

УДК 330.4:004.031

*М. І. Зеленська,
асистент кафедри економічної кібернетики,
ДВНЗ "Українська академія банківської справи Національного банку України
І. Г. Голуб,
студент 5-го курсу кафедри економічної кібернетики,
ДВНЗ "Українська академія банківської справи Національного банку України*

ОСОБЛИВОСТІ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ УПРАВЛІННЯ ІТ-ПРОЕКТАМИ ІЗ ЗАСТОСУВАННЯМ МЕТОДІВ ЕКОНОМІКО-МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

M. Zelenska,
assistant professor, Chair of Economic Cybernetics,
State Higher Educational Institution "Ukrainian Academy of Banking of The National Bank of Ukraine"
I. Golub,
master student, Chair of Economic Cybernetics,
State Higher Educational Institution "Ukrainian Academy of Banking of The National Bank of Ukraine"

FEATURES OF THE DEVELOPMENT OF THE PROJECT MANAGEMENT SYSTEM USING
THE ECONOMIC-MATHEMATICAL MODELING

У статті розглянуто особливості інформаційних систем управління ІТ-проектами, визначено задачі, які вони мають вирішувати, та визначено ряд недоліків, притаманних існуючим системам. У якості одного з можливих шляхів їх усунення запропоновано застосування економіко-математичних методів і моделей при їх розробці. Виділено основні задачі, які можуть бути вирішені за допомогою методів економіко-математичного моделювання та представлено загальний алгоритм розв'язання задачі щодо визначення послідовності виконання робіт проекту та оптимального призначення співробітника на виконання певної задачі.

The article provides information about the features of project management systems, the problems they have to solve and the shortcomings of the existing systems. As one of the possible ways to improve them it was advised to use economic-mathematical methods and models when they are developed. The basic problems that can be solved using the methods of economic-mathematical modeling were defined and the ways to deal with the problem of determining the sequence of operations and the problem of optimal assignment of employee were offered.

Ключові слова: система управління ІТ-проектами, розробка системи управління ІТ-проектами, оптимізація процесу управління ІТ-проектами, метод Монте-Карло, метод PERT, задача про призначення.

Key words: project management system, development of the project management system, optimization of project management, Monte-Carlo method, PERT method, personnel-assignment problem.

ПОСТАНОВКА ПРОБЛЕМИ

Важко уявити сучасний світ без комп'ютерних технологій та широкого використання інформаційних систем. Центром будь-якої подібної системи є програмне забезпечення, що дозволяє вести ефективне управління певним процесом, робити складні розрахунки або автоматизувати окремі аспекти побуту. Кожне програмне забезпечення проходить довгий шлях від задумки до конкретної реалізації у вигляді ІТ-проекту.

Вчасність виконання проекту, понесені витрати та якість кінцевого продукту в значній мірі залежить від ефективності управління процесом його розробки. Однак існуючі на сьогодні інформаційні системи управління ІТ-проектами, серед яких Basecamp, Redmine, JIRA, TeamLab, TeamVox, Rallydev та багато інших, у більшості випадків позбавлені функціоналу, який би забезпечував аналіз та передбачення перебігу проекту. Тож пошук можливих шляхів удосконалення процесу управління ІТ-проектами сьогодні є однією із



Рис. 1. Use-case діаграма системи управління проектами

найбільш важливих задач, з якими стикаються спеціалісти галузі ІТ.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Особливостям процесу управління ІТ-проектами присвячені праці як вітчизняних, так і зарубіжних авторів. Серед них можна відзначити С. Снедакер, І. Селиховкіна, О. Анань-

єва, Т.К. Кравченка, Н.В. Молоткову, І.С. Сахарова, А.В. Ходиревську та ін. У той же час досліджень, пов'язаних з пошуком можливих шляхів поліпшення систем управління ІТ-проектами нами знайдено не було.

З огляду на це метою даної статті є розробка підходу до удосконалення створюваних інформаційних систем управління ІТ-проектами за рахунок застосування методів економіко-математичного моделювання.

