

## ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

## INFORMATION TECHNOLOGY

УДК 004.4:519.816

DOI: 10.20998/2079-0023.2022.02.16

*М. Д. ГОДЛЕВСЬКИЙ, Г. О. БУРЛАКОВ***ПЛАНУВАННЯ РОЗВИТКУ ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МЕТОДА ПОСЛІДОВНОГО АНАЛІЗУ ВАРІАНТІВ**

Зазначено, що процесний підхід є засобом наукового дослідження і практичної зміни навколишнього світу, який дозволяє представляти та реалізовувати виробничі системи у вигляді множини процесів. У роботі розглядаються особливості використання такого підходу до дослідження людської діяльності, пов'язаної з розробкою програмного забезпечення (ПЗ). Представлено перелік науковців, які звернули увагу на необхідність реалізації процесного підходу до розробки ПЗ і на цій основі було введено поняття процесу розробки ПЗ (ПР ПЗ). Визначено поняття моделі життєвого циклу (МЖЦ). Підкреслено, що найбільш відомими наборами МЖЦ є моделі зрілості СММІ та SPICE. Під покращенням ПР ПЗ у роботі розуміється множина дій, спрямованих на поліпшення його характеристик шляхом виконання деякого набору заходів, які базуються на інженерії якості, у тому числі на TQM. Підкреслено, що моделі зрілості відносяться до однієї з двох головних категорій: безперервні (continuous) і дискретні (staged). Модель СММІ реалізується на основі двох підходів, а модель SPICE тільки на основі безперервного підходу. На теперішній час модель СММІ формалізовано, що дозволило планувати розвиток ПР ПЗ або його окремих процесів в умовах обмежених ресурсів. З погляду еталонної моделі SPICE представлена формалізація її структури у вигляді наступної ієрархії: множина груп процесів; множина категорій; множина процесів; множина практик. На основі наведеної структури проведена формалізація оцінки рівня можливості окремого процесу моделі SPICE. Розроблена модель стала основою для розробки оптимізаційної задачі планування розвитку підмножини процесів моделі зрілості SPICE. З метою аналізу цієї задачі і розробки алгоритму її реалізації в роботі вона представлена у скороченому вигляді. Її цільова функція є адитивною і визначає інтегральний показник збільшення рівня можливості підмножини процесів на всьому плановому періоді. Модель має ресурсне обмеження, яке характеризується тим, що ресурси, які не використано на заданому підперіоді, можуть бути реалізовані на наступних підперіодах планування. Показано, що для оптимізації такої моделі найбільш придатним є метод послідовного аналізу варіантів, який може бути реалізованим на основі різних алгоритмів. У роботі використано алгоритм «Київський віник», який є багатокроковим процесом, на кожному підперіоді планування якого реалізується «відмітання» деякої множини неконкурентоспроможних варіантів.

**Ключові слова:** процесний підхід, програмне забезпечення, процес розробки програмного забезпечення, модель життєвого циклу, інженерія якості, модель зрілості, метод послідовного аналізу варіантів.

*M. D. GODLEVSKYI, G. O. BURLAKOV***PLANNING THE EVOLUTION OF THE SOFTWARE DEVELOPMENT PROCESS BASED ON THE METHOD OF SEQUENTIAL OPTION ANALYSIS**

It is noted that the process approach is a means of scientific research and practical change of the surrounding world, which allows representing and implementing production systems in the form of a set of processes. The paper considers the peculiarities of using this approach to the study of human activity related to software development. A list of scientists, who drew attention to the necessity of implementing a process approach to software development, is presented, and on this basis, the concept of the software development process (SDP) was introduced. The concept of a model life cycle (MLC) is defined. It is emphasized that the most well-known sets of MLC are SMMI and SPICE maturity models. The improvement of SDP in the work is understood as a set of actions aimed at improving its characteristics by implementing a certain set of measures that are based on quality engineering, including TQM. It is emphasized that maturity models belong to one of two main categories: continuous and discrete (staged). The SMMI model is implemented on the basis of two approaches, and the SPICE model is implemented only on the basis of a continuous approach. Currently, the SMMI model has been formalized, which made it possible to plan the development of the SDP or its separate processes under conditions of limited resources. From the point of view of the SPICE reference model, the formalization of its structure is presented in the form of the following hierarchy: a set of process groups; multiple categories; set of processes; set of practices. Based on the given structure, the assessment of the possibility level of a separate process for SPICE model was formalized. The developed model became a basis for optimization problem development of planning evolution of a processes subset for the SPICE maturity model. In order to analyze this problem and develop an algorithm for its implementing, it is presented in a shortened form. Its objective function is additive and determines the integral indicator of the increase in the level of possibility of a processes subset over the entire planning period. The model has a resource limitation, which is characterized by the fact that resources that are not used in a given subperiod can be implemented in subsequent planning subperiods. It is shown that the method of sequential analysis of options, which can be implemented based on various algorithms, is the most suitable for the optimization of such a model. The work uses the "Kyiv Broom" algorithm, which is a multi-step process, at each planning sub-period of which "sweeping out" a certain set of non-competitive options is implemented.

