

С. С. ВЕЛИКОДНИЙ

МЕТОД ПРЕДСТАВЛЕННЯ ОЦІНКИ РЕІНЖІНІРИНГУ ПРОГРАМНИХ СИСТЕМ ЗА ДОПОМОГОЮ ПРОЕКТНИХ КОЕФІЦІЄНТІВ

Предметом роботи є проектні коефіцієнти, що уводяться до моделі реінжинірингу програмної системи. **Мета** роботи. Сформував метод представлення проектною оцінкою реінжинірингу програмних систем, на підставі якого приймається остаточне рішення щодо доцільності виконання реінжинірингу. Одним з головних завдань інженерії програмного забезпечення є створення теоретичних і прикладних основ швидкої та якісної побудови складних систем з більш простих програмних елементів, які виконано у сучасних мовах програмування. Фактично рішення цього завдання здійснюється шляхом збирання, об'єднання або інтеграції різномірних програмних ресурсів та компонентів повторного використання, що включають модулі, бібліотеки та програмні реалізації деякої складної програмної системи. За сучасними світовими тенденціями проектування програмного забезпечення: програмні системи повинні бути такими, що розвиваються та еволюціонують. **Завданням** статті є дослідження впливу проектних коефіцієнтів, що уводяться до моделі реінжинірингу програмної системи, на відповідність до фактичних статистичних даних вже реально виконаних проектів з реінжинірингу програмних систем. **Методи.** У статті знайшла свій розвиток методологія Бома, продовження якої відображене у формуванні аналітичних моделей із запропонованими змінами та пропозиціями щодо гнучкості побудови на етапі передпроектного аналізу. До моделей уведено проектні коефіцієнти, що являють собою удосконалення уявлень оцінки проектів за методом проектних точок Карнера та методом констант Якобсона із доданими доповненнями та розширеннями. Візуалізація моделей спирається на методи побудови спіральних моделей Архімеда, векторного представлення годографів Гамільтона та Михайлова, а метод обліку запланованого часу базується на проектних діаграмах Ганта. **Результати:** отримано метод за яким до раніш сформованих моделей реінжинірингу програмних систем, уведено проектні коефіцієнти, що дозволяють підвищити точність оцінки реінжинірингу, яка виражена у зменшенні середнього значення відносної похибки щодо фактичного часу його виконання. **Висновки.** До наукової новизни віднесено наступні положення щодо реінжинірингу програмних систем: удосконалено моделі візуалізації витрат за рахунок введення коефіцієнтів автоматизації та схожості компонентів, які дозволить вносити зміни до конфігурації відповідних годографів. Реінжиніринг програмних систем дозволить: подолати протиріччя між темпами розвитку науки, техніки й процесів проектування; підвищити ефективність технічного супроводу; скоротити експлуатаційні витрати.

Ключові слова: програмна система; реінжиніринг; передпроектні дослідження; проектування; еволюційна спіраль; годограф; проектні коефіцієнти; коефіцієнт автоматизації; коефіцієнт схожості компонентів; компоненти повторного використання.

Вступ

Однією з головних проблем інженерії програмного забезпечення є створення теоретичних і прикладних основ швидкої та якісної побудови складних систем з більш простих програмних елементів, які виконано у сучасних мовах програмування (МП). Фактично рішення цієї проблеми здійснюється шляхом збирання, об'єднання або інтеграції різномірних програмних ресурсів та компонентів повторного використання (КПВ), що включають модулі, бібліотеки та програмні реалізації деякої складної програмної системи (ПС).

У той же час, ПС об'єднує у собі різні види забезпечення: технічне, математичне, програмне, інформаційне, лінгвістичне, методичне, організаційне, ергономічне та правове. Кожне з цих видів забезпечення має параметри і бажані характеристики, які обирають проектувальники ПС із максимальним урахуванням особливостей завдань інженерного проектування, конструювання, технологій кодування та виготовлення ПЗ.

Створення та розвиток ПС завжди здійснюється провідною проектною корпорацією із обов'язковим залученням інших організацій-співвиконавців, у тому числі: наукових установ, профільних університетів, а також консультацій відомих вчених, науковий потенціал і накопичений досвід яких, виражено у сотнях наукових публікацій і десятках захищених

дисертацій під їх керівництвом.

На даний момент, створення потужної галузевої ПС – складна і трудомістка задача, виконання якої під силу тільки великому висококваліфікованому і злагодженому колективу розробників, оскільки тільки сам процес створення вже містить у собі понад десяток стадій: перед проектні (у тому числі соціологічні) дослідження, технічне завдання, технічна пропозиція, ескізний проект, технічний проект, робочий проект, виготовлення (кодування), налагодження, випробування (тестування), введення в дію (інсталяція та конфігурування) та подальший технічний супровід та підтримка.

За актуальними світовими тенденціями інженерії програмного забезпечення (ПЗ): ПС повинні бути такими, що розвиваються. Існує, принаймні, дві вагомі причини, за якими ПС повинні бути змінюваними за часом. По-перше: розробка ПС займає тривалий час і економічно вигідно вводити до експлуатації частини системи по мірі їх готовності (уведена в експлуатацію базова версія – надалі розширюється). По-друге: постійний прогрес об'єктів проектування, технологій виготовлення й кодування ПЗ, обчислювальної техніки та обчислювальної математики призводить до появи нових, більш досконалих математичних моделей і методів, які повинні замінювати старі, менш вдалі аналоги.

