

DOI: <https://doi.org/10.36910/6775-2524-0560-2026-64-41>

УДК 004.41:658

Решетняк Олег Анатолійович, к.т.н., ст. викл.

<https://orcid.org/0009-0001-4879-8686>

Перекрыстов Ігор Сергійович, к.т.н., ст. викл.

<https://orcid.org/0009-0007-3805-8143>

Кочеткова Марина Вікторівна, викл.

<https://orcid.org/0009-0006-3189-4002>

Щерба Сергій Юрійович, викл.

<https://orcid.org/0009-0004-2457-9122>

Державний університет інтелектуальних технологій і зв'язку, м. Одеса, Україна

ДОСЛІДЖЕННЯ ВПЛИВУ ГЛИБИНИ ДЕКОМПОЗИЦІЇ НА ХАРАКТЕРИСТИКИ ОЦІНКИ ПРОЕКТУ: ПРАКТИЧНА МОДЕЛЬ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ

Решетняк О. А., Перекрыстов І. С., Кочеткова М. В., Щерба С. Ю. Дослідження впливу глибини декомпозиції на характеристики оцінки проекту: практична модель оцінювання трудомісткості. Статтю присвячено вирішенню проблеми хронічної недооцінки проектів розробки програмного забезпечення. Авторами проаналізовано причини виникнення систематичних помилок при оцінюванні, зокрема когнітивні упередження, ефект оптимізму та зовнішній тиск замовників. Обґрунтовано необхідність об'єктивізації оцінки шляхом деталізації робіт та мінімізації впливу людського фактора. У роботі представлено комплексну методологію оцінювання, яка базується на визначенні оптимальної глибини декомпозиції та застосуванні статистичних методів (PERT-аналіз) для прогнозування трудомісткості. Використано апарат математичної статистики та центральну граничну теорему, що дозволило формалізувати межі довірчих інтервалів для оцінювання проекту в умовах невизначеності. Запропоновано модель із формалізованими розрахунками «прихованих» робіт, таких як тестування, інтеграція та керування ризиками, що дозволяє виявляти реальні трудовитрати. Обґрунтовано, що ігнорування цих нефункціональних, проте критично необхідних етапів є основною причиною відхилень від графіків. Це підкреслює, що якісна декомпозиція це не лише інструмент планування, а й засіб контролю за цілісністю проекту. Практична цінність дослідження полягає у можливості трансформувати суб'єктивні судження в об'єктивні зобов'язання, що мінімізує ризики бюджетних порушень та зриву термінів. Визначено межі відмови від реалізації проекту у разі ігнорування окремих етапів. Розроблена методика сприяє прийняттю реалістичних рішень на етапі планування та підвищує достовірність оцінювання проектів у сферах з високою невизначеністю вимог, а також створює фундамент для прозорості комунікації між розробниками та замовниками. Запропонований підхід є універсальним інструментом, що легко адаптується до різних методологій розробки, суттєво підвищуючи загальну зрілість проектного менеджменту в IT-компаніях.

Ключові слова: оцінка проектів, декомпозиція робіт, WBS, трудомісткість, програмна інженерія, управління ризиками, статистичне моделювання, PERT, достовірність оцінки.

Reshetnyak O., Perekrstov I., Kochetkova M., Shcherba S. Research of the influence of decomposition depth on project estimation characteristics: a practical effort estimation model. The article is devoted to solving the problem of chronic underestimation of software development projects. The authors analyzed the causes of systematic errors in the estimation, in particular cognitive biases, the optimism effect and external pressure from customers. The need to objectify the estimation by detailing the work and minimizing the influence of the human factor is substantiated. The paper presents a comprehensive estimation methodology based on determining the optimal depth of decomposition and applying statistical methods (PERT analysis) to predict effort. The application of mathematical statistics and the Central Limit Theorem has enabled the formalization of confidence interval boundaries for project estimation under conditions of uncertainty. A model with formalized calculations of «hidden» work, such as testing, integration and risk management, is proposed, which allows identifying real efforts. It is argued that ignoring these non-functional yet critically necessary stages is a primary cause of schedule deviations. This underscores that high-quality decomposition is not merely a planning tool, but also a means of ensuring project integrity. The practical value of the study lies in the ability to transform subjective judgments into objective obligations, which minimizes the risks of budget violations and missed deadlines. The limits of refusal to implement the project in case of ignoring individual stages are determined. The developed methodology contributes to making realistic decisions at the planning stage and increases the reliability of project evaluation in areas with high uncertainty of requirements thereby creating a foundation for transparent communication between developers and clients. The proposed approach serves as a universal tool that is easily adaptable to different development methodologies, significantly increasing the overall maturity of project management in IT companies.