**Keywords:** process approach, software, software development process, model life cycle, quality engineering, maturity model, method of

© М. Д. Годлевський, Г. О. Бурлаков, 2022



**Дослідницька стаття:** Цю статтю опубліковано видавництвом *НТУ «ХПІ»* у збірнику «Вісник Національного технічного університету «ХПІ» Серія: Системний аналіз, управління та інформаційні технології». Ця стаття поширюється за міжнародною ліцензією [Creative Commons Attribution \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/). **Конфлікт інтересів:** Автор/и заявив/или про відсутність конфлікту.



sequential analysis of options.

**Вступ.** Процесний підхід є засобом наукового дослідження і практичної зміни навколишнього світу, який дозволяє представляти та реалізовувати виробничі системи множиною процесів, де під процесом будемо розуміти пов'язаний набір функцій, який перетворює вихідну інформацію в кінцеву інформацію (продукт, послугу) відповідно до попередньо визначених правил. У роботі розглядаються особливості використання такого підходу до дослідження людської діяльності, пов'язаної з розробкою програмного забезпечення. Вихідним матеріалом для такого роду діяльності є вимоги до програмного продукту або сервісу і необхідні ресурси, а її результат – кінцевий програмний продукт або сервіс. Очевидно, що така діяльність відповідає раніше наведеному поняттю процесу і, як слідство, можна ввести поняття процесу розробки програмного забезпечення (ПР ПЗ) [1–5].

Першим дослідником, який звернув увагу на необхідність реалізації процесного підходу до розробки ПЗ був Хамфрі, основні результати досліджень якого відображені у монографії [6]. Документ SWEBOOK [7] (який описує базовий набір знань програмної інженерії та відомий як стандарт ISO/IEC 19759) наводить наступне визначення: під ПР ПЗ розуміється набір діяльностей, методів, практичних процедур та перетворень, які використовуються людьми для розробки і підтримки ПЗ і пов'язаних з ним продуктів. Лавріщева [8] визначає ПР ПЗ (під назвою «базовий процес програмної інженерії») як множину логічно пов'язаних видів організаційної діяльності організації-розробника і набору засобів та інструментів, які відносяться до виготовлення програмного продукту. Як бачимо, ці визначення відрізняються тільки частковостями і є узгодженими в основному: у всіх визначеннях під ПР ПЗ розуміється упорядкована множина дій, які ставлять своєю метою створення програмного продукту. Відмінності фактично пов'язані з інтерпретацією конкретного набору дій. Опис структури конкретного екземпляра ПР ПЗ визначає набір етапів і окремих його елементів. При реалізації конкретних проектів визначається порядок слідування цих етапів і критерії переходу від етапу до етапу. Фактично, необхідно отримати відповідь на два основних питання [2]. Який набір етапів ПР ПЗ використовується в конкретній реалізації? Як задається послідовність цих етапів і умов переходу між ними? Відповідь на перше питання дає визначення процесів життєвого циклу (ЖЦ) ПР ПЗ. Відповідь на друге питання дає визначення моделі ЖЦ ПР ПЗ. Така модель дає послідовність виконання процесів ЖЦ (ПЖЦ) і умов переходу між ними. Існує безліч варіантів ПЖЦ, які використовуються в конкретних реалізаціях процесу розробки (моделі життєвого циклу). Зазвичай такі варіанти мають множину загальних елементів. У роботі [9] наведено загальний набір складових ПЖЦ, який є покриттям більшості існуючих варіантів. Найбільш відомими наборами ПЖЦ, які використовуються на практиці, є: набір ПЖЦ, визначений методом покращення процесу розробки, який базується на моделі зрілості СММІ (Capability Maturity Model Integration); набір ПЖЦ,

який визначається методом покращення процесу розробки, який базується на моделі зрілості SPICE (Software Process Improvement and Capability dEtermination) [10].

**Постановка та мета задачі дослідження.** Під покращенням ПР ПЗ (Software Process Improvement, SPI) [11, 12] розуміється множина дій, спрямованих на поліпшення характеристик процесу розробки в результаті виконання деякого набору заходів. Важливим моментом є той факт, що критерієм успіху цих дій є характеристики процесу розробки (його документування, керованість і т. ін.), а не характеристики програмного забезпечення, яке розробляється. У цьому ця технологія спирається на підходи інженерії якості, у тому числі на TQM (Total Quality Management). Вважається, що покращення характеристик процесу розробки напряму пов'язане з якістю його результатів. Існують різні підходи до покращення процесу розробки. Далі ми зосередимося на підходах, які базуються на моделях зрілості та представляють опис ідеального просування організації (або окремого підрозділу) до бажаного рівня реалізації процесу розробки, який задано декількома послідовними етапами або рівнями. Такий процес доповнюється засобами оцінки повноти виконання етапів або відповідності процесу організації рівням, які описано. Така модель дозволяє організації оцінити свою реалізацію процесу розробки шляхом її порівняння з кращими варіантами такої реалізації (або з варіантами конкурентів). Мета такого порівняння – сформулювати структурований план покращення цієї реалізації процесу.

