

ІДЕАЛІЗОВАНІ МОДЕЛІ РЕІНЖІНІРИНГУ ПРОГРАМНИХ СИСТЕМ

Великодний С. С. – канд. техн. наук, доцент, завідувач кафедри інформаційних технологій Одеської національної академії зв'язку ім. О. С. Попова; доцент кафедри інформаційних технологій Одеського державного екологічного університету, Одеса, Україна.

АНОТАЦІЯ

Актуальність. Програмні системи застосовуються у різноманітних галузях життя й діяльності людини, але найбільше поширення вони отримали у галузях, де необхідною є робота із багатьма рутинними операціями – це стосується промисловості, виробництва, транспорту, навчання та інфокомунікацій. Звичайно, що у процесі експлуатації цих систем відбувається еволюційне старіння видів забезпечення. Така тенденція призводить до погіршення швидкісних, інформаційно-комунікаційних, графічних, часових та інших характеристик, аж до повної відмови програмної системи. Тематика статті спрямована у галузі проектування ергатичних систем та управління проектами та програмами з перепроєктування або удосконалення цих систем. Реінжиніринг дозволяє виконати еволюціонування системи, шляхом внесення позитивних змін до її структури з метою підвищення характеристик експлуатації та технічного супроводу.

Мета. Сформувані представлення оцінки параметрів витрат ресурсів на виконання реінжинірингу програмних систем за допомогою математичного інструментарію опису моделей проектування.

Метод. У ідею статті закладено метод Боєма (Boehm) та принципи побудови спіралі Архімеда. Авторські дослідження призвели до думки об'єднати ідеї побудови структур Боєма та Архімеда, адаптувати їх під методологію програмної інженерії та перенести до циліндричної системи координат. В основу отриманих моделей закладено спіральний принцип організації відліку.

Результати. У статті розглянуто процес створення та візуалізації математичних моделей стосовно до процесу реінжинірингу програмних систем. Операції із поданими моделями можуть відбуватися у проєкціях часу та витрат, у ізометричній проєкції програмних компонентів, у логарифмічній проєкції рядків програмного коду.

Висновки. Вперше сформовані ідеалізовані моделі реінжинірингу, що дозволять підвищити точність оцінки витрат з перепроєктування програмних систем. Удосконалено модель візуалізації витрат на перепроєктування програмних систем, яка дозволить підвищити ефективність прогнозування порядку витрат. Вперше отримано залежності побудови траєкторії графічної моделі витрат від уведених коефіцієнту автоматизації реінжинірингу та коефіцієнту схожості компонентів, які дозволять вносити зміни до конфігурації моделі. Запропоновані ідеалізовані моделі реінжинірингу видів забезпечення програмних систем являють собою еволюційні спіралі, які побудовані у циліндричній системі координат. Реінжиніринг, який буде виконано за допомогою розроблених ідеалізованих моделей реінжинірингу дозволить не тільки скоротити витрати на перепроєктування програмних систем, але й підвищити ефективність технічного супроводу, збільшити життєвий цикл програмних систем, що вже знаходяться у експлуатації та подолати протиріччя між швидкими темпами розвитку науки, техніки і процесів проектування нових програмних систем.

КЛЮЧОВІ СЛОВА: програмна система, реінжиніринг, система автоматизованого проектування, вид забезпечення, CASE-засіб, компонент програмного коду, ідеалізована модель, перепроєктування, еволюційна спіраль, радіус-вектор витрат.

АБРЕВІАТУРИ

CASE – Computer-Aided Software Engineering;
SCADA – Supervisory Control And Data Acquisition;
БД – база даних;
ІМР – ідеалізована модель реінжинірингу;
ІПС – інформаційно-пошукова система;
ІС – програмна система;
РІ – реінжиніринг;
САП – система автоматизованого проектування.

НОМЕНКЛАТУРА

i – ідентифікований програмний компонент;
 j – верифікований рядок програмного коду;
 M – довільна точка площини;
 $M_{i \vee j}$ – побудована послідовність точок, які складають криву спіралі;
 M_x – прямокутна координата (проєкція на вісь OX);
 M_y – прямокутна координата (проєкція на вісь OY);
 n – виток спіралі;
 O – полюс реінжинірингу;
 OA – додатній відрізок (одиниця витрат);
 OM_i – вектор витрат;

OX – вісь витрат (абсциса);
 OY – вісь витрат (ордината);
 OZ – вісь компонентів (апліката);
 t_i – проєкція часу реінжинірингу;
 Z – кількість ідентифікованих програмних компонентів (у лінійному масштабі) або рядків програмного коду (у логарифмічному масштабі);
 δ – коефіцієнт автоматизації реінжинірингу;
 ε – верхня межа граничних витрат;
 θ – коефіцієнт схожості компонентів;
 ρ – радіус-вектор витрат;
 Φ – функція конічної спіралі;
 φ – час, протягом якого відбувається реінжиніринг.