Виклад основного матеріалу. Управління ІТ-проектами — досить нова галузь, яка постійно розвивається. Це методологія (процес) організації, планування, управління, координації трудових, фінансових і матеріально-технічних ресурсів протягом проектного циклу, направлена на ефективне досягнення його цілей шляхом використання сучасних методів, технік і технологій управління та досягнення зазначених в проекті результатів за складом та обсягом робіт, вартості, часу, якості та задоволеності учасників проекту [1, с. 18].

Важливість якісного управління ІТ-проектами впливає з особливостей таких проектів, а саме наявності великої кількості труднощів при їх розробці. Такими труднощами є:

— Складність. Проявляється у значно більшій мірі, ніж у будь-якій іншій галузі людської діяльності. Складність є сутністю програмних об'єктів і нелінійними темпами зростає разом зі зростанням їх обсягів;

— Узгодженість. Передбачає відсутність інформаційної асиметрії між усіма учасниками процесу розробки ПЗ. Неefektivність комунікацій, різне сприйняття системи та використання неперевіраних допущень її розробниками призводять до того, що окремі складові єдиного програмного комплексу, а також взаємозв'язок ПЗ із зовнішніми вимогами стає неузгодженим, що вимагає суттєвих витрат ресурсів на усунення даної проблеми.

— Здатність до змін. Видима в теорії легкість змін ПЗ на практиці виливається у істотні витрати

ресурсів, зростання складності та порушення концептуальної цілісності продукту.

— Незримість. На відміну від карт, схем, рисунків та інших засобів візуального абстрагування, які можуть бути використані для представлення більшості об'єктів матеріального і нематеріального світу, програмні об'єкти дуже слабо піддаються наглядній візуалізації. Прийняті на практиці нотації блок-схем, моделей даних, мови графічного

Таблиця 1. Порівняльна характеристика систем управління проектами

Назва функції	Basecamp	Redmine	JIRA	TeamLab	TeamBox
Можливість створення задач	+	+	+	+	+
Підтримка класів задач	+	+	+	+	+
Підтримка ієрархії задач	+/-	+/-	-	+/-	-
Можливість створювати зв'язки між задачами	-	+	-	-	+
Розрахунок часу закінчення задачі	-	-	-	+	-
Можливість створення подій	+	-	+	+	+
Наявність планувальника	+	-	+	+	+
Можливість автоматичного призначення задач	-	-	-	-	-
Аналіз поточного стану проекту	-	-	+/-	-	-
Попередження про можливі ризики	-	-	-	-	-
Підтримка візуалізації	+/-	+/-	-	+	+/-
Можливість обміну файлами	+	+	+	+	+
Можливість обміну повідомленнями	+	+	+	+	+
Можливість управління робочим часом співробітника	-	+	+	-	+
Можливість управління графіком співробітника	-	-	-	-	-
Підтримка гнучких методів розробки	+	+	+	+	+
Можливість працювати на сервері	+	+	+	+	+
Можливість працювати у хмарі	-	-	-	+	+
Ліцензія	GPL	GPL	Proprietary	GPL	AGPL

моделювання, такі як, наприклад, UML, не здатні достатньо ефективно вирішити проблему наглядного представлення структури ПЗ і принципів його функціонування [2, с.100—102].

Іншим аспектом необхідності якісного управління ІТ-проектами є їх ціна. Середня зарплата програміста за даними ресурсу dou.ua, однієї з найбільших спільнот людей, пов'язаних з інформаційними технологіями, дорівнює приблизно \$2800, а заробітна платня менеджера проектів перевищує \$4100 на місяць [3]. Ці суми зростають в рази, якщо брати більш розвинені країни. Тож можемо уявити загальні витрати, які понесе фірма при неправильному або недостатньо ефективному управлінні проектом.

