

Т.В. Селівьорстова, Д.Г. Зеленцов, Я.В. Романько, І.Р. Тесленко

**СТОХАСТИЧНА МЕРЕЖЕВО-ЧАСОВА МОДЕЛЬ PERT/CPM  
З ІМОВІРНІСНИМИ ПЕРЕРОБКАМИ  
ДЛЯ ОЦІНЮВАННЯ СТРОКІВ ІТ-ПРОЄКТИ**

*Анотація.* У статті розглянуто задачу імовірнісного оцінювання строків завершення ІТ-проектів за умов невизначеності тривалостей робіт і повторних циклів доопрацювання результатів після контрольних перевірок (*rework*). Запропоновано стохастичну мережево-часову модель на основі PERT/CPM, у якій базові тривалості робіт задаються триточковими оцінками PERT, а механізм переробок формалізовано через ймовірність непроходження контрольної етапу та випадкову тривалість переробки (з можливістю врахування ефекту навчання). Для отримання розподілу часу завершення проекту застосовано імітаційне моделювання методом Монте-Карло з мережевим розрахунком ранніх термінів старту/завершення. Обчислюються перцентильні строки завершення (P50/P80/P90), ймовірність виконання заданого дедлайну та показники критичності робіт і чутливості параметрів (*tornado*-аналіз). Запропонований підхід забезпечує більш адекватне оцінювання календарних ризиків порівняно з детермінованими планами та може бути використаний для обґрунтування буферів і пріоритетів управління якістю й ризиками в ІТ-проектах.

*Ключові слова:* PERT, CPM, стохастичне планування, імітаційне моделювання, Монте-Карло, переробки (*rework*), ризик дедлайну, критичний шлях, *criticality index*, чутливість.

**Вступ.** Сучасні ІТ-проекти характеризуються високою динамічністю вимог, складною взаємодією команд і підсистем, а також істотною невизначеністю щодо фактичної тривалості робіт. Навіть за наявності формалізованого календарного плану типу CPM/PERT, реальні строки виконання часто відхиляються від детермінованих оцінок через повторні цикли уточнення, виправлень і повторного приймання результатів. До найбільш поширених причин належать переробки після *code review*, невдалі прогони тестування (QA), повторні ітерації користувацького приймання (UAT), а також додаткові перевірки безпеки й якості. Унаслідок цього, проектні графіки, побудовані на фіксованих тривалостях робіт, здебільшого недооцінюють ризик зриву дедлайнів і не дають коректних імовірнісних характеристик завершення.

Класичний підхід PERT передбачає моделювання тривалостей робіт випадковими величинами (зокрема за триточковими оцінками), а CPM забезпечує обчислення критичного шляху та резервів часу в мережевому графі залежностей. Проте у традиційних

постановках ці методи, як правило, не враховують явний механізм зворотного зв'язку «виконання–перевірка–переробка», який є типовим для процесів розроблення програмного забезпечення. Практика DevOps/CI/CD та сучасні стандарти якості приводять до того, що результат кожної значущої активності проходить одну або кілька контрольних точок, де з певною ймовірністю повертається на доопрацювання. Такий механізм породжує стохастичні цикли переробок, що змінюють як середнє значення тривалості робіт, так і форму розподілу строків завершення проєкту, зокрема верхні перцентилі (P80/P90), які є критично важливими для управління зобов'язаннями та контрактними дедлайнами.

У зв'язку з цим актуальною є побудова математичної моделі, яка поєднує мережеве планування CPM/PERT із імовірнісним описом переробок, дозволяючи перейти від “точкової” дати завершення до повного розподілу строків і оцінки ризику дедлайну. У даній роботі пропонується стохастична мережево-часова модель, у якій для кожної роботи базова тривалість задається PERT-розподілом, а процес переробок після контрольної перевірки моделюється геометричною кількістю повторів та випадковими тривалостями доопрацювання. Такий підхід дає змогу адекватно описувати реальні проєктні цикли та формувати імовірнісні характеристики завершення, включно з оцінюванням перцентильних строків (P50/P80/P90) і ймовірності виконання заданого дедлайну.

**Метою статті** є розроблення та дослідження стохастичної PERT/CPM-моделі з імовірнісними переробками для оцінювання строків завершення IT-проєктів, а також формування обчислювальної процедури отримання розподілу часу завершення на основі імітаційного моделювання.

### **Постановка задачі та припущення моделі.**

#### **1. Мережеве подання IT-проєкту**

Розглядається IT-проєкт, структура якого задається орієнтованим ациклічним графом (мережевою моделлю)

$$G = (V, E),$$

де  $V = \{1, 2, \dots, n\}$  – множина робіт (активностей), а  $E \subseteq V \times V$  – множина технологічних залежностей типу *finish-to-start*. Наявність дуги  $(i, j) \in E$  означає, що робота  $j$  може розпочатися лише після завершення роботи  $i$ . Така постановка відповідає класичним підходам CPM/PERT і використовується для розрахунку ранніх термінів старту/завершення та критичного шляху.

Для кожної роботи  $i \in V$  вводяться моменти старту  $S_i$  та завершення  $F_i$ , а також її тривалість (в подальшому – стохастична)  $T_i$ . Початок проєкту задається як  $S_0 = 0$ , а час завершення проєкту визначається моментом завершення останньої (за топологічним порядком) роботи:

$$T_{\text{proj}} = \max_{i \in V} F_i.$$

#### **2. Стохастичні тривалості базового виконання робіт (PERT-компонента).**

Невизначеність тривалості виконання робіт моделюється випадковими величинами. Нехай  $D_i$  – базова тривалість виконання роботи  $i$  без урахування можливих переробок. У PERT-підході  $D_i$  параметризується триточковими оцінками:

$$(a_i, m_i, b_i),$$

де  $a_i$  – оптимістична,  $m_i$  – найбільш імовірна,  $b_i$  – песимістична тривалість. У загальному вигляді припускається, що

$$D_i \sim \text{PERT}(a_i, m_i, b_i),$$

або інший близький за формою розподіл (наприклад трикутний), що дозволяє відображати асиметрію та обмеженість тривалостей у практиці планування.