Keywords: project estimation, work decomposition, WBS, labor intensity, software engineering, risk management, statistical modeling, PERT, estimation reliability.

Постановка проблеми. Хронічна недооцінка проектів у сфері розробки програмного забезпечення (ПЗ) є системною проблемою індустрії, що регулярно призводить до порушення бюджетів і строків [1, 2]. Першопричини недооцінки полягають у когнітивних упередженнях людини таких як «погане відчуття» часу та «оптимізм» стосовно обсягу робіт [3, 4]. Це призводить до зниження оцінки обсягів робіт на 20-30% та більше [5]. Інша причина, зовнішній тиск: клієнти

розглядають оцінку як критерій вибору, спонукаючи інженера до її «мінімізації» [6]. Тиск часу змушує проводити оцінювання «на ходу» та забувати, що оцінка проекту є також проектом із своїм циклом та правилами. Порушення циклу, наприклад, ігнорування фази виявлення вимог, веде до суттєвої недооцінки та катастрофи проекту у якнайменш 20% випадків [2]. Головна причина таких катастроф – ігнорування прихованої трудомісткості, робіт «під капотом», які не мають прямої функціональності для замовника. До прихованих робіт також належать так звані «інтелектуальні зусилля» на аналіз, дизайн та тестування ідей, які інженери часто змушені виконувати без окремої оцінки. Отже недостатня декомпозиція робіт є найбільшою проблемою, оскільки оцінювання є процесом виявлення конкретних задач, а не просто підрахунку відомих зусиль [7]. Ризики підсилюються тим фактом, що оцінювання здійснюється на ранній стадії невизначеності, коли про проект відомо занадто мало. Проте подолання проблеми вимагає об'єктивізації оцінки через деталізацію, що підвищує достовірність завдяки закону великих чисел та усуненню ефекту сегментації.

Аналіз досліджень. Декомпозиція робіт це процес розбиття загального обсягу робіт проекту на менші, керовані, атомарні задачі або робочі пакети. Декомпозиція може бути як недостатньою (що веде, як правило, до недооцінки проекту), так і надмірною (що веде до переоцінки проекту). На Рис. 1 представлена відома якісна залежність зростання ресурсів необхідних для реалізації проекту від похибки оцінки [5]. Аналіз залежності демонструє, що саме наслідки недооцінки проекту є нелінійно-катастрофічними, тому що відсутність ресурсів для реалізації призводить до необхідності переробляти проект, знову і знову. Переоцінка проекту призводить до додаткових витрат ресурсів або за рахунок простоїв (так званий закон Паркінсона), або, у кращому сценарії, за рахунок підвищення якості та додавання нової функціональності.



Рис. 1. Залежність ресурсів проекту від похибки оцінки

Недостатня декомпозиція призводить до недооцінки по причині когнітивних спотворень, які виникають у оцінщика. По-перше, люди погано оцінюють великі задачі і як правило оцінюють їх оптимістично. Це відомий ефект project fallacy [8]. По-друге, і більш важливе, у разі недостатньої декомпозиції є великі ризики упустити види робіт, які необхідно включити в проект, наприклад, не включити в оцінку роботи пов'язані з фазою приймального тестування. Це відомий ефект сегментування, який виражається у тому що оцінка проекту у цілому є меншою за суму оцінок окремих задач [5]. Цей ефект має і іншу сторону, коли замовники не розуміють необхідності оплати робіт «під капотом» і вимагають їх виключати із оцінки.