Третім аспектом необхідності якісного управління проектами є значні проблеми при зриві строків. Так як при створенні програмних продуктів основним ресурсом є робота програмістів, то логічною дією при появі небезпеки провалу проекту повинне бути додавання додаткової робочої сили, але це не так. Якщо проект не вкладається в строки, то додавання додаткової робочої сили затримає проект ще більше [2, с. 22]. Це створює безвихідну ситуацію, коли час виконання проекту залежить лише від календарного часу, і аж ніяк не від співвідношення між людиною/годинами, та часом, що залишився для виконання проекту.

Саме подібні проблеми допомагає вирішити якісне управління проектом.

Застосування систем управління проектами дозволяє:

- знизити ймовірність невірної оцінки часу виконання проекту;
- проводити більш детальний контроль за виконанням проекту;
- зменшити вірогідність помилок в процесі управління проектом;
- зменшити витрати часу менеджера проектів, в наслідок чого він зможе здійснювати управління декількома проектами;
- зменшити час виконання проекту;
- завчасно передбачити несприятливий розвиток проекту.

Системи управління проектами у загальному випадку покликані допомогти оптимізувати роботу менеджера проекту і роботу усієї команди в цілому. Основними задачами, що вирішуються системами управління проектами, є: створення задач, створення подій, виконання завдань, контроль за виконанням задач, планування ходу виконання проекту, аналіз ходу виконання проекту, надання рекомендацій щодо управління проектом, управління робочим часом.

Реалізація зазначених функцій забезпечує менеджера проектами необхідними засобами для управління процесом перебігу того чи іншого проекту.

Окрім описаних функцій, подібні системи можуть містити й інший допоміжний функціонал, такий як: управління витратами, ведення рахунків та розрахунків, генерація звітності, візуалізація певних частин проекту, забезпечення обміну миттєвими повідомленнями, реалізація форуму та wiki, відеочат тощо.

Для забезпечення інтеграції з інфраструктурою компанії можуть бути присутні такі функціональні можливості: інтеграція з сервером підприємства, інтеграція з системами документообігу, інтеграція з системами фінансового контролю, робота у хмарі, інтеграція з мобільними пристроями, підтримка різних типів даних, у т.ч. специфічних для організації тощо.

Для більш наочної демонстрації функціональності системи управління проектами представимо можливі сценарії їх використання у вигляді UML Use-case діаграми (рис. 1).

На сучасному ринку представлена велика кількість систем управління ІТ проектами, серед яких: Basecamp, Redmine, JIRA, TeamLab, TeamBox, Rallydev та багато інших. Проведемо порівняльний аналіз п'яти найбільш популярних систем згідно з основними функціями, які вони мають виконувати.

Як видно з наведеної таблиці 1, найбільш популярні системи управління проектами не реалізують, або реалізують лише частково функціонал, пов'язаний з аналізом та передбаченням ходу проекту. На нашу думку, це може бути пов'язане з небажанням занадто ускладнювати систему та переобтяжувати її інтерфейс. Однак зазначені функції є ключовими для систем управління проектами, тож надалі розглянемо можливі шляхи їх реалізації.

У якості цілі оптимізації оберемо дві найбільш затратні та часті задачі, які вирішуються менеджером проекту:

- визначення найбільш важливих (критичних) робіт, тобто визначення послідовності виконання робіт;
- визначення члена команди, який виконає певну задачу найкраще.

Для вирішення першої задачі доцільно використовувати метод оцінки і перегляду планів (PERT), а точніше його варіацію з використанням симуляції Монте-Карло. Для вирішення другої задачі щодо оптимального призначення співробітників доцільно застосувати метод спряжених градієнтів.

Тоді загальна модель оптимізації буде мати вигляд:

$$t_m = \min \tag{1}$$

$$t_m = t_{\text{посл}} + t_{\text{призн}}$$

де t_m — витрати часу менеджера;

$t_{\text{посл}}$ — витрати часу менеджера на визначення послідовності виконання задач;

$t_{\text{призн}}$ — витрати часу менеджера на призначення члена команди на виконання задачі.