### 3. Механізм контрольних перевірок та переробок (rework).

Особливістю ІТ-проектів є наявність контрольних точок якості/приймання (код-рев'ю, QA-тестування, UAT, acceptance), після яких результат може бути повернений на доопрацювання. Для відображення цього механізму для кожної роботи  $i$  вводиться множина контрольних етапів (gate):

$$g \in \{1, 2, \dots, G_i\}.$$

Кожен gate  $g$  для роботи  $i$  характеризується:

- $p_{i,g} \in [0,1]$  – ймовірністю непроходження gate (виникнення переробки);
- $R_{i,g}$  – випадковою тривалістю однієї переробки, яка також може задаватися триточковими оцінками  $(a_{i,g}^r, m_{i,g}^r, b_{i,g}^r)$ .

Нехай  $N_{i,g}$  – кількість переробок, спричинених gate  $g$  для роботи  $i$ . Приймається геометрична модель повторів (кількість невдач до першого успішного проходження перевірки):

$$\mathbb{P}(N_{i,g} = n) = (1 - p_{i,g}) p_{i,g}^n, \quad n = 0, 1, 2, \dots$$

тобто

$$N_{i,g} \sim \text{Geom}(1 - p_{i,g}).$$

Ця постановка відображає повторюваний цикл «виконання/перевірка/переробка», який з певною ймовірністю завершується успішним прийманням.

### 4. Ефект навчання в переробках.

У практиці розроблення програмного забезпечення повторні переробки зазвичай виконуються швидше (внаслідок уточнення вимог, локалізації дефектів, стабілізації оточення). Для врахування цього вводиться параметр навчання  $\alpha_{i,g} \in (0,1]$ , який масштабує тривалість  $k$ -го повтору:

$$R_{i,g,k} = \alpha_{i,g}^{k-1} R_{i,g}, \quad k = 1, 2, \dots$$

де  $R_{i,g,k}$  – тривалість  $k$ -го циклу переробки. При  $\alpha_{i,g} = 1$  ефект навчання відсутній; при  $\alpha_{i,g} < 1$  повторні переробки мають меншу очікувану тривалість.

### 5. Ефективна тривалість роботи та правила мережевого розрахунку.

Ефективна тривалість роботи  $i$  визначається як сума базової тривалості та тривалостей переробок на всіх gate:

$$T_i = D_i + \sum_{g=1}^{G_i} \sum_{k=1}^{N_{i,g}} R_{i,g,k}.$$

У випадку моделі навчання замість  $R_{i,g,k}$  використовується залежність з  $\alpha_{i,g}$ , наведена вище.

Подальший розрахунок календарних характеристик у мережі виконується за стандартними правилами СРМ (прямий прохід), але з випадковими  $T_i$ :

$$S_i = \max_{(j,i) \in E} F_j, F_i = S_i + T_i.$$

причому для робіт без попередників приймається  $S_i = 0$ .

### 6. Мета моделювання та вихідні показники.

Метою є отримання імовірнісної оцінки строків завершення проєкту з урахуванням стохастичності базових тривалостей і циклів переробок. Основними вихідними показниками є:

1) Розподіл часу завершення проєкту  $T_{\text{proj}}$ .

2) Перцентильні оцінки строків:

$$P_q(T_{\text{proj}}), q \in \{0.5, 0.8, 0.9\},$$

що відповідають  $P50$ ,  $P80$ ,  $P90$  датам завершення.

3) Ймовірність виконання дедлайну  $T^*$ :

$$\mathbb{P}(T_{\text{proj}} \leq T^*).$$

4) (За необхідності) показники критичності робіт, зокрема частота перебування роботи на критичному шляху (criticality index), що дозволяє ідентифікувати роботи-драйвери ризику.

### 7. Основні припущення та обмеження.

Для забезпечення формальної коректності та обчислювальної реалізованості моделі приймаються такі припущення:

1) Ациклічність мережі. Граф  $G$  є DAG; залежності робіт не змінюються протягом моделювання.

2) Незалежність спроб проходження gate. Кожна спроба проходження gate  $g$  для роботи  $i$  є незалежною, а ймовірність непроходження  $p_{i,g}$  є сталою в межах моделювання.

3) Незалежність тривалостей. Величини  $D_i$ ,  $R_{i,g,k}$ , та  $N_{i,g}$  розглядаються як незалежні (або умовно незалежні за фіксованих параметрів), якщо не зазначено інше.

4) Однорідність переробок одного типу. Для фіксованих  $i$ ,  $g$  тривалості  $R_{i,g,k}$  мають однаковий базовий розподіл (з можливим масштабуванням за  $\alpha_{i,g}$ ).

5) Обрізання надмірних повторів (практичне обмеження). Для уникнення нерелістичних “нескінченних” циклів може вводиться максимальна кількість переробок  $N_{\max}$  gate:

$$N_{i,g} \leftarrow \min\{N_{i,g}, N_{\max}\}.$$

б) Ресурсні обмеження не враховано. У базовій постановці не моделюються обмеження ресурсів (людських/обчислювальних), мультизадачність та черги; у розділі висновків можуть бути наведені напрями розширення моделі на RCPSP-подібні постановки.