ВСТУП

ІС застосовуються у різноманітних галузях життя й діяльності людини, але найбільше поширення вони отримали у галузях, де необхідною є робота із багатьма рутинними операціями, великим обсягом інформації, яку необхідно одночасно оброблювати, змінювати

та доповнювати. Це стосується промисловості, виробництва, транспорту, навчання та інфокомунікацій.

Експлуатація ПС, у кожній окремій галузі застосування (будівництво, телекомунікації та зв'язок, сфера послуг, освіта та ін.), має свої принципові відмінності. Як самостійний приклад, можна навести ергатичні системи, а саме: системи моніторингу та дистанційного управління (SCADA-системи). Їх призначено для спостереження та керування віддаленим об'єктом, який, до речі, може бути небезпечним для здоров'я оператора; крім того, за допомогою SCADA-систем виконується аналіз, накопичення та необхідне сортування робочих даних.

Об'єктом роботи є процес реінжинірингу програмних систем.

Предметом роботи – моделі, призначення яких представити виконання реінжинірингу програмних систем.

Мета роботи – сформувати представлення оцінки параметрів витрат ресурсів на виконання реінжинірингу програмних систем за допомогою математичного інструментарію опису моделей проектування.

1 ПОСТАНОВКА ЗАДАЧІ

Спільної рисою для усіх ПС залишається те, що під впливом часу та інших невід'ємних факторів інформатизації (оновлення: операційних систем, мов програмування, принципів дії розподілених систем обробки даних тощо) відбувається еволюційне старіння видів забезпечення ПС. Така тенденція призводить до погіршення швидкісних, інформаційно-комунікаційних, графічних, часових та інших характеристик, аж до повної відмови ПС.

З цього випливає, що ПС повинна бути такою, що розвивається. За світовими тенденціями ПС спирається на життєвий цикл у 3–4 роки [1]. Звісно, що при оновленні об'єкту – оновлюється й ПС, за допомогою якої об'єкт обслуговується. На цьому етапі виникає питання: що робити, коли ПС жорстко прив'язано до об'єкту експлуатації? Наприклад: суднова система моніторингу та дистанційного управління, що являє собою ПС, яку жорстко прив'язано до суднової енергетичної установки та до вимірювальних каналів, при цьому життєвий цикл судна складає 12–15 років.

Відповідь на це питання – одна: необхідно застосувати реінжиніринг щодо ПС.

При цьому архітектура системи може залишатися незмінною, тобто фактично виконується перепроєктування.

Виконання реінжинірингу (перепроєктування) ПС неможливе без використання сучасних САП. Розробка, впровадження та подальший розвиток елементів і підсистем САП у різних економіки країни – є однією з важливих задач із просування цієї галузі на сучасний рівень світового промислового виробництва.

Розроблені елементи і підсистеми САП дають значну ефективність в тому випадку, коли результати автоматизованого проектування використовуються у технічному виробництві. Ефективність, в даному випадку,

обумовлюється тим, що за сучасних темпів розвитку науки і техніки, виникає протиріччя між зростаючим рівнем науково-технічних досягнень та існуючими методами й технічними засобами проектування.

На подолання цих протиріч спрямована задача, яка вирішується у поданій статті: пропонуються ІМР, що здатні оцінити час та витрати, необхідні для виконання реінжинірингу ПС.

Підсистеми САП визначають зону діяльності в системі проектування з вирішенням всіх завдань, що пов'язані з розробкою проєктів для кожної з виділених груп об'єктів. Кожна підсистема САП поєднує в собі діяльність з проектування всіх об'єктів одного виду та відносяться до однієї певної області проектування.

2 ОГЛЯД ЛІТЕРАТУРИ

Корпорації, що розробляють САП – проєктують багато спеціалізованих програмних продуктів, наприклад – AutoDesk. В них є цілий комплект програм для роботи с машинобудівним профілем (Inventor), архітектурою (ArchiCad), дизайном (3dMAX) та проєктуванням у широкому сенсі (AutoCad) [2].

Завдяки потужним обчислювальним засобам у САП за допомогою ПС, що містять банки і БД готових проєктно-конструкторських рішень, можливо досить швидко внести корегування у необхідні параметри виробів (розміри, форма, порядок обробки тощо), що виготовляються; а також у послідовність технологічних операцій, тобто переорієнтувати увесь виробничий процес [3].

Подібне переорієнтування (у широкому сенсі) ПС та БД, можна визначити як реінжиніринг інформаційного забезпечення [4].

Реінжиніринг містить у собі процеси реорганізації і реструктуризації ПС [5], переведення окремих компонентів системи в іншу, сучаснішу мову програмування, а також процеси модифікації або модернізації структури і системи даних [6].