Метою дослідження є представлення комплексної методології оцінювання проектів розробки ПЗ, апробованої на практиці. Ця методологія дозволяє розрахувати необхідну глибину декомпозиції проекту та статистичні характеристики оцінки, для формалізації достовірності оцінки. Також запропонована модель із формалізованими розрахунками різних видів робіт проекту з розробки ПЗ. Результати досліджень дозволяють трансформувати суб'єктивні оціночні судження різних сторін процесу оцінювання в об'єктивну оцінку та реалістичні зобов'язання.

Основна частина. Отже, головна задача оцінщика детально декомпонувати роботи проекту, після чого можна оцінювати трудовитрати на кожну окрему атомарну задачу (інші задачі оцінщика, як от аналіз ризиків, не є предметом даного дослідження). Якщо проект декомпонувати на N задач, то можна розрахувати коефіцієнт, який називають segmentation factor S :

$$S = \frac{\sum_{i=1}^N E_i}{N} = \frac{E}{N}, \quad (1)$$

де E_i - оцінка кожної атомарної задачі, у людино-годинах;

E - сумарна оцінка проекту, у людино-годинах.

Існує практично доведена межа цього коефіцієнту [5]:

$$\lim_{N \rightarrow \infty} S \rightarrow 4. \quad (2)$$

Це математичне обмеження розкриває інший бік ефекту сегментації: так само як і великі обсяги робіт, занадто дрібні задачі оцінюються людиною неточно. Тому практичною межею деталізації WBS є 4 людино-години; подальша декомпозиція не підвищує достовірність, а лише створює ризик штучної переоцінки проекту через надмірну атомізацію робіт [9].

Якщо кожну окрему задачу розглядати як випадкову величину, то її дуже точно можна описати бета-розподілом [5], параметри якого α та β дуже просто пов'язати з параметрами добре відомої техніки аналізу та оцінки проектів PERT [10]:

$$\alpha = \frac{4M + P - 5O}{P - O}, \quad (3)$$

$$\beta = \frac{5P - O - 4M}{P - O},$$

де O - оптимістична оцінка задачі;

M - ймовірна оцінка задачі;

P - песимістична оцінка задачі.

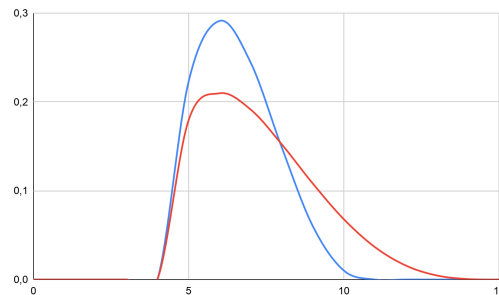


Рис. 2. Щільність ймовірності бета-розподілу задачі з параметрами:
 $O = 4, M = 6, P = 11$ та $O = 4, M = 6, P = 15$

Математичне очікування трудовитрат для однієї задачі (μ_i) за методом PERT розраховується за формулою:

$$\mu_i = \frac{O + 4M + P}{6}. \quad (4)$$

Введемо два припущення:

- по-перше, зважаючи на ризики, песимістична оцінка в середньому перевищує оптимістичну у 5 разів ($P = 5O$);
- по-друге, спираючись на визначену раніше границю (2), ймовірна оцінка ідеально декомповованої задачі становить $M = 4$ години.

Підставивши ці припущення у базову формулу, отримаємо математичне очікування для однієї задачі:

$$\mu_i = \frac{0+16+50}{6} = 0 + \frac{8}{3}. \quad (5)$$

Якщо застосувати Центральну граничну теорему, припустивши статистичну незалежність N задач (що коректно для цілей дослідження), то загальні трудовитрати проекту стають випадковою величиною з нормальним законом розподілу, де математичне очікування всього проекту μ прямує до суми очікувань атомарних задач:

$$\mu \rightarrow \left(0 + \frac{8}{3}\right) N; \quad (6)$$

Відповідно, дисперсія трудовитрат проекту прямує до:

$$\sigma^2 \rightarrow N \frac{40^2}{9}. \quad (7)$$

Статистичні характеристики оцінки трудовитрат проекту можуть дозволити аналізувати достовірність цієї оцінки, тобто виявляти діапазон у межах якого проект може бути виконаний з заданою ймовірністю. Для нормального закону розподілу стандартним діапазоном є $\pm 3\sigma$, який охоплює ймовірність 0,997.