Єдина шкала часу. Усі тривалості вимірюються в одних одиницях (дні/тижні); за потреби календаризація виконується після отримання  $T_{\text{proj}}$ .

**Стохастична мережево-часова модель PERT/CPM з імовірнісними переробками.**

### 1. Позначення та загальна структура моделі.

Нехай IT-проект заданий мережею  $G = (V, E)$ ,  $V = \{1, \dots, n\}$ . Для кожної роботи:

- $D_i$  – базова тривалість виконання роботи без переробок;
- $N_{i,g}$  – кількість переробок, викликаних контрольним етапом (gate)  $g \in \{1, \dots, G_i\}$ ;
- $R_{i,g,k}$  – тривалість  $k$ -го циклу переробки на gate  $g$ .

Ефективна тривалість роботи з урахуванням переробок задається

$$T_i = D_i + \sum_{g=1}^{G_i} \sum_{k=1}^{N_{i,g}} R_{i,g,k}.$$

Таким чином, стохастика моделі формується двома складовими:

- невизначеністю базових тривалостей  $D_i$ ;
- випадковим процесом повторних переробок, параметризованим  $(p_{i,g}, R_{i,g})$  та, за потреби,  $\alpha_{i,g}$ .

### 2. PERT-модель базових тривалостей робіт.

Для кожної роботи  $i$  базова тривалість  $D_i$  задається триточковою оцінкою  $(a_i, m_i, b_i)$ . У рамках PERT припускається

$$D_i \sim \text{PERT}(a_i, m_i, b_i).$$

а математичне сподівання та дисперсія (PERT-наближення) визначаються:

$$\mathbb{E}[D_i] = \frac{a_i + 4m_i + b_i}{6}, \quad \text{Var}(D_i) = \left(\frac{b_i - a_i}{6}\right)^2.$$

За необхідності допускається використання альтернативних розподілів (наприклад трикутного), однак у подальшому викладі використовується стандартна PERT-параметризація.

### 3. Геометрична модель кількості переробок після gate.

Після завершення роботи  $i$  на кожному контрольному етапі  $g$  результат або приймається, або повертається на доопрацювання. Вводиться параметр

$$p_{i,g} \in [0,1),$$

що означає ймовірність непроходження gate  $g$  (тобто необхідність переробки).

Кількість переробок  $N_{i,g}$  моделюється геометричним розподілом (кількість невдач до першого успіху):

$$\mathbb{P}(N_{i,g} = n) = (1 - p_{i,g}) p_{i,g}^n, \quad n = 0, 1, 2, \dots$$

тобто

$$N_{i,g} \sim \text{Geom}(1 - p_{i,g}).$$

Звідси випливають моменти:

$$\mathbb{E}[N_{i,g}] = \frac{p_{i,g}}{1 - p_{i,g}}, \quad \text{Var}(N_{i,g}) = \frac{p_{i,g}}{(1 - p_{i,g})^2}.$$

Для запобігання нереалістично великим кількостям повторів у прикладних розрахунках може застосовуватися обрізання:

$$N_{i,g}^{\text{tr}} = \min\{N_{i,g}, N_{\max}\}.$$

#### 4. Модель тривалості переробок.

Тривалість одного циклу переробки на gate  $g$  для роботи  $i$  задається випадковою величиною  $R_{i,g}$ , параметризованою триточковою оцінкою  $(a_{i,g}^r, m_{i,g}^r, b_{i,g}^r)$ :

$$R_{i,g} \sim \text{PERT}(a_{i,g}^r, m_{i,g}^r, b_{i,g}^r),$$

а

$$\mathbb{E}[R_{i,g}] = \frac{a_{i,g}^r + 4m_{i,g}^r + b_{i,g}^r}{6}, \quad \text{Var}(R_{i,g}) = \left( \frac{b_{i,g}^r - a_{i,g}^r}{6} \right)^2.$$

Надалі припускається, що  $R_{i,g,k} \stackrel{iid}{\sim} R_{i,g}$  для фіксованих  $i, g$ .

#### 5. Ефект навчання в циклах переробок.

Для врахування скорочення тривалості повторних переробок вводиться параметр навчання

$$\alpha_{i,g} \in (0,1].$$

Тоді тривалість  $k$ -го повтору моделюється як

$$R_{i,g,k} = \alpha_{i,g}^{k-1} R_{i,g}, \quad k = 1, 2, \dots$$

і сумарний вклад переробок на gate  $g$  становить

$$W_{i,g} = \sum_{k=1}^{N_{i,g}} \alpha_{i,g}^{k-1} R_{i,g}.$$

Отже, ефективна тривалість роботи:

$$T_i = D_i + \sum_{g=1}^{G_i} W_{i,g} = D_i + \sum_{g=1}^{G_i} R_{i,g} \sum_{k=0}^{N_{i,g}-1} \alpha_{i,g}^k = D_i + \sum_{g=1}^{G_i} R_{i,g} \frac{1 - \alpha_{i,g}^{N_{i,g}}}{1 - \alpha_{i,g}}.$$

При  $\alpha_{i,g} = 1$  формула переходить у випадок без навчання:  $W_{i,g} = \sum_{k=1}^{N_{i,g}} R_{i,g,k}$ .

## 6. Математичне сподівання ефективної тривалості роботи.

За умов незалежності  $D_i$ ,  $N_{i,g}$  та  $R_{i,g}$  отримуємо сподівання.