Проблему реінжинірингу ПС було розглянуто автором у [7]. Методологічні засади реінжинірингу було закладено у [8] та розширено стосовно до САП у [9].

Методологія, що викладена у [10, 11] стала у нагоді у якості базової траєкторії досліджень. Принципи та дослідження у [12], підказали практичні аспекти застосування ІМР.

3 МАТЕРІАЛИ І МЕТОДИ

Згідно з викладеною постановкою задачі, постає необхідність розробки ІМР кожного з видів забезпечення (технічне, математичне, інформаційне, програмне, лінгвістичне, методичне, організаційне, ергономічне тощо) ПС.

Початкова ідея створення ІМР була спрямована до побудови моделі у полярній системі координат. Побудова запропонованої ІМР відбувалася наступним чином: візьмемо у просторі довільну точку O (полюс), яку назвемо полюсом P_1 та побудуємо проміні OX та OY (рис. 1) – це будуть вісі витрат (у євро).

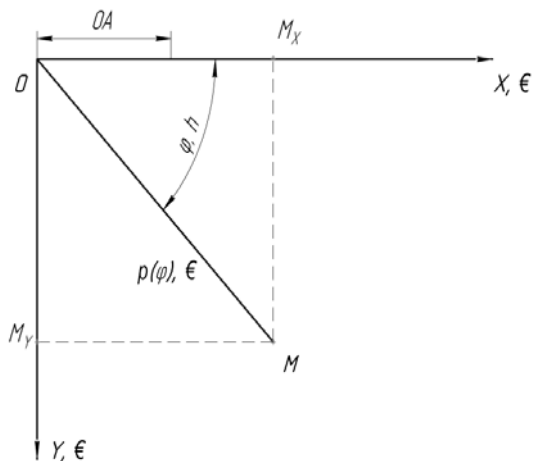


Рисунок 1 – IMP побудована у полярній системі координат

Нехай M – довільна точка площини, M_x та M_y – її прямокутні координати (проекції на вісі OX та OY). Вектор OM – радіус-вектор витрат (модуль якого збільшується). У подальшому позначимо цей радіус-вектор як ρ (рис. 1). Прийнемо будь-який додатній відрізок за одиницю витрат OA . Кут повороту φ – це час, протягом якого відбувається реінжиніринг (зростає за годинниковою стрілкою та, надалі, збільшується кількість повних обертів).

Таким чином, ρ та φ – полярні координати. Тоді:

$$M_x = \rho \cos \varphi; \quad (1)$$

$$M_y = \rho \sin \varphi; \quad (2)$$

а радіус-вектор витрат:

$$\rho = \sqrt{(M_x)^2 + (M_y)^2}. \quad (3)$$

Проте у полярній моделі було знайдено деякі недоліки, що значною мірою ускладнюють її практичне застосування. Ці недоліки винесено автором у розділ «Обговорення».

Метод Боема (Boehm) [10, 11] та дослідження [3, 4, 12], призвели до думки об'єднати ідеї побудови IMP у вигляді спіралі Архімеда та перенести її до циліндричної системи координат. В основу моделі закладено спіральний принцип організації відліку.

4 ЕКСПЕРИМЕНТИ

Початок побудови спіральної IMP дещо схожий із вище розглянутою початковою IMP. Однак, після задання нульової точки реінжинірингу наступають нові етапи:

– вісь OZ (рис. 2) – кількість ідентифікованих програмних компонентів (i) у лінійному масштабі або кількість верифікованих рядків програмного коду (j) у логарифмічному масштабі;

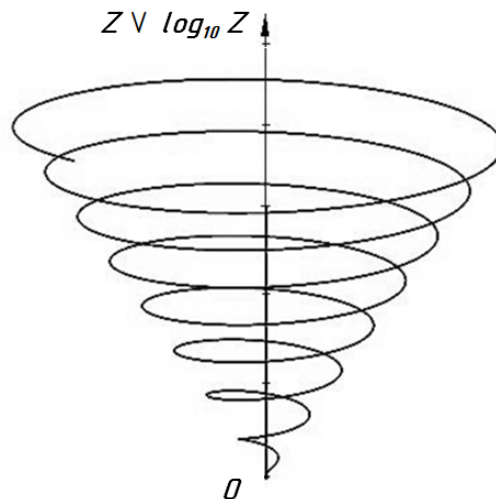


Рисунок 2 – Ескіз побудови спіральної IMP

– побудована послідовність точок M_{ivj} , які, власне, складають криву спіралі (рис. 3) – це компоненти програмного коду (для зручності, нижче, M_{ivj} будемо записувати як M_i ;