На цей час прийнято відносити моделі зрілості до однієї з двох головних категорій: безперервні (continuous) і дискретні (поетапні, staged) підходи. Вище наведена модель СММІ реалізується на основі двох підходів, а модель SPICE тільки на основі безперервного підходу. При цьому модель СММІ дозволяє проводити оцінку ПР ПЗ всієї організації за допомогою дискретного підходу і окремих процесів (фокусних областей) при безперервному підході. Модель SPICE дозволяє оцінювати якість тільки окремих процесів або множини таких процесів. Основним недоліком цих моделей була відсутність їх формального опису, який дозволяє планувати розвиток ПР ПЗ або його окремих процесів в умовах обмежених ресурсів на деякому плановому періоді. Цей недолік було скасовано щодо моделі СММІ у роботах [13, 14]. Проведено перші дослідження про формалізацію процесів еталонної моделі зрілості SPICE [15]. Формалізовано оцінку рівня можливості окремого процесу моделі SPICE і розроблено динамічну модель планування розвитку підмножини процесів моделі зрілості SPICE. Тому метою роботи є розробка алгоритму реалізації цієї моделі на деякому плановому періоді в умовах обмежених ресурсів. Для формування алгоритму попередньо коротко розглянемо модель задачі, яку було розроблено в [15].

**Динамічна модель планування розвитку підмножини процесів еталонної моделі зрілості SPICE.** Введемо наступні позначення, які необхідні

для розробки динамічної моделі планування розвитку підмножини процесів еталонної моделі SPICE:  $G$  – множина груп процесів, де  $g \in G$ ;  $K_g$  – множина категорій процесів  $g$ -ї групи, де  $k \in K_g$ ;  $P_{gk}$  – множина окремих процесів (підпроцесів), де  $p \in P_{gk}$ ;  $I_{gk}^p$  – множина практик, де  $i \in I_{gk}^p$ . Кожна практика характеризується множиною  $A$  атрибутів, де  $\alpha \in A$ , які наведено у [15, табл. 1]. З метою формалізації моделі SPICE введено дискретні змінні  $X_{gk}^{pt}(i, \alpha) \in \{1, 2, 3, 4\}$ , які визначають для  $i$ -ї практики ступінь володіння рисою атрибута  $\alpha$  у  $t$ -му підперіоді планування, де  $t \in [1, T]$ , а  $T$  – кількість підперіодів. Одиниця визначає мінімальне володіння рисою в межах  $[0\% - 15\%]$ , а чотири – максимальне в межах  $[86\% - 100\%]$ . Отже, кожний процес моделі зрілості SPICE характеризується наступною матрицею оцінок

$$X_{gk}^{pt} = \{X_{gk}^{pt}(i, \alpha), i \in I_{gk}^p, \alpha \in A\}, p \in P_{gk}, k \in K_g, g \in G, t \in [1, T].$$

Далі введено поняття граничної матриці оцінок рівня можливості окремої практики

$$\Lambda = \{\Lambda(\gamma, \alpha), \gamma = \overline{0,5}, \alpha \in A\},$$

де  $\gamma$  – рівень можливості,  $\alpha$  – індекс атрибуту. На основі матриці оцінок і граничної матриці шляхом згортки окремих практик в межах відповідного процесу [15] сформовано вектор відповідності рівням можливості для  $p$ -го процесу еталонної моделі SPICE

$$\Theta_{gk}^{pt} = \{\Theta_{gk}^{pt}(\gamma), \gamma = \overline{0,5}\}.$$

Кожний  $p$ -й процес досягає  $\gamma$ -рівня можливості при умові, що  $\Theta_{gk}^{pt}(\gamma) = 1$ . Якщо  $\Theta_{gk}^{pt}(\gamma) < 1$ , то будемо вважати, що ступінь досягнення  $\gamma$ -рівня можливості  $p$ -м процесом дорівнює  $\Theta_{gk}^{pt}(\gamma)$ . Вважається, що інтегральний рівень можливості  $p$ -го процесу у  $t$ -му підперіоді планування визначається наступним чином:

$$\bar{\Theta}_{gk}^{pt} = \prod_{\gamma=1}^5 \Theta_{gk}^{pt}(\gamma), p \in \bar{P}_{gk}, k \in \bar{K}_g, g \in \bar{G}, t \in [1, T],$$