### 6.1. Випадок без навчання ( $\alpha_{i,g} = 1$ )

Оскільки

$$T_i = D_i + \sum_{g=1}^{G_i} \sum_{k=1}^{N_{i,g}} R_{i,g,k},$$

то

$$\mathbb{E}[T_i] = \mathbb{E}[D_i] + \sum_{g=1}^{G_i} \mathbb{E}[N_{i,g}] \mathbb{E}[R_{i,g}] = \mathbb{E}[D_i] + \sum_{g=1}^{G_i} \frac{p_{i,g}}{1-p_{i,g}} \mathbb{E}[R_{i,g}].$$

Таким чином,

$$\mathbb{E}[T_i] = \mathbb{E}[D_i] + \sum_{g=1}^{G_i} \frac{p_{i,g}}{1-p_{i,g}} \mathbb{E}[R_{i,g}].$$

### 6.2. Випадок з навчанням ( $\alpha_{i,g} \in (0,1]$ )

Тоді

$$T_i = D_i + \sum_{g=1}^{G_i} R_{i,g} \frac{1-\alpha_{i,g}^{N_{i,g}}}{1-\alpha_{i,g}}.$$

Для  $N_{i,g} \sim \text{Geom}(1-p_{i,g})$  справедливо:

$$\mathbb{E}[\alpha_{i,g}^{N_{i,g}}] = \sum_{n=0}^{\infty} \alpha_{i,g}^n (1-p_{i,g}) p_{i,g}^n = \frac{1-p_{i,g}}{1-\alpha_{i,g} p_{i,g}}, \quad \alpha_{i,g} p_{i,g} < 1.$$

Тоді

$$\mathbb{E}\left[\frac{1-\alpha_{i,g}^{N_{i,g}}}{1-\alpha_{i,g}}\right] = \frac{1-\mathbb{E}[\alpha_{i,g}^{N_{i,g}}]}{1-\alpha_{i,g}} = \frac{p_{i,g}}{1-\alpha_{i,g} p_{i,g}}.$$

Отже,

$$\mathbb{E}[T_i] = \mathbb{E}[D_i] + \sum_{g=1}^{G_i} \frac{p_{i,g}}{1-\alpha_{i,g} p_{i,g}} \mathbb{E}[R_{i,g}].$$

що узагальнює формулу без навчання (при  $\alpha_{i,g} \rightarrow 1$ ).

## 7. Дисперсія ефективної тривалості (наближена оцінка).

Для випадку без навчання та незалежних компонент використовується стандартна формула для випадкової суми:

$$\text{Var}\left(\sum_{k=1}^{N_{i,g}} R_{i,g,k}\right) = \mathbb{E}[N_{i,g}] \text{Var}(R_{i,g}) + \text{Var}(N_{i,g}) (\mathbb{E}[R_{i,g}])^2.$$

Тоді

$$\text{Var}(T_i) \approx \text{Var}(D_i) + \sum_{g=1}^{G_i} \left[ \mathbb{E}[N_{i,g}] \text{Var}(R_{i,g}) + \text{Var}(N_{i,g}) (\mathbb{E}[R_{i,g}])^2 \right],$$

або з урахуванням  $\mathbb{E}[N_{i,g}]$  та  $\text{Var}(N_{i,g})$ :

$$\text{Var}(T_i) \approx \text{Var}(D_i) + \sum_{g=1}^{G_i} \left[ \frac{P_{i,g}}{1-P_{i,g}} \text{Var}(R_{i,g}) + \frac{P_{i,g}}{(1-P_{i,g})^2} (\mathbb{E}[R_{i,g}])^2 \right].$$

Наведена оцінка застосовується для аналітичної інтерпретації впливу параметрів переробок на варіативність строків.

### 8. Стохастичні рівняння СРМ та час завершення проєкту.

Для реалізації мережевого розрахунку вводяться ранні моменти старту/фінішу:

$$S_i = \max_{(j,i) \in E} F_j, \quad F_i = S_i + T_i.$$

Тоді час завершення проєкту:

$$T_{\text{proj}} = \max_{i \in V} F_i.$$

Еквівалентно, з позиції шляхів у мережі:

$$T_{\text{proj}} = \max_{\pi \in \Pi} \sum_{i \in \pi} T_i,$$

де  $\Pi$  – множина всіх шляхів від стартових до фінальних робіт. На відміну від детермінованої СРМ-моделі, в стохастичному випадку критичний шлях може змінюватися між реалізаціями (симуляціями), що зумовлює потребу в імовірнісних показниках критичності та аналізі чутливості.

### 9. Формулювання задачі імовірнісного оцінювання строків.

Задача полягає в обчисленні (або оцінюванні) розподілу  $T_{\text{proj}}$  та функціоналів цього розподілу:

- перцентильних оцінок  $P_q(T_{\text{proj}})$ , зокрема  $q = 0.5, 0.8, 0.9$ ;
- ймовірності виконання дедлайну  $T^* : \mathbb{P}(T_{\text{proj}} \leq T^*)$ .

Через наявність оператора максимуму по шляхах та стохастичних переробок замкнена аналітична форма розподілу  $T_{\text{proj}}$  у загальному випадку відсутня, тому у наступному розділі розглядається імітаційна процедура Монте-Карло для оцінювання зазначених показників, а також метрики criticality index і аналіз чутливості параметрів  $(p_{i,g}, \mathbb{E}[R_{i,g}], \alpha_{i,g})$ .

### Імітаційне моделювання та обговорення результатів.

#### 1. Мета та загальна схема імітаційного оцінювання.

Оскільки в запропонованій стохастичній мережево-часовій моделі час завершення проєкту  $T_{\text{proj}}$  визначається як максимум по завершеннях робіт мережі, а тривалості робіт включають випадкову суму переробок, аналітичне отримання розподілу  $T_{\text{proj}}$  у загальному випадку є складним. Тому для оцінювання розподілу строків завершення та

пов'язаних показників ризику застосовано імітаційне моделювання методом Монте-Карло.