Існує декілька підходів до декомпозиції робіт [11] та одним з найпоширеніших є побудова ієрархічної структури робіт (Work Breakdown Structure, WBS). Оскільки проект у цілому завжди більше ніж сума окремих його частин, WBS і дозволяє як виявляти окремі задачі проекту, так і дивитися на проект у цілому (наприклад, будувати критичний шлях).

Комплексна модель оцінювання трудовитрат вимагає введення наступних базових компонентів для охоплення всієї трудомісткості проекту (див. Табл. 1).

Таблиця 1. Базові компоненти WBS проекту

Вид роботи	Деталізація декомпозиції	Правила розрахунків
Development (Розробка)	Робота розділяється на: - Базові функціональні задачі: логіка, API, екрани, UI, тощо. - Нефункціональні задачі: аналітика, логування, безпека, моніторинг, міграція, резервне копіювання, адаптація, тощо.	Рекомендації щодо атомарності: - Мінімальна задача - оцінка 4 людино-години. - Обов'язкова декомпозиція задач трудовитрату яких перевищують 80 годин.
Debugging & Bug Fixing (Виправлення помилок)	Робота розділяється на: - Рефакторинг та оптимізація коду. - Усунення технічного боргу. - Безпосереднє виправлення помилок після тестування.	Рекомендації щодо розрахунку: - З практичного досвіду впливає, що 20% від трудовитрат на розробку є базовою оцінкою. - Якщо ПЗ підтримується на декількох платформах, то є сенс додати до 5% на налагодження на всіх платформах.
QA (Забезпечення якості)	Робота розділяється на: - Ручне тестування та звітність. - Автоматизація тестування (тестування API, Unit-тести, інтеграційні тести, тощо). - Спеціалізоване тестування, яке	Рекомендації щодо розрахунку: - На ручне тестування застосовується до 35% від трудовитрат на розробку.

	потребує створення пісочниці (платежі, аудіо/відео стрімінг, тощо).	- Автоматизація, якщо застосовується, буде вимагати якнайменш 25% від розробки, з потенційною можливістю зменшити трудовитрати на ручне тестування. - На спеціалізоване тестування треба виділяти від 5% розробки.
Інтеграція та Delivery Management (Керування поставкою)	Ця робота супроводжує попередні види робіт та розділяється на: - Інфраструктурні задачі: розгортання та підтримка інфраструктури, деплоймент, автоматизація (CI/CD), тощо. - Організаційні задачі: планування, комунікації, звітність, тощо. - Управлінські задачі: моніторинг та контроль (строків, бюджету, якості продукту, якості коду, якості архітектури), управління вимогами, управління ризиками, тощо.	Рекомендації щодо розрахунку: - 20% від загальних трудовитрат на Розробку, Виправлення помилок та Забезпечення якості.
Development Risks (Ризики)	Робота, задіює всіх причетних до розробки проекту, розділяється на: - Створення та підтримання реєстру ризиків. - Втілення планів усунення ризиків. - Втілення планів послаблення ризиків.	Рекомендації щодо розрахунку: - Якнайменш буфер 15% від розробки треба закладати на роботу з ризиками.
Acceptance (Приймання проекту)	Робота розділяється на: - Інфраструктурні задачі підготовки середовища для тестування. - Управлінські задачі: контроль строків, контроль вимог, тощо. - Виправлення помилок.	Трудовитрати на прийомку краще обговорювати на старті проекту і окремо прописувати у договорі.

Отже, об'єктивізація оцінки трудовитрат і проекту в цілому досягається за рахунок кількісних розрахунків «прихованої» трудомісткості, як от нефункціональних вимог, які як фундамент будинку, не надають прямих функцій до проекту, але є необхідними для стійкості та масштабування кінцевої системи. Так, виправлення помилок – це нормальний процес, а очікування, що «розробники повинні писати код без помилок», є необґрунтованим. Ігнорування цих вимог та інших видів робіт, що супроводжують основну розробку, призводить до ризиків повного переписування коду частин проекту, що дорожче, ніж початкове включення цих робіт.

Випадок недостатньої декомпозиції Розробки: дуже часто спостерігається ситуація, коли поверхневі оцінки (ballpark або rough) приймаються за достовірні та такі оцінки використовуються при узгодженні умов розробки проекту.