Метод оцінки і перегляду планів (PERT) є модифікацією методу критичного шляху для ймовірнісних моделей. Алгоритм методу є наступним:

— для кожної роботи необхідно визначити оптимістичну, песимістичну та вірогідну оцінку часу виконання роботи;

— врахувати очікуваний час за допомогою бета розподілу;

— розрахувати дисперсію;

— обчислити критичний шлях.

Зупинимось більш детально на методі обчислення критичного шляху.

Метод критичного шляху відноситься до моделей теорії обмежень. Він передбачає, що тривалість виконання кожної роботи є відома. В результаті використання цього методу можна визначити:

— мінімальний час виконання проекту;

— множину критичних та некритичних робіт;

— час, на який можна відкласти виконання некритичних робіт;

— час, у який повинні початися та закінчитися конкретні роботи.

Критичний шлях — найдовший шлях від вихідної до завершальної події. Критичний шлях характеризує мінімальну тривалість виконання всього комплексу робіт. Роботи та події, що лежать на критичному шляху, називаються критичними.

Нехай i, j — вершини або події проекту, (i, j) — робота проекту, s — подія "початок проекту", f — подія "закінчення проекту", T — довжина критичного шляху.

Введемо наступні змінні:

$t_{(i,j)}$ — час виконання роботи (i, j) ;

$ES_{(i,j)}$ — найбільш ранній час початку роботи (i, j) ;

$EF_{(i,j)}$ — найбільш ранній час закінчення роботи (i, j) ;

$LS_{(i,j)}$ — найбільш пізній час початку роботи (i, j) ;

$LF_{(i,j)}$ — найбільш пізній час закінчення роботи (i, j) ;

E_i — найбільш ранній час настання події i ;

L_i — найбільш пізній час настання події i ;

$R_{(i,j)}$ — повний резерв часу на виконання роботи (i, j) ; час, на який може бути відкладена робота (i, j) без збільшення тривалості виконання всього проекту;

$r_{(i,j)}$ — вільний резерв часу на виконання роботи (i, j) ; час, на який може бути відкладена робота (i, j) без збільшення найбільш раннього часу E_i настання наступної події.

Якщо (i, j) — робота проекту, то мають місце співвідношення:

— для будь-якого j $ES_{(i,j)} = E_j$;

— для будь-якого i $LF_{(i,j)} = L_i$;

Для знаходження критичного шляху, необхідно для кожної роботи (i, j) визначити найбільш ранній час початку і закінчення роботи ($ES_{(i,j)}$ і $EF_{(i,j)}$) і найбільш пізній час початку і закінчення роботи ($LS_{(i,j)}$ і $LF_{(i,j)}$).

Метод критичного шляху описується наступними співвідношеннями:

$$ES_{(s,j)} = 0 \quad (2),$$

для будь-якої роботи (s, j) , що виходить із стартової вершини s проекту.

$$EF_{(i,j)} = ES_{(i,j)} + t_{(i,j)} = E_i + t_{(i,j)} \quad (3),$$

тобто найбільш ранній час закінчення будь-якої роботи (i, j) перевищує найбільш ранній час початку цієї роботи (час настання попередньої події i) на час її виконання.

$$ES_{(q,j)} = \max EF_{(i,q)} \quad (4),$$

тобто найбільш ранній час початку роботи (q, j) дорівнює найбільшому із значень найбільш раннього часу закінчення безпосередньо передуючих їй робіт.

$$T = \max EF_{(i,f)} = E_f \quad (5),$$

тобто довжина критичного шляху дорівнює найбільш ранньому часу завершення проекту.

$$LF_{(i,j)} = T \quad (6),$$

тобто найбільш пізній час закінчення будь-якої роботи, що завершує проект, дорівнює довжині критичного шляху.

$$LS_{(i,j)} = LF_{(i,j)} - t_{(i,j)} = L_j - t_{(i,j)} \quad (7),$$

тобто найбільш пізній час початку будь-якої роботи менше найбільш пізнього часу закінчення цієї роботи (часу настання наступної події) на час її виконання.