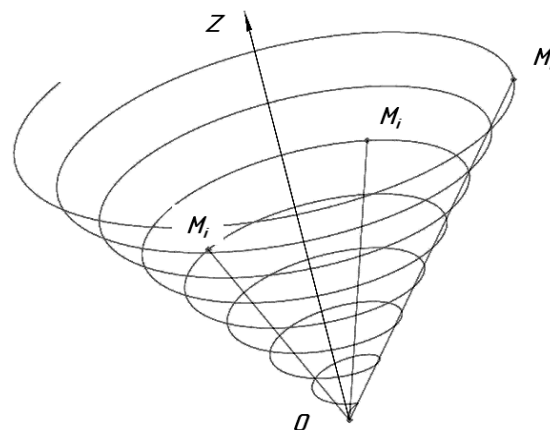


Рисунок 3 – Побудова послідовності точок M_i та вектору витрат OM_i

– побудуємо вектор витрат (OM_i) – він з'єднує полюс реінжинірингу (O) з поточною точкою спіралі M_i (рис. 3) та перевизначимо його як ρ :

$$\rho = OM_{ivj}. \quad (4)$$

Для експериментальної частини було обрано сім проектів з перекодування ПС, написаних однією мовою програмування до іншої. За домовленістю, виконавцями проектів, було надано початкові умови (кількість рядків програмного коду) та фактичний час повного виконання, який було зафіксовано у проектних діаграмах Ганта.

За цими даними були побудовані IMP, числовий матеріал яких подано у табл. 1. При виконанні проектів вимірювання часу відбувається з округленням до півгодини.

Таблиця 1 – Час виконання РІ проекту

Номер проекту	Час, год.	
	Модельний	Фактичний
1	157	185
2	247	190
3	70	58,5
4	432,5	606
5	104,5	102,5
6	807	623,5
7	565	730,5

5 РЕЗУЛЬТАТИ

Кут оберту φ між полюсом РІ та поточною точкою M_i спіралі (кут повороту вектору OM_i) – це час, який необхідно затратити на виконання РІ, причому, чим далі точка M_i від O , тим більше значення φ (рис. 4).

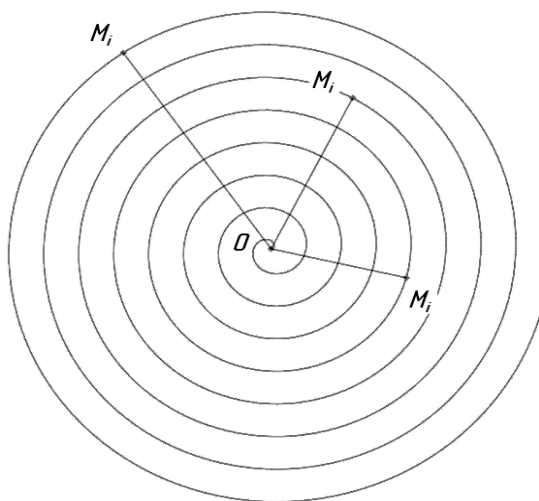


Рисунок 4 – Подання ІМР у проекції часу

Тобто кожний новий виток спіралі (n) додає 2π часу, а час тоді дорівнює:

$$t_i = \varphi_i(n+1), [\text{ум. од. часу}] \quad (5)$$

Проекція представлення ІМР може бути різною, наприклад, якщо подати проекцію уздовж вісі ідентифікованих програмних компонентів (OZ), як показано на рис. 4, то ІМР буде являти собою Архімедову спіраль, що, у нашому випадку, описується рівнянням:

$$\frac{\rho}{M_{i \vee j}} = \frac{t_{i \vee j}}{2\pi} \quad (6)$$

або

$$\rho = \xi t_{i \vee j}, \quad (7)$$

де

$$\xi = \frac{M_{i \vee j}}{2\pi}. \quad (8)$$

Загалом, конфігурація спіралі може бути різною та залежить від багатьох факторів, що закладено у математичній моделі:

$$\Phi(r, \varphi, Z) \left\{ \begin{array}{l} r(i \vee j) = \delta \times (i \vee j), \delta \in [0 \dots \infty] \\ \varphi(i \vee j) = \theta \times (i \vee j), \theta \in [0 \dots \infty] \\ Z(i \vee j) = \varepsilon \times (i \vee j), \varepsilon \in [0 \dots \infty] \end{array} \right\} \wedge (i \vee j) \in [1 \dots \infty] \quad (9)$$

де δ – коефіцієнт автоматизації РІ; θ – коефіцієнт схожості компонентів; ε – верхня межа граничних витрат.

За даними табл. 1 побудована гістограма порівняння прогнозованого за ІМР часу та фактичного часу виконання РІ (рис. 5).

Статистичні результати порівняння зведено до табл. 2.

