджян А. О. Повышение надежности программного обеспечения информационно-измерительных и управляющих систем безопасности ядерных радиационно опасных объектов : автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.11.16 “Информационно-измерительные и управляющие системы” / А. О. Ханджян. — М., 2006. — 27 с.
6. Бек К. Экстремальное программирование / Бек К — М. : Питер, 2002. — 220 с.
- 7 Регресивний аналіз пошуку функції залежності кількості працюючих програм єдиного інформаційного середовища при визначеній класності програмістів / В. Л. Шевченко, Ю. А. Кірпічніков, В. А. Федорієнко [та ін.] // Збірник наукових праць ЦВСД НУОУ ім. І. Черняхівського. — 2015. — № 3 (55). — С. 6 – 12.
8. Кірпічніков Ю. А. Теоретичні підходи для розрахунку штату програмістів, необхідних для підтримки єдиного інформаційного середовища / Ю. А. Кірпічніков, В. А. Федорієнко, О. В. Головченко, О. А. Кошлань // Збірник наукових праць Центру воєнно-стратегічних досліджень Національного університету оборони України імені Івана Черняхівського. — 2014. — № 3. — С. 133-139. — Режим доступу: http://nbuv.gov.ua/UJRN/Znprcvsd_2014_3_24.
9. Федорієнко В. А. Вплив ефекту часових показників завантаженості на оптимізацію створення спеціального програмного забезпечення ERP системи / В.А. Федорієнко // Системи озброєння і військова техніка. — 2018. — № 4 (56). — С. 143-151.
10. Модель оцінки надійності програмної компоненти єдиного інформаційного середовища / В. Л. Шевченко, Ю. А. Кірпічніков, В. А. Федорієнко [та ін.] // Збірник наукових праць ЦВСД НУОУ ім. І. Черняхівського. — 2014. — № 1 (50). — С. 144 – 151.
11. Response-Time Analysis for Mixed Criticality Systems : materials of international Symposium [32nd Real-Time Systems Symposium], (Vienna, November 29–December 2, 2011) , Vienna : IEEE, 2011. — 116-34. Режим доступу до ресурсу: <https://ieeexplore.ieee.org/abstract/document/6121424/metrics>.
12. Шевченко В.Л. Оптимізаційне моделювання в стратегічному плануванні. — К.: Центр воєнно-стратегічних досліджень Національного університету оборони України, 2011. — 283 с.

Стаття надійшла до редакційної колегії 07.04.2019

Шевченко В. Л., д.т.н., професор¹;

Федориєнко В. А.²

¹ - Кафедра Программных систем и технологий Факультета информационных технологий Киевского Национального университета имени Тараса Шевченко;

² - Центр воєнно-стратегічних досліджень Національного університету оборони України імені Івана Черняхівського, Київ

Определение полезного эффекта динамической зависимости работоспособности специального программного обеспечения от специальности программистов

Резюме. В работе на основе статистических данных, собранных в ходе сопровождения проектов информатизации как военного, так и гражданского назначения были найдены аппроксимационные модели зависимостей функции динамических (временных) зависимостей уровня работоспособности специального программного обеспечения трех типов сложности на примере ERP-систем от специальности программистов I, II и III класса. Рассмотрены три подхода к определению полезного эффекта динамической зависимости работоспособности специального программного обеспечения от класса программистов. Наиболее подробно был раскрыт подход по поиску частичного (максимального) полезного эффекта, как определенного максимума развития указанной динамической зависимости.

Ключевые слова: частичный (максимальный) полезный эффект; профессиональный класс программистов; специальное программное обеспечение; программное обеспечение ERP-систем; метод золотого сечения.

V. Shevchenko, DsT, professor¹;

V. Fedoriienko²

¹ - Program system and technologies Department of the Taras Shevchenko National University of Kyiv

² - Center for Military and Strategic Studies of the National Defence University of Ukraine named after Ivan Cherniakhovskiy, Kyiv

Determination of the beneficial effect of the dynamic dependence of the performance of special software on the specialty of programmers

Resume. In the work, based on the statistical data collected during the support of the informatization projects for both military and civilian purposes, approximation models of dependencies of the function of dynamic

(temporal) dependencies of the level of performance of the special software of three types of complexity on the example of ERP-systems from the specialty of programmers I, II and III types were found. Three approaches to determining the useful effect of the dynamic dependence of the workability of special software on the specialty of programmers were considered. The most detailed approach was to find a partial (maximum) useful effect as a defined maximum development of the indicated dynamic dependence.

Keywords: partial (maximum) beneficial effect; class of programmers; special software; ERP systems software; golden section method..