$$LF_{(i,q)} = \min LS_{(q,j)} \quad (8),$$

тобто найбільш пізній час закінчення роботи (i, q) дорівнює найменшому із значень найбільш пізнього часу початку безпосередньо слідуєчих за нею робіт.

$$R_{(i,j)} = LS_{(i,j)} - ES_{(i,j)} = LF_{(i,j)} - EF_{(i,j)} = L_j - t_{(i,j)} - E_i \quad (9),$$

тобто повний резерв часу на виконання будь-якої роботи дорівнює різниці між найбільш пізнім і найбільш раннім часом її початку або різниці між найбільш пізнім і найбільш раннім часом її закінчення.

$$r_{(i,j)} = L_j - ES_{(i,j)} - t_{(i,j)} = L_j - EF_{(i,j)} = L_j - E_i - t_{(i,j)} \quad (10),$$

тобто вільний резерв часу на виконання будь-якої роботи дорівнює різниці між найбільш пізнім часом настання наступної події і найбільш раннім часом закінчення роботи.

З приведених вище визначень і співвідношень безпосередньо витікають наступні твердження:

— довжина критичного шляху дорівнює T ;

— якщо $R_{(i,j)} = 0$, то робота (i, j) лежить на критичному шляху;

— якщо $R_{(i,j)} > 0$, то робота (i, j) не лежить на критичному шляху;

— якщо час початку роботи (i, j) , що не лежить на критичному шляху, відкласти на термін менший, ніж $r_{(i,j)}$, то найбільш ранній час настання наступної події не зміниться;

— якщо час початку роботи (i, j) , що не лежить на критичному шляху, відкласти на термін менший, ніж $R_{(i,j)}$, то час, необхідний на виконання всього проекту не збільшиться [4, с. 156 — 180].

Метод PERT має серйозний недолік — пряму та сильну залежність від оцінок розробників, які в свою чергу можуть допускати помилки. Тому задля більшої надійності результатів доцільно провести рандомізацію алгоритму, а саме застосувати метод Монте-Карло для генерації вхідних даних. Це дозволить отримати більш достовірні результати завдяки аналізу не одного спостереження, а вибірки, тому за законом "Великих чисел" буде отримано результат, що є близьким до реального.

Метод статистичних випробувань (метод Монте-Карло) полягає у розгляді мережі в якості ймовірнісної моделі, на якій оцінки тривалості окремих робіт можуть приймати довільні значення в крайніх (мінімум і максимум) вказаних експертами межах, і навіть виходити за ці межі настільки, наскільки це дозволяють закони теорії ймовірності. Сутність методу статистичних випробувань полягає в отриманні на комп'ютері дуже великої кількості окремих реалізацій розглядуваної мережевої моделі, що відрізняються одна від одної тим, що тривалості робіт у всіх варіантах моделі випадково вибираються за законами, що характеризують розподіл тривалості кожної з окремих оцінок [5, с. 130].

Виходячи з наведеного опису, представимо загальну блок-схему комплексного методу (рис. 2).

Отже, завдяки використанню методів PERT та Монте-Карло маємо можливість визначення критичного шляху і



Рис. 2. Загальна блок-схема комплексного методу зі знаходження критичного шляху

як наслідок визначення найбільш оптимальної послідовності виконання робіт, що в свою чергу призведе до оптимізації витрат часу менеджера проекту на виконання цієї задачі.

Для визначення найбільш відповідного співробітника для виконання роботи використаємо класичну задачу про призначення.

Загальне формулювання задачі про призначення: потрібно виконати n видів робіт, на які претендують n кандидатів. Продуктивність праці (компетентність) i -го кандидата при виконанні j -ої роботи дорівнює

$p_{ij} (i, j = \overline{1, n})$. Кожен кандидат може бути призначений лише на одну роботу, і кожна робота має виконуватися лише одним кандидатом. Потрібно знайти оптимальне призначення кандидатів на виконання робіт, за якого сумарна продуктивність на виконання всіх робіт буде максимальною.