У зв'язку з цим, ПС повинна мати властивість зручності використання та можливості розширення за

допомогою підключення розроблених та / або удосконалених видів забезпечення. Ось тут постає питання подальшого вирішення цієї проблеми, а саме: це може бути нова розробка ПС або реінжиніринг (reengineering), тобто перепроєктування із наслідуванням позитивних якостей та властивостей ПС, що знаходиться у експлуатації, та відмовою від негативних.

Об'єктом роботи є процес реінжинірингу ПС.

Предметом роботи є проектні коефіцієнти, що уводяться до моделі реінжинірингу ПС.

Головним завданням роботи є дослідження впливу проектних коефіцієнтів, що уводяться до моделі реінжинірингу ПС, на відповідність до фактичних статистичних даних вже реально виконаних проектів з реінжинірингу ПС.

Аналіз останніх досліджень і публікацій

Застосування реінжинірингу (PI) щодо інженерії ПЗ, як початковий етап розвитку наукової думки з перепроєктування ПС розпочато у [1, 2].

Завдяки уведеній методології та обчислювальному інструментарію [3] можна виконати оцінку проектних ресурсів, спираючись на ідею Карнера [4], що закладено у основу розробленого метода розрахунку показників оцінки проекту при виконанні PI ПС [5].

Саме ж поняття "реінжиніринг", спочатку з'явилося у представленні оптимізації бізнес-процесів корпорації [6, 7], як переорієнтування та реорганізація структури забезпечення (у широкому сенсі) [8].

У роботі [9] було застосовано об'єктний підхід до бізнес-процесів у формі реструктуризації, та переведення окремих компонентів системи до іншої, сучаснішої, а також розглянуто процеси модифікації та модернізації відкритих структур і систем даних [10].

Проблему PI щодо ПС було розглянуто автором у [11], спираючись на оцінки ризиків [12]. Методологічні засади PI ПС було закладено у [13], спираючись на моделі Бома (Boehm) [14, 15].

Методологія, що викладена у [16] стала у нагоді у якості базової траєкторії досліджень. Принципи, розглянуті у [17] та дослідження у [18], підказали практичні аспекти оцінки проектів [19], що підлягають PI, наприклад ПС з [20, 21].

Виділення частин загальної проблеми. Мета роботи

З комерційної точки зору, реінжиніринг (PI) часто вважають єдиним способом збереження успадкованих модулів у експлуатації ПС, у той час як нову розробку – не рекомендується розглядати не тільки з точки зору дефіциту часу, що вже було затрачено на первинну розробку, та як наслідок – збільшення економічних витрат, а й з точки зору ризику виникнення структурних помилок.

У цей же час PI дає змогу виконати еволюціонування ПС шляхом позитивних змін видів її

забезпечення з метою підвищення зручності її експлуатації та супроводу.

Звісно, що не будь-яку ПС можна піддати PI: іноді зміни у програмному кодї тягнуть більші витрати ніж нова розробка, оскільки необхідно не тільки знайти спеціалістів у визначеній мові програмування, але й оплатити їхній час на аналіз та розібрання старої версії ПС, що цікавить.

Таким чином, у якості вхідних даних розв'язуваної задачі уводяться:

- вартість перепроєктування ПС із уведеним обмеженням, а саме: мінімізація вартості;

- швидкість уведення до експлуатації оновленої ПС із уведеним обмеженням на затягування, тобто максимізація швидкості;

- виявлення помилок у застарілій версії ПС, які необхідно намагались виключати у оновленій, тобто: мінімізація помилок.

У зв'язку з цим, вихідними даними, що будуть являти собою бажані результати розв'язання задачі цієї статті – формалізація критеріїв рентабельності PI ПС, за якими після моделювання визначених порівняльних характеристик буде прийматись однозначне рішення щодо застосування або відхилення PI.

Проте, необхідно зауважити, що ні в якому разі не можна остаточно оцінювати порівняльні характеристики тільки за критеріями, що охоплюють лише аналітику програмного коду реалізації ПС. Критерії оцінювання повинні базуватися на всебічному аналізі усіх видів забезпечення ПС, що підлягає удосконаленню.

Таким чином, на підставі викладення загальної проблеми, можна сформулювати мету роботи: сформулювати метод представлення проектної оцінки реінжинірингу ПС, на підставі якого приймається остаточне рішення щодо доцільності виконання реінжинірингу.

Матеріали та методи

Метод, що закладений у мету роботи, є розвитком запропонованої Бомом (Boehm) методології [12, 14 – 16], що знайшла своє продовження у аналітичних моделях [22] із запропонованими змінами та пропозиціями щодо гнучкості побудови на етапі передпроектного аналізу.

Теоретичні відомості з матеріалів досліджень [1, 10, 14 – 16] для практичного застосування потребують розширення керованості при моделюванні, для чого необхідно увести до моделі проектні коефіцієнти, що є предметом роботи.

Ці коефіцієнти являють собою удосконалення уявлень оцінки проектів за методом проектних точок Карнера (Kagner) [4] та методом констант Якобсона (Jacobson) [9] із доданими доповненнями та розширеннями, що відображені у методі розрахунку показників [5].

Проте, у методі [5] не міститься механізму візуалізованого подання оцінки PI ПС, що присутні у

моделях з дослідження [22]. Елементи такої візуалізації також спираються на методи побудови спіральних моделей Архімеда [23], векторного представлення годографів Гамільтона та Михайлова, а облік запланованого часу базується на проектних діаграмах Ганта.

Сформований у статті метод проектних коефіцієнтів ґрунтується на розширеному математичному відтворенні ідеалізованих моделей, що представлені у роботі [22]. Математичні основи цього методу сформовано на підставі узагальнення матеріалів з [22].

Результати досліджень

Оскільки модель РІ ПС та схема її побудови отримані автором цієї статті у [22], то відразу перейдемо до уведення проектних коефіцієнтів до моделі та побудови годографів РІ.