Імітаційний підхід полягає у багаторазовій генерації реалізацій тривалостей робіт з урахуванням переробок та виконанні мережевого розрахунку СРМ для кожної реалізації. У результаті формується вибірка  $\{T_{\text{proj}}^{(s)}\}_{s=1}^S$ , яка апроксимує розподіл часу завершення проекту. На основі вибірки обчислюються перцентилі (наприклад,  $P50$ ,  $P80$ ,  $P90$ ), ймовірність виконання дедлайну, а також показники критичності робіт і чутливості до параметрів переробок.

## 2. Алгоритм Монте-Карло для стохастичної СРМ/PERT-мережі з переробками.

Нехай кількість ітерацій моделювання дорівнює  $S$ . Для кожної ітерації  $s \in \{1, \dots, S\}$  виконується:

1) Генерація базових тривалостей робіт. Для кожної роботи  $i \in V$  генерується значення  $D_i^{(s)}$  відповідно до PERT-розподілу, параметризованого триточковими оцінками  $(a_i, m_i, b_i)$ .

2) Генерація переробок на контрольних етапах. Для кожного gate  $g \in \{1, \dots, G_i\}$  генерується кількість переробок  $N_{i,g}^{(s)}$  за геометричним законом із параметром  $p_{i,g}$ . Далі генеруються тривалості переробок  $R_{i,g,k}^{(s)}$  (або обчислюється сумарний вклад із урахуванням ефекту навчання  $\alpha_{i,g}$ ). Сумарний вклад переробок по gate формує величину  $W_{i,g}^{(s)}$ , а повна тривалість роботи:

$$T_i^{(s)} = D_i^{(s)} + \sum_{g=1}^{G_i} W_{i,g}^{(s)}.$$

3) Мережевий розрахунок СРМ (прямий прохід). Використовуючи топологічний порядок графа  $G$ , обчислюються ранні моменти старту й завершення для всіх робіт:

$$S_i^{(s)} = \max_{(j,i) \in E} F_j^{(s)}, \quad F_i^{(s)} = S_i^{(s)} + T_i^{(s)}.$$

4) Фіксація часу завершення проекту. Для кожної ітерації:

$$T_{\text{proj}}^{(s)} = \max_{i \in V} F_i^{(s)}.$$

5) Агрегування результатів. Після виконання  $S$  ітерацій отримують вибірку  $\{T_{\text{proj}}^{(s)}\}$ , за якою оцінюються перцентилі, ймовірність дедлайну та інші метрики.

Зазначимо, що збільшення  $S$  підвищує точність оцінок перцентилів і ймовірностей. У практичних експериментах достатньо  $S$  порядку  $10^4 - 2 \cdot 10^4$  для стабільних оцінок  $P50$ ,  $P80$ ,  $P90$  у типових мережах середнього розміру.

## 3. Показники результативності та ризику.

Для аналізу календарних ризиків використовуються такі показники.

1) Перцентильні строки завершення:

$$P_q = \text{Quantile}_q(T_{\text{proj}}), \quad q \in \{0.5, 0.8, 0.9\}.$$

Зокрема,  $P_{80}$  відповідає даті, до якої проєкт завершується з імовірністю 0.8, і є практично релевантним орієнтиром для управлінських зобов'язань.

2) Ймовірність виконання дедлайну  $T^*$ :

$$P_{\text{deadline}} = \mathbb{P}(T_{\text{proj}} \leq T^*),$$

яка оцінюється часткою симуляцій, у яких  $T_{\text{proj}}^{(s)} \leq T^*$ .

3) Критичність робіт (Criticality Index). Через стохастичність тривалостей критичний шлях може змінюватися між ітераціями. Тому для кожної роботи  $i$  обчислюється показник:

$$CI_i = \frac{1}{S} \sum_{s=1}^S 1\{i \in \text{критичною в ітерації } s\}.$$

Високі значення  $CI_i$  ідентифікують “критичне ядро” мережі — роботи, які найчастіше визначають строк завершення.

4) Локальна чутливість (one-at-a-time) та tornado-аналіз. Для ключових параметрів  $\theta \in \{p_{i,g}, \mathbb{E}[R_{i,g}], \alpha_{i,g}\}$  виконується варіювання  $\theta$  на малу відносну величину (наприклад  $\pm 10\%$ ), після чого оцінюється зміна  $P_{80}$ . Це дозволяє ранжувати фактори, що найбільше впливають на ризикові строки.

#### 4. Візуалізація результатів.

Для інтерпретації імітаційних даних застосовано такі графіки.

1) S-крива (емпірична функція розподілу, ECDF) для  $T_{\text{proj}}$  (рисунок 1). Вона дозволяє візуально оцінювати  $P_{50}/P_{80}/P_{90}$  та ймовірність виконання дедлайну як значення ECDF у точці  $T^*$ . S-крива також демонструє ширину “зони невизначеності” – інтервал, у якому крива зростає найшвидше.