Нехай: є три проекти; прогнозується, що розробка (Development) кожного проекту дорівнює 480 людино-годин; оцінка першого проекту зроблена з глибиною декомпозиції $N = 12$ задач, другого проекту з $N = 120$ і третього $N = 240$. Якщо скористатися формулами (6) та (7), можна отримати графіки представлені на Рис. 3.

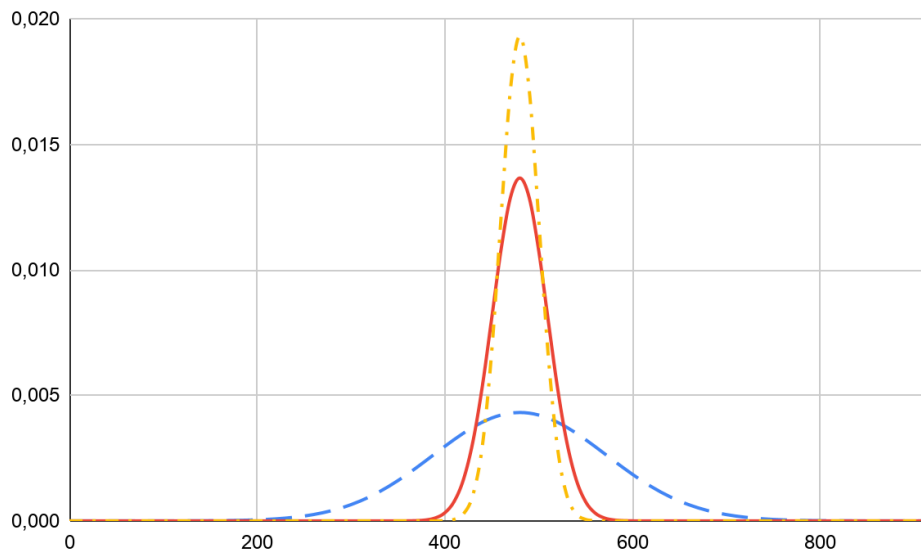


Рис. 3. Щільність ймовірності трудовитрат проекту у цілому

- 1) $N = 12$, штрихова лінія
- 2) $N = 120$, суцільна лінія
- 3) $N = 240$, штрих-пунктирна лінія

Перша оцінка є недостовірною, оскільки діапазон оцінки проекту з ймовірністю 0,997 дорівнює 202-752 людино-години, що 115% від прогнозу оцінки розробки. Цей відсоток і є характеристикою ризиків того, що проект не буде реалізований. Отже поверхневі оцінки не можуть використовуватися для взяття зобов'язань про реалізацію проекту.

Якщо порівняти другий та третій проекти, то подвоєння глибини декомпозиції призводить до підвищення достовірності оцінки лише на 10%, але супроводжується подвоєнням трудовитрат на виконання декомпозиції, значно ускладнює аналіз декомпозиції та значно підвищує ризики переоцінки проекту.

Випадок ігнорування робіт: розглянемо другий проект з попереднього випадку, де базовим параметром чистої розробки (математичним очікуванням) є 480 людино-годин. З огляду на глибину декомпозиції $N = 120$, розрахований для нормального розподілу діапазон з ймовірністю 0,997 становить від 392 до 568 людино-годин. У Табл. 2 наведено приклад розрахунку загальних трудовитрат цього проекту, сформований на основі меж даного довірчого інтервалу та запропонованих правил WBS.

Таблиця 2. Розрахунок загальних трудовитрат проекту

Вид роботи	Нижня границя оцінки трудоуитрат, людино- години	Верхня границя оцінки трудоуитрат, людино- години
Development (Розробка)	392	568
Debugging & Bug Fixing (Виправлення помилок)	79	114
QA (Забезпечення якості)	138	199

Інтеграція та Delivery Management (Керування поставкою)	122	177
Development Risks (Ризики)	59	86
Acceptance (Прийомка проекту)	40	80
Загальні трудовитрати	830	1224
Виключена Прийомка	790	1144
Додатково виключено буфер на Ризики	731	1058
Додатково виключено Керування поставкою	609	881
Додатково виключене Виправлення помилок	530	767

Висновки та перспективи подальших досліджень. Перший висновок, який є найбільш красномовним та спірним, загальні трудовитрати проекту приблизно вдвічі перевищують трудовитрати на розробку (якщо спростити, на написання коду).