де  $\bar{G}$  – підмножина груп процесів,  $\bar{G} \subseteq G$ ;  $\bar{K}_g$  – підмножина категорій процесів,  $\bar{K}_g \subseteq K_g$ ;  $\bar{P}_{gk}$  – підмножина процесів (підпроцесів),  $\bar{P}_{gk} \subseteq P_{gk}$ . Рівень можливості  $\gamma = 0$  вважається досягнутим по визначенню. Вектор відповідності  $\Theta_{gk}^{pt}$  залежить від матриці оцінок  $X_{gk}^{pt}$ . Тому інтегральний рівень можливості підмножини процесів у  $t$ -му підперіоді може бути визначеном у такий спосіб

$$\bar{F}^t(\bar{X}^t) = \sum_{g \in \bar{G}} \bar{\rho}_g \sum_{k \in \bar{K}_g} \bar{\rho}_{gk} \sum_{p \in \bar{P}_{gk}} \bar{\rho}_{gk}^p \cdot \bar{\Theta}_{gk}^{pt}(X_{gk}^{pt}),$$

де  $\{\bar{\rho}_g\}, \{\bar{\rho}_{gk}\}, \{\bar{\rho}_{gk}^p\}$  – відповідні вагові коефіцієнти, а

$$\bar{X}^t = \{X_{gk}^{pt}(i, \alpha), i \in I_{gk}^p, \alpha \in A, p \in \bar{P}_{gk}, k \in \bar{K}_g, g \in \bar{G}\}, t \in [1, T].$$

В результаті цільова функція динамічної моделі планування розвитку підмножини процесів моделі зрілості SPICE записується у такий спосіб

$$F(\bar{X}) = \sum_{t \in [1, T]} \xi_t \cdot \Phi_t(\bar{X}^{t-1}, \bar{X}^t), \quad (1)$$

де

$$\Phi_t(\bar{X}^{t-1}, \bar{X}^t) = \bar{F}^t(\bar{X}^t) - \bar{F}^{t-1}(\bar{X}^{t-1}),$$

а  $\{\xi_t\}$  – вектор вагових коефіцієнтів важливості окремих підперіодів планування.

Перейдемо до розгляду питання формування функції фінансових витрат на  $(t-1)$ -му підперіоді планування, які забезпечують приріст рівня зрілості підмножини процесів у  $t$ -му підперіоді на величину  $\Phi_t(\bar{X}^{t-1}, \bar{X}^t)$ . З цією метою введені трикутні матриці варіантів розвитку окремих атрибутів практик (табл. 1), елементи яких

$${}^{\alpha}r_{gk}^{pi}(X_{gk}^{p(t-1)}(i, \alpha), X_{gk}^{pt}(i, \alpha)), \alpha \in A, i \in I_{gk}^p, p \in \bar{P}_{gk}, k \in \bar{K}_g, g \in \bar{G}$$

визначають необхідні фінансові ресурси при переході від ступеня володіння рисою  $X_{gk}^{p(t-1)}(i, \alpha)$   $\alpha$ -атрибуту до ступеня  $X_{gk}^{pt}(i, \alpha)$ . Отже, сумарні фінансові витрати у  $(t-1)$ -му підперіоді планування мають наступний вигляд:

$$\bar{R}_t(\bar{X}^{t-1}, \bar{X}^t) = \sum_{g \in \bar{G}} \sum_{k \in \bar{K}_g} \sum_{p \in \bar{P}_{gk}} \sum_{i \in I_{gk}^p} \sum_{\alpha \in A} {}^{\alpha}r_{gk}^{pi}(X_{gk}^{p(t-1)}(i, \alpha), X_{gk}^{pt}(i, \alpha)).$$

Таблиця 1 – Трикутна матриця для  $\alpha$ -атрибуту

$X_{gk}^{pt}(i, \alpha)$				
$X_{gk}^{p(t-1)}(i, \alpha)$				
1	0	${}^{\alpha}r_{gk}^{pi}(1,2)$	${}^{\alpha}r_{gk}^{pi}(1,3)$	${}^{\alpha}r_{gk}^{pi}(1,4)$
2	0	0	${}^{\alpha}r_{gk}^{pi}(2,3)$	${}^{\alpha}r_{gk}^{pi}(2,4)$
3	0	0	0	${}^{\alpha}r_{gk}^{pi}(3,4)$
4	0	0	0	0

Тоді формується множина  $[1, T]$  фінансових обмежень

$$\sum_{\tau=1}^t \bar{R}_{\tau}(\bar{X}^{\tau-1}, \bar{X}^{\tau}) \leq \sum_{\tau=0}^{t-1} R_{\tau} = \hat{R}^{t-1}, t \in [1, T], \quad (2)$$

де  $R_{\tau}$  – фінансові ресурси, які виділяються ІТ-компанією на розвиток підмножини процесів у  $\tau$ -підперіоді. Якщо вони не використані у  $\tau$ -підперіоді, вони переносяться на наступні підперіоди. Отже, на

основі сформованої моделі планування розвитку підмножини процесів моделі зрілості SPICE ставиться наступна задача. Знайти оптимальне значення  $\bar{X} = \{^* \bar{X}^t\}$ , яке забезпечує максимальне значення критерія (1) при обмеженнях (2).