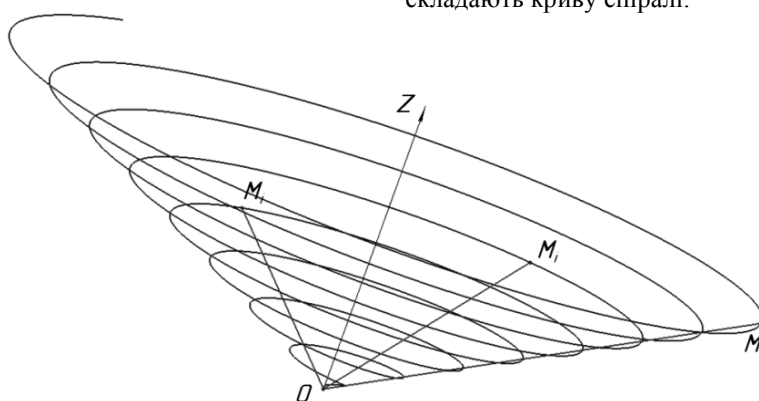


Рис. 1. Годограф з високим коефіцієнтом автоматизації РІ ПС: O – полюс реінжинірингу; Z – кількість ідентифікованих програмних компонентів (у лінійному масштабі) або рядків програмного коду (у логарифмічному масштабі); OZ – вісь компонентів (апліката)

Навпаки, якщо РІ відбувається, значною мірою, у ручному режимі, то форма спіралі – витягнута (рис. 2), що показує здороження вартості за рахунок вагомій складовій інтелектуальної праці програмістів:

$$\text{if } \delta \rightarrow \min \therefore OM_{i,j} \rightarrow \max; \quad (3)$$

$$\lim_{\delta \rightarrow \min} \langle OM_{i,j}(\delta) \rangle = \max. \quad (4)$$

2. Коефіцієнт схожості компонентів. Цей коефіцієнт залежить від ступеня впровадження КПВ. Для проведення порівняльного експерименту, візьмемо годограф, що представлено у [22, рис. 3]: для нього коефіцієнт схожості компонентів (θ) буде базовим.

Чим більше в процесі РІ застосовуються КПВ, тим густіше розміщуються витки спіралі (n) (рис. 3). З представлення годографа бачимо, що:

$$\text{if } \theta \rightarrow \max \therefore OM_{i,j} \rightarrow \min \left\{ M_{i,j} \mid \theta \equiv M_{i,j} \mid \theta_{\max} \right\}; \quad (5)$$

$$\lim_{\theta \rightarrow \max} \langle OM_{i,j}(\theta) \rangle = \min. \quad (6)$$

1. Коефіцієнт автоматизації РІ. Чим більше процес РІ автоматизований, тим більш плоскої форми набирає модель (рис. 1), а це значить, що вектор витрат (OM_i) із зростанням кількості компонентів програмного коду (i) чи верифікованих рядків програмного коду (j) зростає не значно, а витрачений час на виконання РІ t_i – визначається машинним часом. Іншими словами – якщо узяти ПС із визначеною фіксованою i , у яку помістимо довільну точку площини (M_i), та збільшимо у моделі коефіцієнт автоматизації реінжинірингу (δ), то OM_i буде зменшуватись:

$$\text{if } \delta \rightarrow \max \therefore OM_{i,j} \rightarrow \min; \quad (1)$$

$$\lim_{\delta \rightarrow \max} \langle OM_{i,j}(\delta) \rangle = \min, \quad (2)$$

при чому $M_{i,j}$ – побудована послідовність точок, які складають криву спіралі.

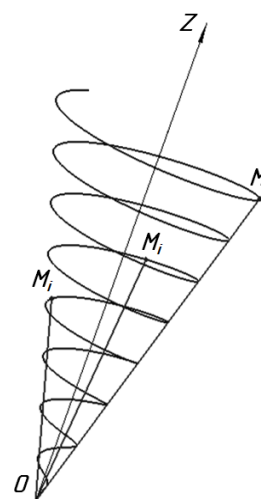


Рис. 2. Годограф із низьким коефіцієнтом автоматизації РІ ПС

Тобто при збільшенні θ – зменшується вектор витрат OM_i при однаково узятих M_i , відповідно до випадків з θ та θ_{\max} та навпаки: при використанні унікальних програмних компонентів – витки спіралі

розміщуються нещільно (рис. 4), тобто збільшується OM_i при однаково узятих M_i , відповідно до випадків з θ та θ_{\min} :

$$\text{if } \theta \rightarrow \min \therefore OM_{i,j} \rightarrow \max \{M_{i,j} | \theta \equiv M_{i,j} | \theta_{\min}\}; \quad (7)$$

$$\lim_{\theta \rightarrow \min} \langle OM_{i,j}(\theta) \rangle = \max. \quad (8)$$

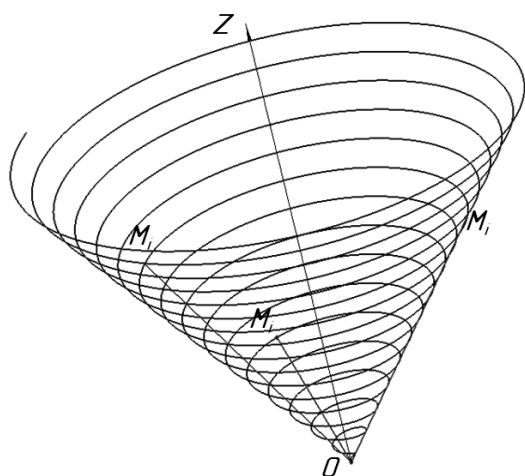


Рис. 3. Годограф РІ ПС із високим коефіцієнтом схожості компонентів

Експеримент із побудови годографів РІ проводився із дванадцятьма ПС, деякі з яких узяті з [20, 21]. За домовленістю, виконавцями проектів, було надано початкові умови щодо кількості рядків програмного коду та фактичного часу повного виконання ними РІ ПС, який було зафіксовано у проектних діаграмах Ганта. Також цей час було ретельно перевірено за протоколами виконання та актами виконаних робіт.