Другий висновок, заради оптимізації оцінки або на вимогу замовника про зниження оцінки деякі види робіт можна виключати і мати шанси на реалізацію проекту. Отже, якщо вилучити Прийомку з оцінки (замовник відмовляється сплачувати за цей етап), новий діапазон буде перетинатися з Загальними трудовитратами на 80%. Якщо вилучити буфер на Ризики (замовник стверджує, що всі дані надані та ризики не спрацюють), перетин буде дорівнювати 58%; при вилученні Керування поставкою (розробники це висококваліфіковані спеціалісти якими не треба керувати) – 13%.

Висновок третій, є межа вилучення (або ігнорування) видів робіт, при перетині якій реалізація проекту стає неможливою у заданих умовах. Якщо замовник очікує, що розробники повинні писати код без помилок та вимагає додатково виключити пункт Виправлення помилок, отриманий діапазон не перетинається з оцінкою загальних трудовитрат і реалізувати такий проект буде неможливо. Тому від реалізації проекту треба відмовлятися, або змінювати умови.

Висновки та перспективи подальших досліджень

Отримання достовірних і точних оцінок проектів є надскладною задачею, складність якої нелінійно зростає зі складністю проекту. Важливо відзначити, що одне з рішень, яке дуже широко поширене у проектах розробки ПЗ – не оцінювати проект в цілому, а оцінювати лише обсяг робіт на найближчий інтервал (спринт) [11]. Інше рішення, якого достатньо у більшості випадків, зосередитися на достовірності оцінки, для чого розраховуються діапазони оцінок та додаткові (статистичні) характеристики оцінок. У статті представлена опробована практикою методологія оцінювання трудомісткості проектів розробки ПЗ, яка забезпечує достатню для реалізації достовірність.

Перший та надважливий висновок, полягає у тому, що використання методології, що ґрунтується на формалізованих математичних методах є єдиним способом трансформувати суб'єктивне «оціночне судження» замовника або оцінювача у фактично обґрунтовану оцінку. Якісна оцінка – це та, що забезпечує достатньо ясне уявлення про реальний стан трудовитрат проекту і дозволяє приймати рішення, що ведуть до поставки проекту.

Друге, оцінка в широкому сенсі є процесом виявлення конкретних задач (управління змістом), а не лише виявленням трудозатрат на вже відомі задачі. Недостатня декомпозиція робіт є найбільшою проблемою, оскільки призводить до недооцінки через ризик упустити необхідні види робіт:

- Глибина декомпозиції прямо впливає на достовірність оцінки завдяки Закону великих чисел: тобто помилки в оцінках дрібних підзадач частково компенсують одна одну. Проте, існує межа деталізації: не рекомендовано декомпонувати завдання менше ніж на 4 людино-години, щоб уникнути ризиків переоцінки. Завдання, які перевищують 80 годин, підлягають обов'язковій декомпозиції. Поверхневі оцінки є недостовірними, оскільки діапазон оцінки

проекту може становити вище 100% від прогнозованої оцінки розробки, що є неприйнятним для взяття зобов'язань.

- Об'єктивізація оцінки вимагає обов'язкового включення та кількісного розрахунку «прихованої» трудомісткості, яка супроводжує основну розробку. Такі роботи можуть збільшувати загальні трудовитрати проекту вдвічі та більше порівняно з початковою оцінкою лише безпосередньої розробки. Ігнорування таких робіт, може призвести до ситуації, коли реалізація проекту стає неможливою у заданих умовах.

Подальші дослідження проблеми оцінки трудовитрат проекту можуть бути зосереджені на двох напрямках: теоретично-управлінському та психологічно-поведінковому. Це включає аналіз конфлікту між оцінкою, ціллю та зобов'язанням, а також дослідження впливу когнітивних упереджень та адміністративного тиску на кінцеву реалістичність оцінки.