**Алгоритм методу послідовного аналізу варіантів.** Оптимізаційні задачі характеризуються тим, що чим більше обмежень накладається на змінні, тим складніше задача оптимізації. З іншої сторони, чим менша область допустимих варіантів функціонування, тим більш оптимістичною є задача, яка зводиться до спрямованого або повного перебору варіантів. Фактично, метод послідовного аналізу варіантів побудовано на цій основі шляхом використання процедур непрямих оцінок, які дозволяють відкинути не оптимальні рішення. Отже, в ході реалізації метода відкидаються неконкурентні варіанти поки не залишається декілька або один варіант. При наявності декількох варіантів вони порівнюються між собою і вибирається кращий. Ця ідея метода послідовного аналізу варіантів була реалізована академіками Н. З. Шором і В. С. Міхалевичем як алгоритм «Київський віник», який є багатокроковим процесом. На кожному підперіоді планування вирішується задача «відмітання» множини неконкурентоспроможних варіантів. Використання алгоритму дозволяє знайти оптимальний варіант розв'язання задачі.

Перейдемо до його адаптації по відношенню до моделі планування розвитку підмножини процесів еталонної моделі зрілості SPICE. Зробимо наочну інтерпретацію алгоритму з метою більш детального його розгляду. Введемо простори  $\Pi^0, \Pi^1, \dots, \Pi^T$  змінних, відповідно,  $\bar{X}^0, \bar{X}^1, \dots, \bar{X}^T$  моделі (1), (2). З погляду на ресурсні обмеження (2) будемо вважати, що на кожному підперіоді планування існують дискретні множини  $\Omega^0, \Omega^1, \dots, \Omega^T$ , які визначають області змінних, відповідно,  $\bar{X}^0, \bar{X}^1, \dots, \bar{X}^T$ . У результаті маємо:

$$\bar{X}^0 \in \Omega^0 \subset \Pi^0, \bar{X}^1 \in \Omega^1 \subset \Pi^1, \dots, \bar{X}^T \in \Omega^T \subset \Pi^T.$$

Змінна  $\bar{X}^0 = ^* \bar{X}^0$  визначає початковий стан підмножини процесів моделі зрілості SPICE і тому множина  $\Omega^0$  складається з одного елемента. результатом виникає задача визначення ламаної  $^* \bar{X}^0, ^* \bar{X}^1, \dots, ^* \bar{X}^T$ , «довжина» якої максимальна і яка задовольняє умовам

$$^* X_{gk}^{pt}(i, \alpha) \in \{1, 2, 3, 4\}, ^* X_{gk}^{p0}(i, \alpha) \in \{1, 2, 3, 4\},$$

$$^* X_{gk}^{pt-1}(i, \alpha) \leq ^* X_{gk}^{pt}(i, \alpha), i \in I_{gk}^p,$$

$$\alpha \in A, p \in \bar{P}_{gk}, k \in \bar{K}_g, g \in \bar{G}, t \in [1, T].$$

Розглянемо безпосередньо реалізацію алгоритму «Київський віник», спираючись на наочну інтерпретацію алгоритму, яку наведено на рисунку 1. Попередньо формується множина

$$\Omega^1 = \{ \bar{X}^1 : ^* \bar{X}^0 \leq \bar{X}^1 \leq \bar{X}^T, \bar{R}_1(^* \bar{X}^0, \bar{X}^1) \leq R_0 \}.$$

Для кожної змінної  $\bar{X}^1 \in \Omega^1$  визначається «довжина відрізка»  $\Psi_1(\bar{X}^1) = \xi_1 \Phi_1(^* \bar{X}^0, \bar{X}^1)$ .

Величина  $\Psi_1(\bar{X}^1)$  визначає приріст цільової функції на першому підперіоді планування при різних значеннях  $\bar{X}^1$ . Далі розглянемо цільову функцію

$$F(\bar{X}), \text{ де } \bar{X} = \{ \bar{X}^t, t \in [0, T] \}.$$

Представимо її у наступному вигляді

$$\max_{\bar{X}^0 \in \Omega^0} F(\{ \bar{X}^t, t \in [0, T] \}) = \Psi_1(\bar{X}^1) + \sum_{t=2}^T \xi_t \Phi_t(\bar{X}^{t-1}, \bar{X}^t).$$

Можна сказати, що будь-яка ламана, в якій відсутній відрізок  $\Psi_1(\bar{X}^1)$ , не є претендентом для пошуку оптимального рішення. Якщо вважати, відповідно до концепції алгоритму «Київський віник», що на кожному  $t$ -му кроці алгоритму проводиться «відмітання» множини  $\Theta^t$  неконкурентоспроможних варіантів, про яку стає відомо, що вона не містить оптимального варіанта, то  $\Theta^0 = \emptyset$  оскільки ця множина складається з одного елемента. Далі розглянемо наступний крок алгоритму і сформуємо множину

$$\Omega^2 = \left\{ \bar{X}^2 : ^* \bar{X}^0 \leq \bar{X}^2 \leq \bar{X}^T, \sum_{t=1}^2 \bar{R}_t(\bar{X}^{t-1}, \bar{X}^t) \leq \sum_{t=0}^1 R_t \right\}.$$