Таблиця 1. Час виконання РІ ПС

Номер проекту	Час, год.		
	Модельний за ІМР	Модельний із проектними коефіцієнтами	Фактичний
1	157	174	185
2	247	187,5	190
3	70	66,5	58,5
4	432,5	575	606
5	104,5	108,5	102,5
6	807	726	623,5
7	565	686	730,5
8	118	129	134
9	226	229,5	270
10	90	84	76,5
11	789	811	878
12	652	516	511

За даними табл. 1 побудована діаграма (рис. 5) порівняння фактичного часу (actualtime) виконання РІ ПС та змодельованих часів: за ІМР (idea lized model) та із уведеними проектними коефіцієнтами (with project coefficients).

Статистичні результати порівняння зведено до табл. 2.

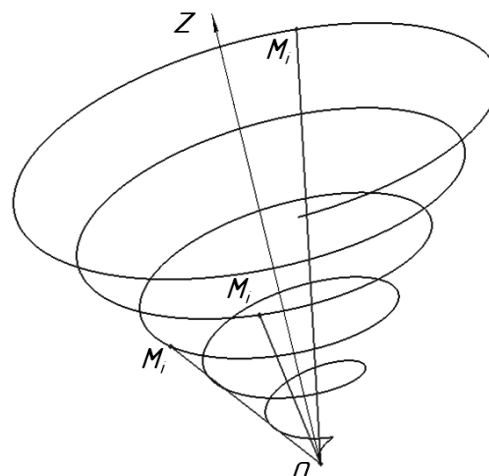


Рис. 4. Годограф РІ ПС із низьким коефіцієнтом схожості компонентів

За отриманими моделями було побудовано до графі РІ ПС, числовий матеріал яких подано у табл. 1. У таблиці наведено результати моделювання без використання проектних коефіцієнтів – за ідеалізованими моделями реінжинірингу (ІМР), отриманою у [22], та із уведенням описаних у цій статті коефіцієнтів. При виконанні проектів обчислення вимірювання часу відбувається з округленням до півгодини.

Обговорення

При аналізі та обговоренні результатів статті, можна побачити, що метод проектних коефіцієнтів, виходячи з табл. 1 та рис. 5, має значно меншу розбіжність з фактичним часом виконання РІ ПС ніж оцінка часу за ІМР.

Таблиця 2. Розбіжність порівняння за часом РІ ПС

Номер проекту	Оцінка за ІМР		Оцінка із проектними коефіцієнтами	
	Абсолютна похибка, год.	Відносна похибка, %	Абсолютна похибка, год.	Відносна похибка, %
1	28	15,1	11	5,9
2	57	30,0	2,5	1,3
3	11,5	19,7	8	13,7
4	173,5	28,6	31	5,1
5	2	2,0	6	5,9
6	183,5	29,4	102,5	16,4
7	165,5	22,7	44,5	6,1
8	16	11,9	5	3,7
9	44	16,3	40,5	15,0
10	13,5	17,6	7,5	9,8
11	89	10,1	67	7,6
12	141	27,6	5	1,0
Середнє значення, %:		19,3	7,6	

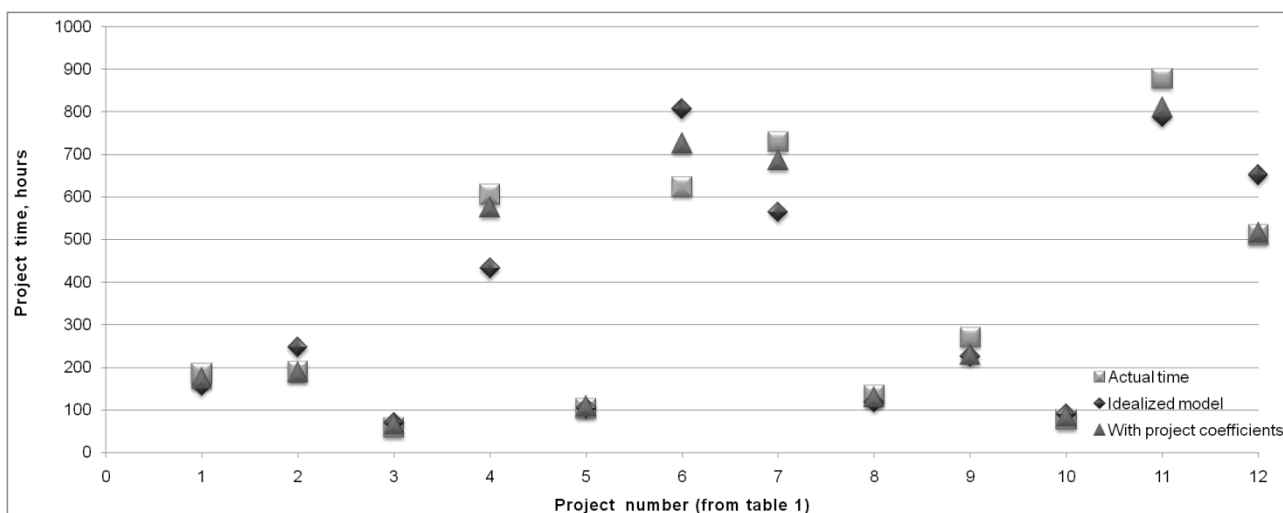


Рис. 5. Діаграма порівняння фактичного часу та змодельованих часів виконання РІ ПС

Максимальне значення відносної похибки оцінки РІ ПС за методом проектних коефіцієнтів не перевищує 16,4 % (табл. 2), проти 30 % при оцінці за ІМР. Середнє значення відносної похибки 7,6 % за методом проектних коефіцієнтів, проти 19,3 % – за ІМР.