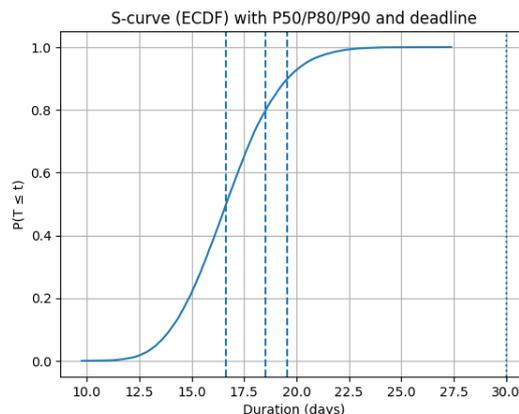


Рисунок 1 – S-крива (емпірична функція розподілу, ECDF) тривалості завершення проєкту  $T_{\text{proj}}$  з позначенням перцентилів  $P_{50}/P_{80}/P_{90}$  та дедлайну  $T^*$

2) Залежність  $P_{\text{deadline}}$  від імовірності переробок  $p$  (рисунок 2). Такий графік показує, наскільки дедлайн є чутливим до зростання частоти відхилень на певному gate

(наприклад review чи QA). Якщо крива наближається до 1 на всьому діапазоні  $p$ , це означає, що дедлайн має надлишковий запас і не є обмежувальним.

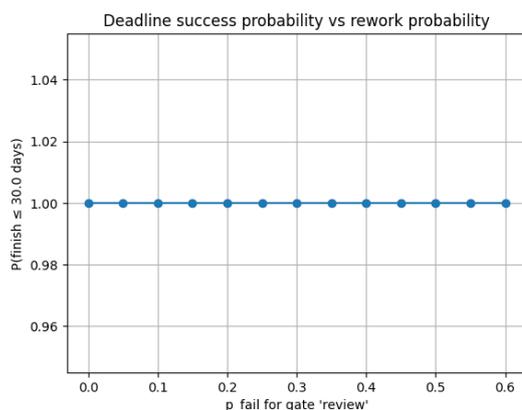


Рисунок 2 – Залежність ймовірності виконання дедлайну  $\mathbb{P}(T_{\text{proj}} \leq T^*)$  від імовірності переробки  $p_{i,g}$  на контрольному етапі review (за фіксованого  $T^* = 30$  днів)

3) Залежність  $P80$  від параметра навчання  $\alpha$  (рисунок 3). Монотонне зростання  $P80$  при  $\alpha \rightarrow 1$  підтверджує адекватність моделі: слабший ефект навчання збільшує верхні перцентилі строків.

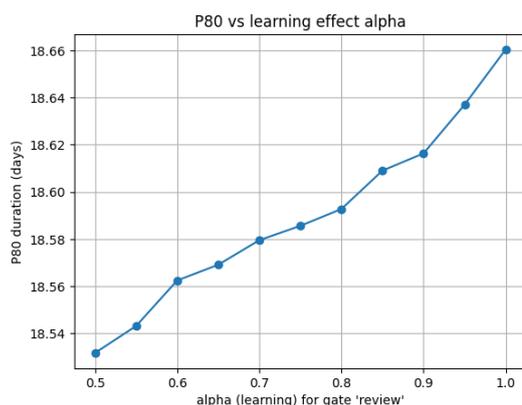


Рисунок 3 – Залежність перцентиля  $P80$  тривалості завершення проекту  $T_{\text{proj}}$  від параметра навчання  $\alpha$  для контрольного етапу review

4) Tornado-діаграма чутливості  $P80$  (рисунок 4). Вона дозволяє кількісно порівняти, які параметри (наприклад  $p_{i,g}$  для QA/UAT чи масштаб переробок у review) є основними драйверами зміщення  $P80$ .

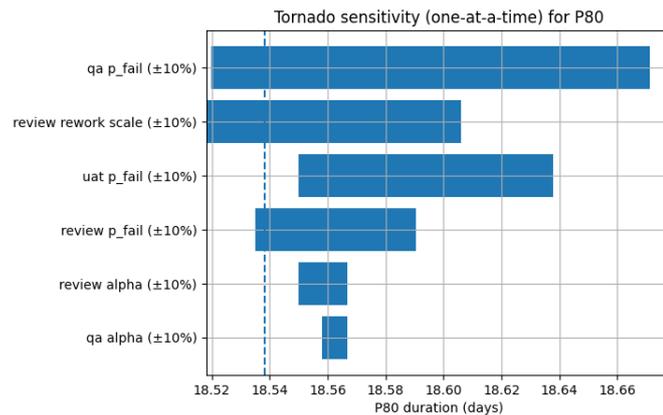


Рисунок 4 – Tornado-діаграма чутливості (one-at-a-time) перцентиля  $P_{80}$  тривалості завершення проекту  $T_{proj}$  до варіації ключових параметрів моделі переробок ( $\pm 10\%$ ); штрихова лінія відповідає базовому значенню  $P_{80}$

5) Criticality Index по роботах (рисунок 5). Стовпчикова діаграма  $CI_i$  показує, які роботи формують стійке критичне ядро, а які лише зрідка стають критичними через випадкові коливання тривалостей.

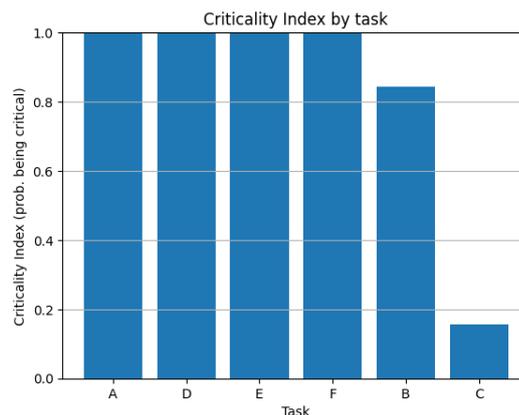


Рисунок 5 – Індекс критичності робіт  $CI_i$  (ймовірність належності роботи  $i$  до критичного шляху) за результатами Монте-Карло моделювання для мережі проекту

### 3.5. Обговорення результатів.

Отримані результати підтверджують, що явне моделювання механізму переробок після контрольних перевірок істотно впливає на оцінювання строків, особливо у верхніх перцентилях розподілу ( $P_{80}/P_{90}$ ). Саме ці показники є найбільш релевантними для управлінських рішень, оскільки характеризують “надійні” строки завершення за наявності ризиків.