Ресурси, які не використано на попередньому підперіоді, можуть бути використані на наступних підперіодах планування. Тому  $\Omega^1 \subseteq \Omega^2$ . Далі будемо вважати, що ламана «максимальної довжини», яка з'єднує  $\bar{X}^2$  з  $^* \bar{X}^0$ , визначається наступним чином

$$\Psi_2(\bar{X}^2) = \max_{\bar{X}^1 \in \Omega^1} \{ \Psi_1(\bar{X}^1) + \xi_2 \Phi_2(\bar{X}^1, \bar{X}^2) \}, \bar{X}^2 \in \Omega^2.$$

В результаті максимізації відкидається деяка множина  $\Theta^1$  неконкурентоспроможних варіантів. Максимізація реалізується шляхом повного перебору елементів  $\bar{X}^1 \in \Omega^1$  для кожного елемента  $\bar{X}^2 \in \Omega^2$ .

При цьому вважається, що  $\bar{X}^1 \leq \bar{X}^2$ . Далі по аналогії з попереднім кроком алгоритму

$$\Omega^3 = \left\{ \bar{X}^3 : ^* \bar{X}^0 \leq \bar{X}^3 \leq \bar{X}^T, \sum_{t=1}^3 \bar{R}_t(\bar{X}^{t-1}, \bar{X}^t) \leq \sum_{t=0}^2 R_t \right\}.$$

Ламана «максимальної довжини», яка з'єднує  $\bar{X}^3$  з  $^* \bar{X}^0$ , визначається наступним чином (рис. 1):

$$\Psi_3(\bar{X}^3) = \max_{\bar{X}^2 \in \Omega^2} \{ \Psi_2(\bar{X}^2) + \xi_3 \Phi_3(\bar{X}^2, \bar{X}^3) \}, \bar{X}^3 \in \Omega^3.$$

У результаті максимізації відкидається деяка множина  $\Theta^2$  неконкурентоспроможних варіантів. На основі отриманих результатів на перших кроках алгоритму проведемо їх узагальнення на прикладі  $t$ -го кроку, для якого формується множина

$$\Omega^t = \left\{ \bar{X}^t : * \bar{X}^0 \leq \bar{X}^t \leq \bar{X}^T, \sum_{\tau=1}^T \bar{R}_\tau(\bar{X}^{\tau-1}, \bar{X}^\tau) \leq \sum_{\tau=0}^{t-1} R_\tau \right\}, \\ t \in [1, T].$$

Як і було представлено вище, ламана «максимальної довжини», яка з'єднує  $\bar{X}^t$  з  $* \bar{X}^0$  визначається таким чином:

$$\Psi_t(\bar{X}^t) = \max_{\bar{X}^{t-1} \in \Omega^{t-1}} \left\{ \Psi_{t-1}(\bar{X}^{t-1}) + \xi_t \Phi_t(\bar{X}^{t-1}, \bar{X}^t) \right\}, \bar{X}^t \in \Omega^t.$$

На кожному кроці алгоритму «відмітається» деяка множина  $\Theta^{t-1}$  неконкурентоспроможних варіантів. В результаті на останньому  $T$ -му кроці ми отримуємо множину  $\{\Psi_T(\bar{X}^T), \bar{X}^T \in \Omega^T\}$ . Крайній варіант визначається наступним чином:

$$\Psi^* = \max_{\bar{X}^T \in \Omega^T} \left\{ \Psi_T(\bar{X}^T) \right\}.$$

Необхідно підкреслити, що алгоритм «Київський віник» дозволяє знайти глобальний екстремум цільової функції. При цьому на окремі функції  $\Phi_t(\bar{X}^{t-1}, \bar{X}^t)$  ніяких умов (опуклість, увігнутість) не накладається.

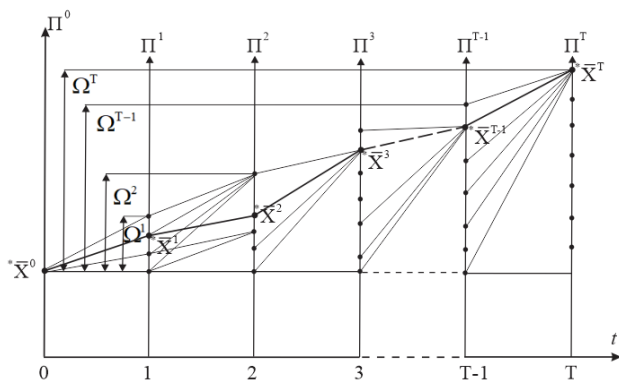


Рис. 1. Наочна інтерпретація алгоритму

Але одним з недоліків алгоритму є його велика трудомісткість. Тому виникає задача розробки деяких алгоритмів, які, з одного боку, не дозволяють знайти глобальний екстремум, але дають можливість значно зменшити його трудомісткість. Одним з підходів до вирішення цієї проблеми є метод «Блукаюча трубка». Він має характер метода послідовних наближень і не забезпечує знаходження глобального екстремуму, але має значно меншу трудомісткість по відношенню до алгоритму «Київський віник».