Нехай x_{ij} дорівнює одиниці, якщо i -ий кандидат виконує j -ту роботу, та дорівнює нулю в протилежному разі. Тоді умову, що кожен кандидат має виконувати лише одну ро-

боту, запишемо у вигляді: $\sum_{i=1}^n x_{ij} = 1 (j = \overline{1, n})$. Умова виконання кожної роботи лише одним кандидатом має вигляд: $\sum_{j=1}^n x_{ij} = 1 (i = \overline{1, n})$. Цільова функція має такий вигляд:

$F = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$. Отже, маємо таку математичну модель [6, с. 250]:

$$\max F = \sum_{i=1}^n \sum_{j=1}^n p_{ij} x_{ij}$$

$$\begin{cases} \sum_{j=1}^n x_{ij} = 1 (i = \overline{1, n}); \\ \sum_{i=1}^m x_{ij} = 1 (j = \overline{1, n}); \\ x_{ij} \in \{0; 1\} (i = \overline{1, n}; j = \overline{1, n}). \end{cases} \quad (11).$$

Для розв'язку даної системи будемо використовувати метод спряжених градієнтів.

Метод спряжених градієнтів — метод знаходження локального мінімуму (максимуму) функції на основі інформації про її значення і її градієнт.

Вектори A і B називають спряженими відносно матриці Q (чи Q -спряженими), якщо скалярний добуток векторів A і QB дорівнює нулю, тобто $A^T QB = 0$. Спряженість векторів є узагальненням поняттям ортогональності, оскільки, якщо $Q = E$, то Q -спряженість векторів означає їх ортогональність.

Нехай пошук починається в точці X_0 з початковим напрямком S_0 вибраним довільно. Тоді нова точка

$$X_1 = X_0 + hS_0, \quad (12)$$

де значення кроку h слід визначити за умови мінімуму функції $f(X_0 + hS_0)$ по h .

Для цього розкладемо функцію $f(X)$ у ряд Тейлора в околі точки $X = X_0$

$$f(X) \approx f(X_0) + \nabla f(X_0)^T (X - X_0) + \frac{1}{2} (X - X_0)^T H(X_0) (X - X_0) \quad (13),$$

$$\text{де } H(X_0) = \begin{bmatrix} \frac{\partial^2 f(X_0)}{\partial x_1^2} & \frac{\partial^2 f(X_0)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(X_0)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(X_0)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(X_0)}{\partial x_2^2} & \dots & \frac{\partial^2 f(X_0)}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f(X_0)}{\partial x_1 \partial x_n} & \frac{\partial^2 f(X_0)}{\partial x_2 \partial x_n} & \dots & \frac{\partial^2 f(X_0)}{\partial x_n^2} \end{bmatrix} -$$

матриця Гессе, тоді:

$$f(X_0 + hS_0) = f(X_0) + \nabla f(X_0)^T hS_0 + \frac{1}{2} hS_0^T H(X_0) hS_0.$$

З умови мінімуму останньої функції маємо:

$$\frac{df(X_0 + hS_0)}{dh} = \nabla f(X_0)^T S_0 + hS_0^T H(X_0)hS_0 = 0 \quad (14).$$

Звідси:

$$h_0 = -\frac{\nabla f(X_0)^T S_0}{S_0^T H(X_0)hS_0} \quad (15).$$

Можна показати, що після того, як за (12) і (15) обчислено точку X_p для продовження пошуку потрібно обрати новий напрям S_j , так, щоб:

$$S_i^T H(X_0)S_0 = 0. \quad (16),$$

тобто новий напрям S_j має бути спряженим до старого напрямку S_r .