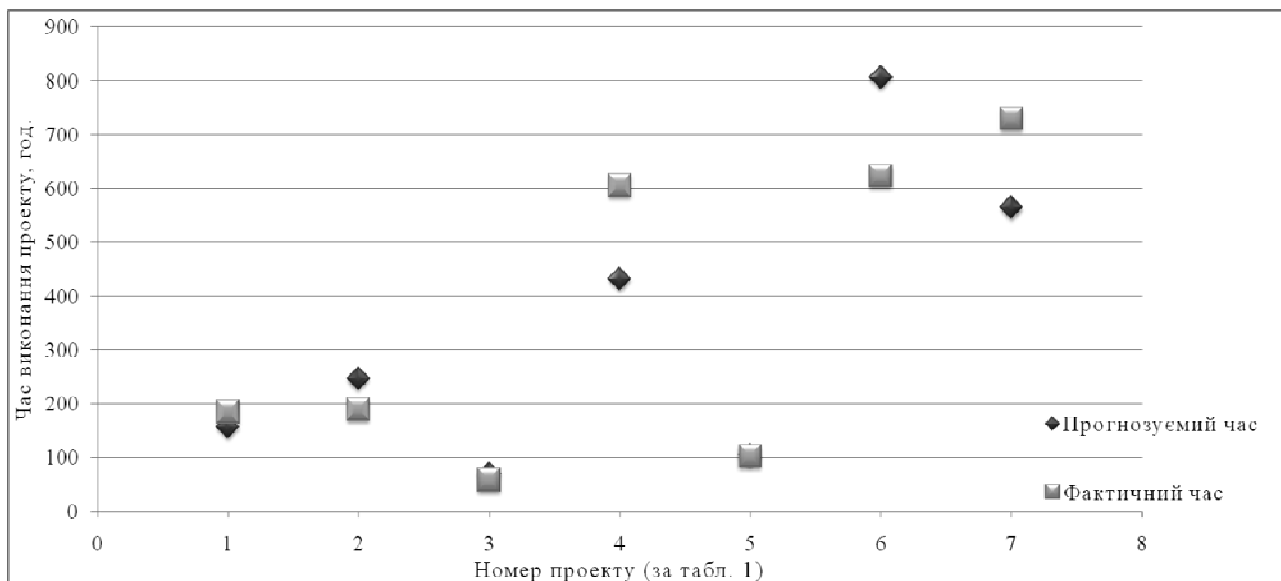


Рисунок 5 – Гістограма порівняння розрахованого за ІМР прогнозованого часу та часу фактичного виконання РІ проекту

Таблиця 2 – Розбіжність прогнозованого з фактичним часом виконання РІ проекту

Номер проекту	Абсолютна похибка, год.	Відносна похибка, %
1	28	15,1
2	57	30,0
3	11,5	19,7
4	173,5	28,6
5	2	2,0
6	183,5	29,4
7	165,5	22,7
Середнє значення відносної похибки, %:		21,1

6 ОБГОВОРЕННЯ

При оцінці полярної моделі РІ, експертами у галузі проектування та математичного моделювання було знайдено недоліки, що значною мірою перешкоджають її застосуванню системними аналітиками:

1) жодної з формул (1)–(3) недостатньо для розрахунку часу реінжинірингу φ , що потім було математично проаналізовано та уточнено [5], у зв'язку із чим час було фіксовано за фактом;

2) відсутня можливість обліку кількості ідентифікованих програмних компонентів (фізичних модулів коду), що необхідно наводити у лінійному масштабі, або кількості верифікованих рядків програмного коду – у логарифмічному масштабі;

3) подання ІМР тільки у проекції часу та витрат й неможливість реалізувати модель у декількох інших проекціях, що цікавлять системного архітектора, а саме:

– у ізометричній проекції програмних компонентів;

– у логарифмічній проекції рядків програмного коду.

З аналізу та обговорення цих недоліків було сформовано обмеження на застосування полярної моделі РІ: полярна система координат не відповідає висунутим вимогам подання проекцій, а значить необхідно шукати шляхи побудови ІМР у інших системах координат.

Подальші дослідження ІМР, що наведені у статті, узагальнили досвід роботи із системами координат у яких можливе застосування спіральної організації відліку.

З наведених графічних побудов функцій $\Phi(r, \varphi, Z)$ (рис. 3) – можна зробити висновки, що OM_i є утворюючою. Поверхня ІМР, для якої функція $\Phi(r, \varphi, Z)$ є направляючою, являє собою лінійчату конічну поверхню, що може бути зворотньовідновлена рухом утворюючої OM_i .

Експериментальні дослідження спіральної ІМР показали прогнозування витрат на РІ із максимальною похибкою 30% (табл. 2). Може здаватися, що діапазон похибки занадто великий, проте для первинної оцінки порядку витрат (при тому, що жодних інших даних на етапі прийняття рішення не існує), спіральна ІМР – цілком придатна.