Для справедливості, необхідно обговорити, що п'ятий проект (табл. 1, 2) має оцінку ближчу до фактичної за ІМР ніж за методом проектних коефіцієнтів (104,5 год. проти 108,5 год. при фактичному часі 102,5 год.) та відповідно меншу відносну похибку оцінки (2% проти 5,9 %).

На думку автора, це пов'язано із помилковими даними (заниженням приблизно на 6 – 8 год.) фактичного часу виконання проекту, який було надано розробниками, проте автори працювали із вже готовими показниками виконання у вигляді діаграм Ганта та не проводили корегування цих годин.

Що ж до наведених графічних побудов годографів РІ (рис. 1 – 4), то можна зробити висновки, що поверхня, для якою годограф є направляючою, являє собою лінійчатую конічну поверхню (ЛКП), що

може бути зворотно відновлена рухом вектора OM_i , як утворюючої.

Крім того, у подальших дослідженнях варто проаналізувати отримані ЛКП на предмет практичного застосування наступної висунутої гіпотези: якщо обчислити площу ЛКП, то можна дістати кількісну характеристику виконаної роботи щодо здійснення РІ ПС:

$$S = \int_0^{\phi(n+1)} \Phi(\rho, \phi) d\phi, \quad (9)$$

де S – площа лінійчатої конічної поверхні; ϕ – час, протягом якого відбувається реінжиніринг; ρ – радіус-вектор витрат; Φ – функція конічної спіралі.

Якщо обчислити об'єм ЛКП, то – визначити обсяг витрачених ресурсів:

$$V = \int_0^Z \Phi(\rho, \phi, Z) dZ, \quad (10)$$

де V – об'єм лінійчатої конічної поверхні.

Висновки та перспективи подальшого розвитку

Головна відмінність між РІ і новою розробкою полягає в тому, що процес перепроектування ПС починається не з "нуля", а з розгляду можливостей успадкування позитивних якостей старої системи.

РІ виконується за допомогою комплексу засобів, у тому числі за рахунок застосування КПВ та Computer-Aided Software Engineering (CASE) систем. Згідно з оптимістичними прогнозами застосування КПВ здатне у 4 рази знизити вартість ПС у порівнянні з новою розробкою. Проте заради об'єктивності слід помітити, що у деяких випадках, все ж таки раціональніше застосувати нову розробку.

Для того, щоб прийняти мотивоване рішення щодо застосування або відмови від РІ, згідно з постановкою задачі, у статті було представлено метод проектної оцінки РІ ПС, на підставі якого формується остаточний висновок щодо доцільності виконання РІ.

Згідно з цим методом до моделей РІ ПС, що були сформовані у [22], уводяться проектні коефіцієнти, що дозволяють підвищити точність оцінки РІ, яка виражена у зменшенні середнього значення відносно похибки щодо фактичного часу виконання РІ з 19,3 % (при побудові за ІМР) до 7,6 %.

Таким чином, до результатів наукової новизни слід віднести удосконалення моделі візуалізації витрат на РІ ПС за рахунок уведення коефіцієнтів автоматизації та схожості компонентів, які дозволить вносити зміни до конфігурації відповідних годографів РІ.

Подальші наукові дослідження у цій області включають у себе всебічне вивчення висунутої авторами гіпотези щодо представлення кількісних характеристик виконаної роботи з РІ ПС та визначення обсягів витрачених ресурсів у формі обчислення площин та об'ємів ЛКП, що утворюються годографами РІ.

Також до перспектив дослідження за темою статті, вже після прийняття рішення щодо застосування РІ, слід віднести задачу розробки логічної схеми РІ, що є похідною від логічної схеми проектування, яка починається цільовою програмою та включає в себе алгоритмічний вибір компонентів

РІ. Ця логічна схема представляє цільовий засіб отримання нової ПС шляхом виконання послідовності операцій внесення змін, модернізації або модифікації, а також перепрограмування окремих її компонентів. Логічна схема РІ повинна починатися цільовою програмою, об'єднуючи зовнішню та внутрішню частину, відповідно: формування цілей та моделювання.

Отже РІ ПС дозволить подолати: протиріччя між темпами розвитку науки, техніки й процесів проектування; підвищити ефективність технічного супроводу ПС та скоротити експлуатаційні витрати.

Сформовані у статті результати, з наукової точки зору, увійдуть до основ методології РІ ПС, а з практичної – стануть у нагоді системним програмістам, які задіяні у перепроектуванні й переробці ПС, а також системним архітекторам, які працюють із мультимовними надбудовами ПС, що вже знаходяться у кількарічній експлуатації і набувають еволюційного розвитку із плином часу та удосконалення у процесі використання.

Подяки

Перш за все, хочу висловити подяку доктору технічних наук зі спеціальності 05.12.13 – системи автоматизації проектувальних робіт, професору кафедри системотехніки Харківського національного університету радіоелектроніки – Безкоровайному Володимирі Валентиновичу за первинну позитивну оцінку ідеї створення ІМР, що було йому викладено олівцем у вигляді ескізу на аркуші зошита у 2012 році. Після його настанов та консультацій, у процесі 7-ми річної роботи, було отримано наукові результати, що наведено у поданій статті.