**Висновки, шляхи подальших досліджень.** У роботі підкреслена важливість процесного підходу у дослідженнях, пов'язаних з розробкою програмного забезпечення. Представлені найбільш відомі дослідники, які першими визначили необхідність реалізації процесного підходу до розробки ПЗ. Наведено найбільш відомі набори процесів життєвого циклу, до яких відносяться моделі зрілості CMMI та SPICE. На основі попередньо розробленої динамічної моделі планування розвитку підмножини процесів еталонної моделі зрілості SPICE поставлена задача розробки

алгоритму реалізації цієї моделі на деякому плановому періоді в умовах обмежених ресурсів. В роботі стисло представлена динамічна модель в більш зрозумілому ракурсі з метою її використання при розробці алгоритму. Аналіз моделі показав, що найбільш дієвим методом для її реалізації є метод послідовного аналізу варіантів. В результаті подальшого аналізу моделі стало зрозумілим, що для її реалізації найбільше підходить алгоритм «Київський віник». В роботі проведено адаптацію цього алгоритму відносно до конкретної моделі та його представлення як придатним для практичної реалізації.

Подальші дослідження будуть проводитись у наступних напрямках:

- зменшення трудомісткості алгоритму за рахунок використання метода «Блукаюча трубка»;
- реалізація інформаційної технології підтримки прийняття рішень при плануванні розвитком підмножини процесів еталонної моделі зрілості SPICE;
- дослідження розробленої інформаційної технології на тестових прикладах реальної розмірності.

#### Список літератури

1. Goodman F. A. *Defining and deploying software processes*. Auerbach Publications, 2006. 221 p.
2. Madachy R. J. *Software process dynamics*. Hoboken, NJ: IEEE Press, Wiley Interscience, 2008. 601 p.
3. Li T. *An Approach to Modelling Software Evolution Processes*. Berlin-Heidelberg: Springer, 2008. 213 p.
4. *Agile Processes in Software Engineering and Extreme Programming* / P. Abrahamsson, R. Baskerville, K. Conboy, et al. (eds.). Berlin-Heidelberg: Springer, 2008. 258 p.
5. Андон Ф. И., Коваль Г. И., Коротун Т. М. *Основы инженерии качества программных систем*. Киев: Академперіодика, 2007. 672 с.
6. Humphrey W. S. *Managing the software process*. Boston, MA: Addison-Wesley, 1989. 512 p.
7. *Software Engineering — Guide to the Software Engineering Body of Knowledge (SWEBOK)*, ISO/IEC TR 19759:2005. ISO, 2005. 173 p.
8. Лаврішчева К. М. *Програмна інженерія*. Київ: Академперіодика, 2008. 319 с.
9. Scacchi W., Marciniak J. *Process models in software engineering: Encyclopedia of Software Engineering*. New York: John Wiley & Sons, 2002. P. 993–1005.
10. *Оценка и аттестация зрелости процессов создания и сопровождения программных средств и информационных систем (ISO/IEC TR 15504-CMM)*. Москва: Книга и бизнес, 2001. 348 с.
11. Persse J. R. *Process Improvement Essentials*. O'Reilly, 2006. 352 p.
12. Poulin L. A. *Reducing risk with software process improvement*. Auerbach Pubs, 2005. 288 p.
13. Godlevskiy M. D., Goloskokova A. A., Chipizhenko A. A. Medium-term planning information technology for quality improvement of the software development process based on the CMMI model. *Вісник Національного технічного університету «ХПІ»*. Серія: Системний аналіз, управління та інформаційні технології. Харків: НТУ «ХПІ», 2017. № 51 (1272). С. 32–37.
14. Godlevskiy M. D., Goloskokova A. A., Bielous O. S. Information technology of a static model solving for quality improvement of the software development process based on the CMMI model. *Вісник Національного технічного університету «ХПІ»*. Серія: Системний аналіз, управління та інформаційні технології. Харків: НТУ «ХПІ», 2017. № 55 (1276). С. 26–30.
15. Godlevskiy M. D., Goloskokova A. A., Burлаков Г. О. Динамічна модель планування розвитку підмножини процесів еталонної моделі зрілості SPICE. *Вісник Національного технічного університету «ХПІ»*. Серія: Системний аналіз, управління та інформаційні технології. Харків: НТУ «ХПІ», 2020. № 2 (4). С. 10–16.