ВИСНОВКИ

Таким чином, з наведених матеріалів, зрозуміло, що реінжиніринг дозволяє виконати еволюціонування ПС, шляхом внесення позитивних змін до її структури, з метою підвищення зручності її ж експлуатації та технічного супроводу.

До результатів наукової новизни слід віднести:

1) вперше сформовані ІМР, що дозволяють підвищити точність оцінки витрат з перепроєктування програмних систем;

2) удосконалено модель візуалізації витрат на перепроєктування ПС, яка дозволить підвищити ефективність прогнозування порядку витрат;

3) вперше отримано залежності побудови траєкторії графічної моделі витрат від уведених коефіцієнту автоматизації РІ та коефіцієнту схожості компонентів, які дозволяють вносити зміни до конфігурації моделі.

Запропоновані ІМР видів забезпечення ПС являють собою еволюційні спіралі, які побудовані у циліндричній системі координат. Операції з ІМР можуть відбуватися у наступних проекціях: у проекції часу та витрат; у ізометричній проекції програмних компонентів; у логарифмічній проекції рядків програмного коду.

Перспективи подальших досліджень полягають:

– у аналітичному визначенні коефіцієнту автоматизації РІ (δ), що базується на декількох складових процесу РІ (при відомому коефіцієнті автоматизації РІ – ІМР набуде валідної конфігурації та може бути застосована з високою точністю до реальних завдань);

– у визначенні та завданні коефіцієнту схожості компонентів (θ);

– у математичному встановленні у моделі верхньої межі граничних витрат (ϵ).

Реалізація та застосування перерахованих пунктів перспектив – дозволять зменшити середнє значення відносної похибки прогнозування витрат з теперішнього значення 21,1% до перспективного $7 \div 10\%$.

Прогнозується, що реінжиніринг, який буде виконано за допомогою розроблених ІМР дозволить не тільки скоротити витрати на перепроєктування ПС, але й підвищити ефективність технічного супроводу, збільшити життєвий цикл ПС, що вже знаходяться у експлуатації та подолати протиріччя між швидкими темпами розвитку науки, техніки і процесів проектування нових ПС.

ПОДЯКИ

Перш за все хочу висловити подяку доктору технічних наук зі спеціальності 05.12.13 – системи автоматизації проєктувальних робіт, професору, професору кафедри системотехніки Харківського національного університету радіоелектроніки – Безкорвайному 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 6-ти річної роботи, було отримано наукові результати, що наведено у поданій статті.

ЛІТЕРАТУРА / LITERATURE

1. Тимченко А. А. Основы системного проектирования та системного аналізу складних об'єктів. Кн. 1. Основы САПР та системного проектирования складных объектов / А. А. Тимченко. – К. : Либідь, 2003. – 272 с.
2. Blum B. Software engineering: a holistic view [Electronic resource]. – Access mode: <https://dl.acm.org/citation.cfm?id=SERIES9569.128915>.
3. Klein M. Reengineering methodologies and tools. A Prescription for Enhancing Success [Electronic resource]. – Access mode: <https://www.tandfonline.com/doi/abs/10.1080/10580539408964633>. DOI: 10.1080/10580539408964633
4. Grover V., Malhotra M. Business process reengineering: A tutorial on the concept, evolution, method, technology and application [Electronic resource]. – Access mode: <https://www.sciencedirect.com/science/article/abs/pii/S0272696396001040>. DOI: 10.1016/S0272-6963(96)00104-0
5. Manganelli R., Klein M. The Reengineering Handbook: A Step-by-Step Guide to Business Transformation [Electronic resource]. – Access mode: https://www.sciencedirect.com/science/article/pii/S00https://journals.lww.com/jhqonline/Citation/1995/03000/The_Reengineering_Handbook_A_Step_by_Step_Guide.11.aspx. DOI: 10.1097/01445442-199503000-00011
6. Jacobson I., Ericsson M., Jacobson A. The Object Advantage: Business Process Reengineering with Object Technology. ACM Press. [Electronic resource]. – Access mode: <http://eaststemcell.com/files/storage.cloud.php?id=MDIwMTQyMjg5MQ==>.
7. Boehm B. Software Risk Management [Electronic resource]. – Access mode: https://link.springer.com/chapter/10.1007%2F3-540-51635-2_29. DOI: 10.1007/3-540-51635-2_29
8. Hammer M., Champy J. Reengineering the corporation: A manifesto for business revolution [Electronic resource]. – Access mode: <https://www.sciencedirect.com/science/article/pii/S0007681305800643?via%3Dihub>. DOI: 10.1016/S0007-6813(05)80064-3
9. Великодний С. С. Методологические основы реинжиниринга систем автоматизированного проектирования [Текст] / С. С. Великодний // Управляющие системы и машины. – 2014. – № 2. – С. 39–43.
10. Boehm B. Spiral Development: Experience, Principles and Refinements : Special Report : CMU / SEI-2000-SR-008, 2000. – 37 p.
11. Boehm B. A Spiral Model of Software Development and Enhancement / B. Boehm // ACM SIGSOFT Software Engineering Notes. – 1986. – Vol. 11, Iss. 4. – P. 14–24. DOI: 10.1145/12944.12948
12. Selby R. W. Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management and Research / R. W. Selby. – New Jersey : John Wiley & Sons, 2007. – 818 p.
13. Выгодский М. Я. Справочник по высшей математике / М. Я. Выгодский. – М. : ГИТТЛ, 1957. – 784 с.
Стаття надійшла до редакції 06.09.2018.
Після доробки 22.10.2018.