Також автор статті висловлює подяку Одеському державному екологічному університету, як базовій організації, у якій проводяться дослідження із поданої тематики, за сприяння у фінансуванні та організаційній підтримці: наукових досліджень; наукового спілкування із галузевими спеціалістами у формі відрядження автора до участі у конференціях та форумах; випуску двох монографій за розглянутою тематикою коштами університету.

Список літератури

1. Blum B. Software engineering: a holistic view. URL: <https://dl.acm.org/citation.cfm?id=SERIES9569.128915> (дата звернення : 23.02.2019).
2. Cohn M. Agile Estimating and Planning. NY : Prentice Hall, 2005. 368 p.
3. Klein M. Reengineering methodologies and tools. A Prescription for Enhancing Success. URL: <https://www.tandfonline.com/doi/abs/10.1080/10580539408964633> (дата звернення : 23.02.2019). DOI: <https://doi.org/10.1080/10580539408964633>.
4. Karner G. Resource Estimation for Objectory Projects: project report. SF AB: Objective Systems, 1993. 9 p.
5. Великодний С. С., Тимофеева О. С., Зайцева-Великодна С. С. Метод розрахунку показників оцінки проекту при виконанні реінжинірингу програмних систем. *Радіоелектроніка, інформатика, управління*. 2018. № 4. С. 135–142. DOI: <https://doi.org/10.15588/1607-3274-2018-4-13>.
6. Hammer M., Champy J. Reengineering the corporation: A manifesto for business revolution. URL: <https://www.sciencedirect.com/science/article/pii/S0007681305800643?via%3Dihub> (дата звернення : 23.02.2019). DOI: [https://doi.org/10.1016/S0007-6813\(05\)80064-3](https://doi.org/10.1016/S0007-6813(05)80064-3).
7. Grover V., Malhotra M. Business process reengineering: A tutorial on the concept, evolution, method, technology and application. URL : <https://www.sciencedirect.com/science/article/abs/pii/S0272696396001040> (дата звернення : 23.02.2019). DOI: [https://doi.org/10.1016/S0272-6963\(96\)00104-0](https://doi.org/10.1016/S0272-6963(96)00104-0).

8. Manganelli R., Klein M. The Reengineering Handbook: A Step-by-Step Guide to Business Transformation. URL : https://www.researchgate.net/publication/304544531_The_Reengineering_Handbook_A_Step-by-Step_Guide_to_Business_Transformation (дата звернення : 23.02.2019). DOI: <https://doi.org/10.1097/01445442-199503000-00011>.
9. Jacobson I., Ericsson M., Jacobson A. The Object Advantage: Business Process Reengineering with Object Technology. ACM Press. URL: <http://eaststemcell.com/files/storage.cloud.php?id=MDIwMTQyMjg5MQ==> (дата звернення : 23.02.2019).
10. O'Reilly T. Open Source Paradigm Shift by Tim O'Reilly. URL : http://archive.oreilly.com/pub/a/oreilly/tim/articles/paradigmshift_0504.html (дата звернення : 23.02.2019).
11. Великодний С. С. Проблема реінжиніринга видів забезпечення систем автоматизованого проектування. *Управляючі системи і машини*. 2014. № 1. С. 57–61, 76.
12. Boehm B. Software Risk Management. URL: https://link.springer.com/chapter/10.1007%2F3-540-51635-2_29 (дата звернення : 23.02.2019). DOI: https://doi.org/10.1007/3-540-51635-2_29
13. Великодний С. С. Методологічні основи реінжиніринга систем автоматизованого проектування. *Управляючі системи і машини*. 2014. № 2. С. 39–43.
14. Boehm B. Spiral Development: Experience, Principles and Refinements: special Report. CMU. SEI-2000-SR-008. 2000. 37 p.
15. Boehm B. A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes. 1986. Vol. 11, Issue. 4. P. 14–24. DOI: <https://doi.org/10.1145/12944.12948>.
16. Selby R. W. Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management and Research. New Jersey : John Wiley & Sons, 2017. 818 p.
17. Anda B. Effort Estimation of Use Cases for Incremental Large-Scale Software Development. 27-th International Conference on Software Engineering, 15–22 may, 2015 : proceedings. St. Louis : MO, 2015. P. 303–311.
18. Carroll E. R. Estimating Software Based on Use Case Point. OOPSLA '05: Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications: proceedings. San Diego : CA, 2005. P. 257–265. DOI: <https://doi.org/10.1145/1094855.1094960>.
19. Clemmons R. Project Estimation with Use Case Points. *Cross Talk*. 2016. Vol. 2, Issue February. P. 18–22.
20. Невлюдов И. Ш., Великодний С. С., Омаров М. А. Использование CAD/CAM/CAE/CAPP при формировании управляющих программ для станков с ЧП. *Восточно-Европейский журнал передовых технологий*. 2010. Т. 2. Вып. 2 (44). С. 37–44.
21. Великодний С. С. Реінжиніринг систем моніторингу та дистанційного управління судовими енергетичними установками. XXII Міжн. конф. з автом. управл. "Автоматика 2015", 10–11 вер., 2015. Одеса. С. 133–134.
22. Великодний С. С. Ідеалізовані моделі реінжинірингу програмних систем. *Радіоелектроніка, інформатика, управління*. 2019. № 1.