Аналіз S-кривих демонструє, що включення переробок зсуває розподіл  $T_{proj}$  вправо та розширює інтервал невизначеності, що безпосередньо підвищує ризик порушення жорстких дедлайнів. Водночас залежність  $P_{deadline}$  від  $p$  може мати ділянки насичення: якщо дедлайн суттєво перевищує типові значення  $P_{50}/P_{80}/P_{90}$ , то ймовірність виконання дедлайну стає близькою до 1 і втрачає чутливість до параметрів переробок. У та-

кому випадку більш інформативними є графіки перцентилів, а також tornado-аналіз, який виявляє відносний внесок різних gate у ризикові строки.

Результати критичності підтверджують, що в стохастичній мережі критичний шлях є змінною величиною: одні роботи залишаються критичними майже завжди, тоді як інші стають критичними лише за певних сценаріїв (наприклад через додаткові переробки на окремих етапах приймання). Це дає підстави для практичних висновків щодо пріоритизації управлінських дій: зниження ймовірностей провалу на найбільш впливових gate та скорочення тривалостей переробок у “критичному ядрі” мережі забезпечують найбільший ефект у зменшенні  $P50 / P80 / P90$ .

У підсумку, імітаційне моделювання в межах запропонованої математичної моделі забезпечує перехід від детермінованих оцінок строків до ймовірнісного опису, що є більш адекватним для реальних ІТ-проектів з багатоетапним контролем якості та повторними переробками. Отримані показники можуть використовуватися для формування буферів, прийняття зобов’язань за контрактними строками, а також для вибору пріоритетних заходів з управління ризиками.

**Висновки.** У роботі сформовано мережево-часову постановку на основі СРМ/PERT для оцінювання строків завершення ІТ-проектів за умов невизначеності, з переходом від “точкових” оцінок до імовірнісних характеристик.

Запропоновано врахування ризиків затримок через керування параметрами задач критичного шляху (зменшення тривалості та дисперсії), що дозволяє формалізувати задачу зниження ризику зриву дедлайну. Реалізовано обчислювальний експеримент (умовний ІТ-проект із 9 задачами) з визначенням критичного шляху та оцінкою ймовірності завершення при заданому дедлайні; показано практичну ефективність підходу. Оптимізаційний вплив на параметри задач забезпечив зростання ймовірності завершення проекту у строк з 15.78% до 43.34%, що демонструє чутливість календарного ризику до навіть помірних змін параметрів мережі. У частині імітаційного аналізу обґрунтовано використання показників ризику та інтерпретаційних візуалізацій: ймовірність виконання дедлайну, індекс критичності робіт (criticality index) і tornado-аналіз чутливості, а також S-криву (ECDF) для перцентильних строків.

Отримані результати підтверджують доцільність поєднання PERT із математичним моделюванням для підвищення надійності планування строків ІТ-проектів і зниження ризику затримок.

#### ЛІТЕРАТУРА / REFERENCES

1. Chen, L., Lu, Q., & Han, D. (2023). A Bayesian-driven Monte Carlo approach for managing construction schedule risks of infrastructures under uncertainty. *Expert Systems with Applications*, 212, 118810. <https://doi.org/10.1016/j.eswa.2022.118810>
2. Fitzsimmons, J. P. (2022). Construction schedule risk analysis. *Electronic Journal of Information Technology in Construction (ITcon)*, 27(5), 100–118. <https://doi.org/10.36680/j.itcon.2022.005>

3. Ichsán, M., Nugraha, D., & Lee, S. (2025). Monte Carlo simulation for enhancing the schedule completion forecast in construction projects. *Applied Sciences*, 15(2), 935–950. <https://doi.org/10.3390/app15020935>
4. Maier, J. F., Eckert, C. M., & Clarkson, P. J. (2017). Simulating progressive iteration, rework and change propagation in complex design processes. *Research in Engineering Design*, 28(1), 25–50. <https://doi.org/10.1007/s00163-016-0232-1>
5. Nelson, R. G., Azaron, A., & Aref, S. (2016). The use of a GERT based method to model concurrent product development processes. *European Journal of Operational Research*, 249(3), 879–890. <https://doi.org/10.1016/j.ejor.2015.08.041>
6. Pregina, K., & Kannan, M. R. (2025). A graphical evaluation and review technique for scheduling construction projects. *Journal of Scheduling*, 28(3), 311–328. <https://doi.org/10.1007/s10951-024-00893-0>
7. Sadri, S., & Fatemi Ghomi, S. M. T. (2024). Evaluating time–cost trade-off in a resource constraint GERT-type project scheduling problem considering rework. *International Journal of Systems Science: Operations & Logistics*. <https://doi.org/10.1080/23302674.2024.2394115>
8. Saiz, M., Juan, A. A., & Martí, R. (2024). A simheuristic for project portfolio optimization combining Monte Carlo simulation and critical path analysis. *Computers & Operations Research*, 159, 106234. <https://doi.org/10.1016/j.cor.2024.106234>
9. Senses, S. (2024). Trade-off between time and cost in project planning under uncertainty using Monte Carlo simulation. *Simulation*, 100(4), 289–303. <https://doi.org/10.1177/00375497241234567>
10. Tao, L., Wu, D., Liu, S., & Lambert, J. H. (2017). Schedule risk analysis for new-product development: The GERT method extended by a characteristic function. *Reliability Engineering & System Safety*, 167, 464–473. <https://doi.org/10.1016/j.res.2017.06.009>

Received 03.11.2025.  
Accepted 27.11.2025.