УДК 004.942: 519.876.5

ИДЕАЛИЗИРОВАННЫЕ МОДЕЛИ РЕИНЖИНИРИНГА ПРОГРАММНЫХ СИСТЕМ

Великодний С. С. – канд. техн. наук, доцент, заведующий кафедры информационных технологий Одесской национальной академии связи им. А. С. Попова; доцент кафедры информационных технологий Одесского государственного экологического университета, Одесса, Украина.

АННОТАЦИЯ

Актуальность. Программные системы применяются в различных отраслях жизни и деятельности человека, но наибольшее распространение они получили в отраслях, где необходима работа со многими рутинными операциями – это касается промышленности, производства, транспорта, обучения и инфокоммуникаций. Конечно, в процессе эксплуатации этих систем происходит эволюционное старение видов обеспечения. Такая тенденция приводит к ухудшению скоростных, информационно-коммуникационных, графических, временных и других характеристик, вплоть до полного отказа программной системы. Тематика статьи направлена в области проектирования эргатических систем и управления проектами и программами по перепроектированию или усовершенствованию этих систем. Реинжиниринг позволяет выполнить эволюционирование системы, путем внесения позитивных изменений в ее структуру с целью повышения характеристик эксплуатации и технического сопровождения.

Цель. Сформировать представление оценки параметров расходов ресурсов на выполнение реинжиниринга программных систем с помощью математического инструментария описания моделей проектирования.

Метод. В идею статьи заложен метод Бозма (Boehm) и принципы построения спирали Архимеда. Авторские исследования привели к мысли объединить идеи построения структур Бозма и Архимеда, адаптировать их под методологию программной инженерии и перенести в цилиндрическую систему координат. В основу полученных моделей заложен спиральный принцип организации отчета.

Результаты. В статье рассмотрен процесс создания и визуализации математических моделей применительно к процессу реинжиниринга программных систем. Операции с представленными моделями могут происходить в проекциях времени и затрат, в изометрической проекции программных компонентов, в логарифмической проекции строк программного кода.

Выводы. Впервые сформированы идеализированные модели реинжиниринга, которые позволят повысить точность оценки затрат на перепроектирование программных систем. Усовершенствована модель визуализации затрат на перепроектирование программных систем, которая позволит повысить эффективность прогнозирования порядка затрат. Впервые получены зависимости построения траектории графической модели расходов от введенных коэффициента автоматизации реинжиниринга и коэффициента подобия компонентов, которые позволят вносить изменения в конфигурацию модели. Предложенные идеализированные модели реинжиниринга видов обеспечения программных систем представляют собой эволюционные спирали, которые построены в цилиндрической системе координат. Реинжиниринг, который будет выполнен с помощью разработанных идеализированных моделей реинжиниринга позволит не только сократить расходы на перепроектирование программных систем, но и повысить эффективность технического сопровождения, увеличить жизненный цикл программных систем, которые уже находятся в эксплуатации и преодолеть противоречия между быстрыми темпами развития науки, техники и процессов проектирования новых программных систем.

КЛЮЧЕВЫЕ СЛОВА: программная система, реинжиниринг, система автоматизированного проектирования, вид обеспечения, CASE-средство, компонент программного кода, идеализированная модель, перепроектирование, эволюционная спираль, радиус-вектор затрат.

UDC 004.942: 519.876.5

THE IDEALIZED MODELS OF SOFTWARE SYSTEMS REENGINEERING

Velykodniy Stanislav – PhD in Computer Science, head of the Information Technologies Department in O. S. Popov Odessa National Academy of Telecommunications; Associate Professor of the Information Technologies Department Odessa State Environmental University, Odessa, Ukraine.

ABSTRACT

Content. Software systems are used in a variety of branches of life and human activities, but they are most widely used in industries where it is necessary to work with many routine operations – this applies to industry, production, transportation, training and info communications. Of course, in the process of exploiting these systems there is an evolutionary aging of the different types of security. This trend leads to a deterioration of high-speed, information-communication, graphic, time and other characteristics, up to the complete failure of the software system. The subject of the article is aimed to the design of ergative systems, project management and programs for re-designing or improving these systems. Reengineering allows the system to evolve, by making positive changes to its structure in order to improve operational characteristics and technical support.