References

1. Blum, B. "Software engineering: a holistic view", available at : <https://dl.acm.org/citation.cfm?id=SERIES9569.128915> (last accessed 23.02.2019).
2. Cohn, M. (2005). *Agile Estimating and Planning*, Prentice Hall, NY, 368 p.
3. Klein, M. "Reengineering methodologies and tools. A Prescription for Enhancing Success", available at : <https://www.tandfonline.com/doi/abs/10.1080/10580539408964633> (last accessed 23.02.2019). DOI: <https://doi.org/10.1080/10580539408964633>.
4. Karner, G. (1993), *Resource Estimation for Objectory Projects* : project report, Objective Systems, SF AB, 9 p.
5. Velykodniy, S. S., Tymofeieva, O. S., Zaitseva-Velykodna, S. S. (2018), "The calculation method for indicators project estimation in the implementation of software systems re-engineering" ["Metod rozrakhunku pokaznykiv otsinky proektu pry vykonanni reinzhyrnyhu prohramnykh system"], *Radio Electronics, Computer Science, Control*, No. 4, P. 135–142. DOI: <https://doi.org/10.15588/1607-3274-2018-4-13>.
6. Hammer, M., Champy, J. "Reengineering the corporation: A manifesto for business revolution", available at : <https://www.sciencedirect.com/science/article/pii/S0007681305800643?via%3Dihub> (last accessed 23.02.2019). DOI: [https://doi.org/10.1016/S0007-6813\(05\)80064-3](https://doi.org/10.1016/S0007-6813(05)80064-3).
7. Grover, V., Malhotra M., "Business process reengineering: A tutorial on the concept, evolution, method, technology and application", available at : <https://www.sciencedirect.com/science/article/abs/pii/S0272696396001040> (last accessed 23.02.2019). DOI: [https://doi.org/10.1016/S0272-6963\(96\)00104-0](https://doi.org/10.1016/S0272-6963(96)00104-0).
8. Manganelli, R., Klein, M. "The Reengineering Handbook: A Step-by-Step Guide to Business Transformation", available at : https://www.researchgate.net/publication/304544531_The_Reengineering_Handbook_A_Step-by-Step_Guide_to_Business_Transformation (last accessed 23.02.2019). DOI: <https://doi.org/10.1097/01445442-199503000-00011>.
9. Jacobson, I., Ericsson, M., Jacobson, A. "The Object Advantage: Business Process Reengineering with Object Technology. ACM Press", available at : <http://eaststemcell.com/files/storage.cloud.php?id=MDIwMTQyMjg5MQ==> (last accessed 23.02.2019).
10. O'Reilly, T. "Open Source Paradigm Shift by Tim O'Reilly", available at: http://archive.oreilly.com/pub/a/oreilly/tim/articles/paradigmshift_0504.html (last accessed 23.02.2019).
11. Velykodniy, S. S. (2014), "The reengineering problem of ensures types CAD/CAM/CAE-systems" ["Problema reinzhyrnyhu vidov obespecheniya sistem avtomatizirovannogo proektirovaniya"], *Control Systems and Computers*, No. 1, P. 57–61, 76.
12. Boehm, B. "Software Risk Management", available at : https://link.springer.com/chapter/10.1007%2F3-540-51635-2_29 (last accessed 23.02.2019). DOI: https://doi.org/10.1007/3-540-51635-2_29.
13. Velykodniy, S. S. (2014), "The methodological bases of reengineering CAD/CAM/CAE-systems" ["Metodologicheskie osnovy reinzhyrnyhu sistem avtomatizirovannogo proektirovaniya"], *Control Systems and Computers*, No. 2, P. 39–43.
14. Boehm, B. (2000), *Spiral Development: Experience, Principles and Refinements* : special Report, CMU, SEI-2000-SR-008, 37 p.
15. Boehm, B. A (1986), "Spiral Model of Software Development and Enhancement", *ACM SIGSOFT Software Engineering Notes*, Vol. 11, Issry 4, P. 14–24. DOI: <https://doi.org/10.1145/12944.12948>.
16. Selby, R. W. (2017), *Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management and Research*, John Wiley & Sons, New Jersey, 818 p.

17. Anda, B. (2015), "Effort Estimation of Use Cases for Incremental Large-Scale Software Development", *27-th International Conference on Software Engineering, 15–22 may : proceedings, MO, St. Louis, P.* 303–311.
18. Carroll, E. R. (2005), "Estimating Software Based on Use Case Point", *OOPSLA '05 : Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications: proceedings, CA, San Diego, P.* 257–265. DOI: <https://doi.org/10.1145/1094855.1094960>.
19. Clemmons, R. (2016) "Project Estimation with Use Case Points", *Cross Talk*, Vol. 2, Issue February, P. 18–22.
20. Nevlyudov, I. Sh., Velykodniy, S. S., Omarov, M. A. "Using CAD / CAM / CAE / CAPP when forming control programs for CNC machines" ["Ispol'zovanie CAD/CAM/CAE/CAPP pri formirovaniy upravlyayushchikh programm dlya stankov s ChPU"], *Eastern-European Journal of Enterprise Technologies*, Vol. 2, Issue 2 (44), P. 37–44.
21. Velykodniy, S (2015), "Reengineering of SCADA-systems by shipping energy plants ["Reinzhiniring sistem monitoringu ta distantsiyogo upravlinnya sudnovimi energetichnimi ustanovkami"]", *22th International Conference "Automatic 2015", 10–11 sep. : proceedings, Odessa, P.* 133–134.
22. Velykodniy, S. S. (2019), "The idealized models of software systems reengineering" ["Idealizovani modeli reinzhiniryntu prohrannykh system"], *Radio Electronics, Computer Science, Control*, No. 1.