### ***Stochastic network–time PERT/CPM model with probabilistic rework for forecasting IT project completion times***

*Information technology (IT) projects are often affected by two sources of uncertainty. First, the duration of each activity is not known in advance and can vary widely due to technical complexity, changing requirements, and team-related factors. Second, many deliverables must pass through multiple verification and acceptance gates—such as code review, quality assurance (QA) testing, security checks, and user acceptance testing (UAT). At these gates, outcomes may fail to meet agreed criteria and therefore require rework. This iterative “do–check–rework” cycle is a common reason why projects deviate from deterministic schedules. Traditional CPM schedules and many classical PERT applications typically treat activities as one-pass tasks and do not explicitly represent repeated rework loops. As a result, they can underestimate completion-time risk, especially in the upper tail of the distribution, and provide overly optimistic target dates.*

*This paper presents a stochastic network–time scheduling model that integrates the PERT/CPM framework with an explicit probabilistic representation of rework. The project is modeled as a precedence network where nodes represent activities and directed links represent finish-to-start constraints. Each activity has a base duration described by a three-point*

*estimate (optimistic, most likely, and pessimistic) and is treated as a random variable following a PERT-type distribution. To represent rework, each activity may be associated with one or more verification gates. For each gate, the model specifies a probability that the activity fails the gate and must be reworked, and a probability distribution for the rework effort (also described using three-point estimates). The number of rework cycles is modeled as a random count driven by repeated attempts until the gate is passed. Optionally, the model includes a learning effect: repeated rework cycles may become faster over time as the team gains understanding, defects become localized, and corrective actions become more targeted.*

*To obtain project completion-time forecasts, the study uses Monte Carlo simulation. In each simulation run, base durations and rework outcomes are sampled for all activities and gates, and then a standard CPM forward pass computes start and finish times consistent with network dependencies. Repeating this process many times produces an empirical distribution of total project duration. From this distribution, the paper reports percentile completion dates (such as median, 80th percentile, and 90th percentile) and the probability of meeting a specified deadline. Percentile-based dates are emphasized because they are more suitable for reliable commitments than mean-based estimates in the presence of rework-driven tail risk.*

*In addition to overall completion-time forecasts, the paper provides diagnostic analyses to support risk management. First, an S-curve (empirical cumulative distribution) visualizes the completion-time distribution and allows direct reading of deadline probabilities and percentile dates. Second, parameter sweeps demonstrate how deadline success probability changes when rework probability increases at a specific gate and how upper-percentile completion times respond to changes in the learning effect. Third, a tornado-style one-at-a-time sensitivity analysis ranks which rework-related parameters have the greatest impact on a risk-oriented completion metric (e.g., the 80th percentile). Finally, the paper computes a Criticality Index for each activity—the fraction of simulation runs in which the activity lies on the critical path—showing that criticality can shift between branches under uncertainty and highlighting a “critical core” of activities that most frequently drive the project finish date.*

*The results indicate that explicitly modeling probabilistic rework materially affects forecasted completion dates and expands the uncertainty range, particularly in the upper tail. The approach supports more realistic buffer sizing and deadline-risk assessment than deterministic planning. Sensitivity and criticality outputs provide actionable guidance by identifying which gates (e.g., QA or UAT) and which activities should be prioritized for quality improvements, automation, clearer acceptance criteria, or process changes that reduce the frequency and cost of rework.*

*Overall, the proposed stochastic PERT/CPM model with probabilistic rework offers a practical and mathematically grounded method for forecasting completion dates and quantifying deadline risk in IT projects with multi-stage verification and iterative rework. Future research may extend the model to incorporate resource constraints, correlations between activities and risks, and data-driven estimation of model parameters from project repositories and quality metrics.*

*Keywords: project scheduling; PERT; CPM; Monte Carlo simulation; schedule risk analysis; probabilistic rework; verification gates; learning effect; completion forecasting; critical path variability.*

**Селівьорова Тетяна Віталіївна** – кандидат технічних наук, доцент, доцент кафедри інформаційних технологій і систем, Український державний університет науки і технологій, ORCID: <https://orcid.org/0000-0002-2470-6986>.

**Зеленцов Дмитро Гегемонович** – доктор технічних наук, професор, завідувач кафедри Інформаційних систем, Український державний університет науки і технологій, ORCID: <https://orcid.org/0000-0002-5785-9858>.

**Романько Ярослав Вікторович** – кандидат технічних наук, доцент, доцент кафедри екології, теплотехніки та охорони праці, Український державний університет науки і технологій, ORCID: <https://orcid.org/0000-0001-6333-6888>.

**Тесленко Іван Романович** – студент, Український державний університет науки і технологій, ORCID: <https://orcid.org/0009-0001-0234-868X>.

**Selivorstova Tatyana** – candidate of technical sciences, associate professor, associate professor department of Information Technologies and Systems, Ukrainian State University of Science and Technologies, ORCID: <https://orcid.org/0000-0002-2470-6986>.

**Zelentsov Dmytro** – doctor of technical sciences, professor, head of the department of information systems, Ukrainian State University of Science and Technology, ORCID: <https://orcid.org/0000-0002-5785-9858>

**Romanko Yaroslav** – candidate of technical sciences, associate professor, associate professor of the department of ecology, thermal engineering and occupational safety, Ukrainian State University of Science and Technology, ORCID: <https://orcid.org/0000-0001-6333-6888>.

**Teslenko Ivan** – student, Ukrainian State University of Science and Technology, ORCID: <https://orcid.org/0009-0001-0234-868X>.