Objective. To form a representation of the estimation of resource cost parameters for the reengineering of software systems with the help of a mathematical tools for describing design models.

Method. The idea of the article is based on the Boehm method and the principles of building an Archimedes spiral. Author's research led to the idea of combining Boehm and Archimedes structures, adapting them to the software engineering methodology and transferring them to a cylindrical coordinate system. The basis of the obtained models is laid spiral principle of the organization of the reference.

Results. The article deals with the process of creating and visualizing mathematical models in relation to the process of reengineering software systems. Transactions with these models can occur in projections of time and cost, in the isometric projection of software components, and in the logarithmic projection of lines of software code.

Conclusions. For the first time, idealized models of reengineering have been formed, which will allow to increase the accuracy of cost estimation from the redesign of software systems. The model of cost visualization for reprogramming of software systems has been improved, which will increase the efficiency of forecasting the order of expenses. For the first time dependencies were obtained on the construction of the graphical cost model trajectory from the introduced reengineering automation coefficient and the similarity coefficient of the components that would make changes to the configuration of the model. The proposed idealized models for reengineering of software system security are evolutionary spirals, which are constructed in a cylindrical coordinate system. Reengineering, which will be accomplished with the help of developed idealized reengineering models, will not only reduce the cost of reprogramming software systems, but also increase the efficiency of technical support, increase the life cycle of software systems which are already in service and overcome the contradictions between the rapid pace of development of science, technology and processes of designing new software systems.

KEYWORDS: software system, reengineering, automated design, security, CASE-tool, software code component, idealized model, redesign, evolutionary spiral, radius-vector expenditure.

REFERENCES

1. Tymchenko A. A. Osnovy systemnogo proektuvannja ta systemnogo analizu skladnyh ob'ektiv. Book. 1. Osnovy SAPR ta systemnogo proektuvannja skladnyh ob'ektiv. Kiev, Lybid', 2003, 272 p.
2. Blum B. Software engineering: a holistic view [Electronic resource]. Access mode: <https://dl.acm.org/citation.cfm?id=SERIES9569.128915>.
3. Klein M. Reengineering methodologies and tools. A Prescription for Enhancing Success [Electronic resource]. Access mode: <https://www.tandfonline.com/doi/abs/10.1080/10580539408964633>. DOI: 10.1080/10580539408964633
4. Grover V., Malhotra M. Business process reengineering: A tutorial on the concept, evolution, method, technology and application [Electronic resource]. Access mode: <https://www.sciencedirect.com/science/article/abs/pii/S0272696396001040>. DOI: 10.1016/S0272-6963(96)00104-0
5. Manganelli R., Klein M. The Reengineering Handbook: A Step-by-Step Guide to Business Transformation [Electronic resource]. Access mode: https://www.sciencedirect.com/science/article/pii/S00https://journals.lww.com/jhqonline/Citation/1995/03000/The_Reengineering_Handbook_A_Step_by_Step_Guide.11.aspx. DOI: 10.1097/01445442-199503000-00011
6. Jacobson I., Ericsson M., Jacobson A. The Object Advantage, Business Process Reengineering with Object Technology. ACM Press. [Electronic resource]. Access mode: <http://eaststemcell.com/files/storage.cloud.php?id=MDIwMTQyMjg5MQ==>.
7. Boehm B. Software Risk Management [Electronic resource]. – Access mode: https://link.springer.com/chapter/10.1007%2F3-540-51635-2_29. DOI: 10.1007/3-540-51635-2_29
8. Hammer M., Champy J. Reengineering the corporation: A manifesto for business revolution [Electronic resource]. Access mode: <https://www.sciencedirect.com/science/article/pii/S0007681305800643?via%3Dihub>. DOI: 10.1016/S0007-6813(05)80064-3
9. Velykodniy S. S. Metodologicheskie osnovy reinzhiniringa sistem avtomatizirovannogo proektirovanija, *Upravljajushhie sistemy i mashiny*, 2014, No. 2, pp. 39–43.
10. Boehm B. Spiral Development: Experience, Principles and Refinements, Special Report, CMU / SEI-2000-SR-008, 2000, 37 p.
11. Boehm B. A Spiral Model of Software Development and Enhancement, *ACM SIGSOFT Software Engineering Notes*, 1986, Vol. 11, Iss. 4, pp. 14–24. DOI: 10.1145/12944.12948
12. Selby R. W. Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management and Research. New Jersey, John Wiley & Sons, 2007, 818 p.
13. Vygodskij M. Ja. Spravochnik po vysšej matematike. Moscow, GITTL, 1957, 784 p.