Надійшла (Received) 24.02.2019

Відомості про авторів / Сведения об авторах / About the Authors

Великодний Станіслав Сергійович – кандидат технічних наук, доцент, Одеський державний екологічний університет, доцент кафедри інформаційних технологій, Одеса, Україна; e-mail: velykodniy@gmail.com; ORCID ID: <https://orcid.org/0000-0001-8590-7610>.

Великодний Станіслав Сергеевич – кандидат технических наук, доцент, Одесский государственный экологический университет, доцент кафедры информационных технологий, Одесса, Украина.

Velykodniy Stanislav – PhD (Computer Science), Associate Professor, Odessa State Environmental University, Associate Professor at the Department of Information Technologies, Odessa, Ukraine.

МЕТОД ПРЕДСТАВЛЕНИЯ ОЦЕНКИ РЕИНЖИНИРИНГА ПРОГРАММНЫХ СИСТЕМ ПРИ ПОМОЩИ ПРОЕКТНЫХ КОЭФФИЦИЕНТОВ

Предметом работы являются проектные коэффициенты, которые вводятся в модель реинжиниринга программной системы. **Цель** работы. Сформировать метод представления проектной оценки реинжиниринга программных систем, на основании которого принимается окончательное решение о целесообразности выполнения реинжиниринга. Одной из главных задач инженерии программного обеспечения является создание теоретических и прикладных основ быстрого и качественного построения сложных систем из более простых программных элементов, выполненных на современных языках программирования. Фактически решение этой задачи осуществляется путем сбора, объединения или интеграции разнородных программных ресурсов и компонентов повторного использования, включающих модули, библиотеки и программные реализации некоторой сложной программной системы. По современным мировым тенденциям проектирования программного обеспечения: программные системы должны быть развивающимися и эволюционировать. **Задачей** статьи является исследование влияния проектных коэффициентов, которые вводятся в модель реинжиниринга программной системы, на соответствие фактическим статистическим данным уже реально выполненных проектов по реинжинирингу программных систем. **Метод.** В статье нашла свое развитие методология Бома, продолжение которой отражено в формировании аналитических моделей с предложенными изменениями относительно гибкости построения на этапе предпроектного анализа. В модели введены проектные коэффициенты, представляющие собой усовершенствование представлений оценки проектов по методу проектных точек Карнера и методу констант Якобсона с приложенными дополнениями и расширениями. Визуализация моделей опирается на методы построения спиральных моделей Архимеда, векторного представления годографов Гамильтона и Михайлова, а метод учета запланированного времени базируется на проектных диаграммах Ганта. **Результаты.** В статье получен метод, в соответствии с которым к ранее сформированным моделям реинжиниринга программных систем, введены проектные коэффициенты, позволяющие повысить точность оценки реинжиниринга, выраженной в уменьшении среднего значения относительной погрешности фактического времени его выполнения. **Выводы.** К научной новизне относятся следующие положения касательно реинжиниринга программных систем: усовершенствованы модели визуализации расходов за счет введения коэффициентов автоматизации и сходства компонентов, позволяющих вносить изменения в конфигурации соответствующих годографов. Реинжиниринг программных систем позволит: преодолеть противоречия между темпами развития науки, техники и процессов проектирования; повысить эффективность технического сопровождения; сократить эксплуатационные расходы.

Ключевые слова: программная система; реинжиниринг; предпроектные исследования; проектирование; эволюционная спираль; годограф; проектные коэффициенты; коэффициент автоматизации; коэффициент сходства компонентов; компоненты повторного использования.

METHOD OF PRESENTING THE ASSESSMENT FOR REENGINEERING OF SOFTWARE SYSTEMS WITH THE PROJECT COEFFICIENTS HELP

The **subject** of the work are design coefficients that are introduced into the reengineering model of a software system. The **goal** of the work. Form a method for presenting a project assessment of software system reengineering, according to which a final decision is made about the feasibility of reengineering. One of the main task of software engineering is the creation of theoretical and applied foundations for the rapid and high-quality construction of complex systems from simpler software elements made in modern

programming languages. In fact, this task is solved by collecting, combining or integrating disparate software resources and reusable components, including modules, libraries and software implementations of some complex software system. According to modern world trends of software design: software systems must be continually developing and evolving. The **task** of the article is to study the influence of project coefficients, which are introduced into the model of the reengineering of a software system, in accordance with the actual statistical data of the actually completed projects on the reengineering of software systems. **Method.** The article has developed the Boehm's methodology, the continuation of which is reflected in the formation of analytical models with the proposed changes regarding the flexibility of construction at the stage of pre-project analysis. The design factors are introduced in the model, which are representing an improvement in the representations of project appraisals using the Karner's design points method and the Jacobson's constant method with applied additions and extensions. The visualization of the models is based on the methods of constructing the spiral models of Archimedes, the vector representation of Hamilton's and Mikhailov's hodographs, and the method of accounting for the planned time is based on the Gantt's project diagrams. **Results.** The article obtained a method according to which the previously formed models of software systems reengineering, introduced design coefficients, allowing to increase the accuracy of reengineering estimates, expressed in reducing the average value of the relative error of the actual time of its implementation. **Conclusions.** Scientific novelties include the following provisions regarding the reengineering of software systems: models of cost visualization were improved by introducing automation coefficients and similarity of components, allowing to make changes in the configuration of the relevant hodographs. Reengineering of software systems will allow: to overcome the contradictions between the pace of development of science, technology and design processes; improve the efficiency of technical support; reduce operating costs.

Keywords: software system; reengineering; pre-design studies; design; evolutionary spiral; hodograph; project coefficients; automation coefficient; coefficient of components similarity; reuse components.