Список бібліографічного опису

1. Флів'єрґ Б., Бадзьєр А. Чому ваш IT-проект може бути ризикованішим, ніж ви думаєте. Harvard Business Review. 2011. Т. 89, № 9. С. 23–25. DOI:10.2139/ssrn.2229735
2. Арчадіаконо Г. Порівняльне дослідження рівня невдач IT-проектів: поздовжнє оновлення 2025 року. PM World Journal. 2026. Т. XV, Вип. I. URL: <https://pmworldjournal.com/wp-content/uploads/2026/01/pmwj160-Jan2026-Arcidiacono-research-on-IT-project-failure-rate-2025-update.pdf>
3. Канеман Д., Тверські А. Інтуїтивне прогнозування: упередження та коригувальні процедури. TIMS Studies in Management Science. 1979. Т. 12. С. 313–327.
4. Шарот Т. Упередження оптимізму. Current Biology. 2011. Т. 21, Вип. 23. С. R941–R945. DOI: 10.1016/j.cub.2011.10.030
5. МакКоннелл С. Оцінювання програмного забезпечення: демістифікація «чорного мистецтва». Microsoft Press, 2006. 352 с.
6. Флів'єрґ Б. Стимування упередження оптимізму та стратегічного викривлення в плануванні: прогнозування на основі класів порівняння на практиці. European Planning Studies. 2008. Т. 16, № 1. DOI:10.1080/09654310701747936
7. Брукс Ф. Міфічний людино-місяць: есе про програмну інженерію. Addison-Wesley, 1975. 195 с.
8. Фрід Дж., Ганссон Г. Д. Переробка. Crown Currency, 2010. 288 с.
9. Йоргенсен М. Експертне оцінювання трудомісткості розробки програмного забезпечення «зверху-вниз» та «знизу-вгору». Information and Software Technology. 2004. Т. 46, Вип. 1. С. 3–16. DOI: 10.1016/S0950-5849(03)00093-4
10. Керзнер Г. Управління проектами: системний підхід до планування, графіків та контролю. Wiley, 2005. 1040 с.
11. Кон М. Agile-оцінювання та планування. Pearson, 2005. 368 с.

References

1. Flyvbjerg B., Budzier A. Why Your IT Project May Be Riskier Than You Think. Harvard Business Review. 2011. Vol. 89, no. 9. P. 23–25. DOI:10.2139/ssrn.2229735
2. Arcidiacono G. Comparative research on IT project failure rates: A 2025 longitudinal update. PM World Journal. 2026. Vol. XV, Issue I. URL: <https://pmworldjournal.com/wp-content/uploads/2026/01/pmwj160-Jan2026-Arcidiacono-research-on-IT-project-failure-rate-2025-update.pdf>
3. Kahneman D., Tversky A. Intuitive Prediction: Biases and Corrective Procedures. TIMS Studies in Management Science. 1979. Vol. 12. P. 313–327.
4. Sharot T. The optimism bias. Current Biology. 2011. Vol. 21, Issue 23, P. R941–R945. DOI: 10.1016/j.cub.2011.10.030
5. McConnell S. Software Estimation: Demystifying the Black Art (Developer Best Practices). Microsoft Press, 2006. 352 p.
6. Flyvbjerg B. Curbing Optimism Bias and Strategic Misrepresentation in Planning: Reference Class Forecasting in Practice. European Planning Studies. 2008. Vol. 16, No. 1. DOI:10.1080/09654310701747936
7. Brooks F. The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley, 1975. 195 p.
8. Fried J., Hansson H. D. Rework. Crown Currency, 2010. 288 p.
9. Jørgensen M. Top-down and bottom-up expert estimation of software development effort. Information and Software Technology. 2004. Vol. 46, Issue 1, P. 3–16. [https://doi.org/10.1016/S0950-5849\(03\)00093-4](https://doi.org/10.1016/S0950-5849(03)00093-4)
10. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Wiley, 2005. 1040 p.
11. Cohn M. Agile Estimating and Planning (Robert C. Martin Series). Pearson, 2005. 368 p.

Історія статті:

Отримано: 18.06.2026 Доопрацьовано: 22.09.2026 Прийнято до друку: 23.09.2026 Опубліковано: 29.09.2026