Алгоритм загального випадку методу спряжених градієнтів буде таким:

— обчислити вектор $S_i = -\nabla f(X_i)S_0$;

— знайти мінімум $f(X)$ одним з методів одновимірного пошуку в напрямі S_r . Звідси знаходимо $X_{i+1}, \nabla f(X_{i+1})$;

— визначити новий спряжений напрям S_{i+1} із співвідношення:

$$S_{i+1} = -\nabla f(X_{i+1}) + S_i \frac{\nabla f(X_{i+1})^T \nabla f(X_{i+1})}{f(X_i)^T \nabla f(X_i)} \quad (17);$$

— знайти мінімум $f(X)$ одним з методів одновимірного пошуку в напрямі S_{i+1} ;

— якщо умова достатності пошуку не виконується, то перейти до кроку 1 [7, с. 185—188].

Блок-схема методу оптимального призначення співробітника для виконання певної задачі матиме вигляд, представлений на рисунку 3.

Таким чином, при розробці інформаційних систем управління проектами доцільно використовувати метод PERT та Монте-Карло для оптимізації вибору найбільш пріоритетних задач та метод спряжених градієнтів і задачу про призначення для визначення співробітника, що найкраще виконає поставлену задачу.

ВИСНОВКИ

У результаті дослідження можемо зробити висновок, що одним із можливих шляхів підвищення ефективності управління процесом розробки ПЗ є створення інформаційних систем управління проектами, які поєднують у собі риси класичних систем подібного типу, принципи гнучких методів розробки та математичні методи. На нашу думку, ключовими задачами, які можуть бути вирішені за допомогою застосування методів економіко-математичного моделювання, є визначення послідовності виконання робіт та оптимізація процесу розподілу задач між співробітниками.

Література:

1. Мазур І.І., Шапіро В.Д., Ольдерогге Н.Г. Управління проектами, навч. посібник. — Москва, 2004 р, Омега-Л. — 405 с.
2. Frederic Brooks, The Mythical Man Month. — Addison-Wesley Professional, Anniversary edition, 1995. — 336 с.
3. Зарплати програмістів [Електронний ресурс]. — Режим доступу: <http://jobs.dou.ua/salaries/>
4. Хілл П. Наука та мистецтво проектування. Методи проектування та наукове обґрунтування рішень. — М.: Мир, 1973. — 264 с.
5. Кобелев Н.Б. Основи імітаційного моделювання складних економічних систем / Н.Б. Кобелев. — М.: Дело, 2003. — 336 с.
6. Наконечний С.І., Савіна С.С. Математичне програмування: навч. посібн. — К.: КНЕУ, 2003. — 452 с.



Рисунок 3. Загальна блок-схема комплексного методу оптимального призначення співробітника для виконання певної задачі

7. Прокопенко Ю.В., Татарчук Д.Д., Казіміренко В.А. Обчислювальна математика. — К.: Політехніка, 2011. — 223 с.

References:

1. Mazur, I.I. Shapiro, V.D. and Ol'derohge, N.H. (2004), Upravlinnia proektamy [Project management], Omeha-L, Moskva, Russia.
 2. Brooks, F. (1995), The Mythical Man Month, Addison-Wesley, Boston, USA.
 3. Programmers community (2014), "Salaries of IT-specialists", available at: <http://jobs.dou.ua/salaries/> (Accessed at 17 May 2013).
 4. Khill, P. (1973), Nauka ta mystetstvo proektuvannia. Metody proektuvannia ta naukove obhruntuvannia rishen' [Science and art of modeling. Methods of modeling and scientific reasoning of decisions], Myr, Moscow, Russia.
 5. Koblelev, N.B. (2003), Osnovy imitatsijnoho modeljuvannia skladnykh ekonomichnykh system [Basics of imitational modeling of complex economic systems], Delo, Moscow, Russia.
 6. Nakonechnyj, S.I. and Savina, S.S. (2003), Matematyche prohramuvannia: navch. posib [Mathematical programming: study guide], KNEU, Kyiv, Ukraine.
 7. Prokopenko, Yu.V. Tatarchuk, D.D. and Kazimirenko, V.A. (2011), Obchysliuval'na matematyka [Calculus mathematics], Politekhnika, Kyiv, Ukraine.
- Стаття надійшла до редакції 19.01.2014 р.