## References (transliterated)

1. Goodman F. A. *Defining and deploying software processes*. Auerbach Publ., 2006. 221 p.
2. Madachy R. J. *Software process dynamics*. Hoboken, NJ: IEEE Press, Wiley Interscience, 2008. 601 p.
3. Li T. *An Approach to Modelling Software Evolution Processes*. Berlin-Heidelberg: Springer, 2008. 213 p.
4. *Agile Processes in Software Engineering and Extreme Programming* / P. Abrahamsson, R. Baskerville, K. Conboy, et al. (eds.). Berlin-Heidelberg: Springer, 2008. 258 p.
5. Andon F. I., Koval G. I., Korotun T. M. *Osnovy inzhenerii kachestva programnykh sistem* [Fundamentals of Software Systems Quality Engineering]. Kyiv, Akademperiodika, 2007. 672 p.
6. Humphrey W. S. *Managing the software process*. Boston, MA, Addison-Wesley, 1989. 512 p.
7. *Software Engineering — Guide to the Software Engineering Body of Knowledge (SWEBOOK), ISO/IEC TR 19759:2005*. ISO, 2005. 173 p.
8. Lavrisheva K. M. *Programna Inzheneriya* [Software Engineering]. Kyiv, Akademperiodika, 2008. 319 p.
9. Scacchi W., Marciniak J. *Process models in software engineering*. Encyclopedia of Software Engineering. New York: John Wiley & Sons, 2002, pp. 993–1005.
10. *Otsenka i attestatsiya zrelosti protsessov stvoreniya i sprovodzhenniya programnykh sredstv i informatsionnykh sistem (ISO/IEC TR 15504-CMM)* [Assessment and certification of the maturity of the processes of creating and maintaining software tools and information systems (ISO/IEC TR 15504-CMM)]. Moscow, Book and business, 2001. 348 p.
11. Perse J. R. *Process Improvement Essentials*. O'Reilly, 2006. 352 p.
12. Poulin L. A. *Reducing risk with software process improvement*. Auerbach Pubs, 2005. 288 p.
13. Godlevskiy M. D., Goloskokova A. A., Chipizhenko A. A. Medium-term planning information technology for quality improvement of the software development process based on the CMMI model. *Visnyk NTU "KhPI": zb. nauk. pr. Seriya: Sy'stemny'j analiz, upravlinnya ta informacijni tehnologiyi*. [Bulletin of NTU "KhPI". Series: System analysis, control and information technology]. Kharkiv, NTU "KhPI" Publ., 2017, no. 51 (1272), pp. 32–37.
14. Godlevskiy M. D., Goloskokova A. A., Bielous O. S. Information technology of a static model solving for quality improvement of the software development process based on the CMMI model. *Visnyk NTU "KhPI": zb. nauk. pr. Seriya: Sy'stemny'j analiz, upravlinnya ta informacijni tehnologiyi*. [Bulletin of NTU "KhPI". Series: System analysis, control and information technology]. Kharkiv, NTU "KhPI" Publ., 2017, no. 55 (1276), pp. 26–30.
15. Godlevskiy M. D., Goloskokova A. A., Burlakov G. O. Dinamichna model planuvannya rozvitku pidmnozhini protsesiv etalonnogo modeli zrilosti SPICE [A dynamic model for development planning of process subsets for the SPICE reference maturity model]. *Visnyk NTU "KhPI": zb. nauk. pr. Seriya: Sy'stemny'j analiz, upravlinnya ta informacijni tehnologiyi*. [Bulletin of NTU "KhPI". Series: System analysis, control and information technology]. Kharkiv, NTU "KhPI" Publ., 2020, no. 2 (4), pp. 10–16.

Надійшло (received) 25.10.2022

## Відомості про авторів / About the Authors

**Годлевський Михайло Дмитрович** – доктор технічних наук, професор, Національний технічний університет «Харківський політехнічний інститут», директор інституту комп'ютерних наук та інформаційних технологій; м. Харків, Україна; ORCID: <https://orcid.org/0000-0003-2872-0598>; e-mail: [god\\_asu@kpi.kharkov.ua](mailto:god_asu@kpi.kharkov.ua).

**Бурлаков Георгій Олександрович** – «SheerChain» ОУ, головний виконуючий директор; ORCID: <https://orcid.org/0000-0002-0423-6024>; e-mail: [george@sheerchain.com](mailto:george@sheerchain.com).

**Godlevskiy Mykhaylo Dmytrovych** – Doctor of Technical Sciences, Full Professor, National Technical University "Kharkiv Polytechnic Institute", Director of the Institute of Computer Science and Software Engineering; Kharkiv, Ukraine; ORCID: <https://orcid.org/0000-0003-2872-0598>; e-mail: [god\\_asu@kpi.kharkov.ua](mailto:god_asu@kpi.kharkov.ua).

**Burlakov Heorhii Oleksandrovych** – «SheerChain» ОУ, Chief Executive Officer; ORCID: <https://orcid.org/0000-0002-0423-6024>; e-mail: [george@sheerchain.com](mailto:george@sheerchain.com).