УДК 004.94+519.2
DOI https://doi.org/10.32782/2663-5941/2024.3.1/23

Москалик Д.О.
Державний університет «Житомирська політехніка»

Антонюк Д.С.
Державний університет «Житомирська політехніка»

ІМОВІРНІСНИЙ РОЗПОДІЛ СКЛАДНОСТІ ВХІДНИХ ЗАДАЧ ДЛЯ ДИСКРЕТНО-ПОДІЙНОГО ТА АГЕНТНО-ОРІЄНТОВАНОГО МОДЕЛЮВАННЯ ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У даній роботі досліджується імовірнісний розподіл складності задач розробки програмного забезпечення шляхом аналізу набору історичних даних розробки програмних проєктів з відкритим вихідним кодом, що входять до фонду "Apache Software Foundation". На основі гістограми імовірнісного розподілу витраченого часу на задачі по виправленню дефектів, покращенню існуючих функцій та створення нових, перевірена та спростована гіпотеза відповідності фактичного розподілу імовірностей формі логнормального розподілу. Натомість запропоновано та доведено доцільність застосування зворотного розподілу Гауса, що точніше повторює реальний розподіл імовірностей, а також розраховано його числові параметри. Для практичного застосування при низькорівневому моделюванні процесу розробки програмного забезпечення наведено алгоритм генерації значень випадкової величини за даним розподілом. При порівнянні функції густини імовірності зворотного розподілу Гауса з гістограмою фактичного розподілу витраченого часу виявлено аномальне розходження графіків в точці найпоширенішого значення рівного 20 хвилинам. Для компенсації даної аномалії розроблено адаптацію алгоритму імовірнісної генерації складності задач розробки програмного забезпечення, що зберігає оригінальну форму зворотного розподілу Гауса та відтворює аномально високу частоту генерації найпоширенішого значення. Додатково, при аналізі історичних даних виявлено, що більше 99 % значень витраченого часу на одну задачу є кратними 10, що дозволяє модифікувати поточний алгоритм для генерації дискретних значень шляхом округлення згенерованих значень вгору до найближчого числа, кратного 10, що виключає можливість генерації нульових значень та дозволить оптимізувати обчислення при використанні даного алгоритму у відповідних симуляціях. Так як зворотний розподіл Гауса не обмежений максимальним значенням, для запобігання спотворенню результатів симуляцій доцільно відкидати всі згенеровані значення більші за певний максимальний поріг, що має визначатись згідно вимог відповідних симуляцій. Таким чином зберігається загальна форма розподілу імовірностей, а сам розподіл обмежений скінченним діапазоном значень.

Ключові слова: *моделювання, симуляції, складність задач, розробка програмного забезпечення, зворотний розподіл Гауса, логнормальний розподіл, алгоритм.*

Постановка проблеми. Вибір архітектурної моделі при проєктуванні майбутньої програмної системи є одним з ключових факторів, що визначають можливість успішного завершення даного проєкту та його подальшого ефективного розвитку і підтримки. В сучасному світі при виборі архітектурного підходу найбільша увага приділяється функціональним вимогам до системи та найбільш поширеним нефункціональним вимогам, таким як швидкодія, надійність, безпека, зручність використання тощо. В той же час, ефективність процесу розробки такого програмного продукту зазвичай не враховується через складність її оцінки та відсутність загальновідомих порівняльних характеристик впливу різних архітектурних підходів на ефективність розробки. Як показало дослідження 5400 IT-проєктів, в середньому великі проєкти перевищують початковий бюджет на 45 %, 7 % затримуються і не встигають завершитись до встановленого кінцевого терміну та містять на 56 % менше функцій, ніж було заплановано від самого початку [1, с. 4].

Проведення масштабного експерименту шляхом розробки однієї і тієї ж програмної системи із застосуванням різних архітектурних підходів для подальшого порівняльного аналізу є недоцільним через його високу вартість та необ'єктивність отриманих результатів вимірювань через різницю

в знаннях та досвіді учасників експерименту, а також неможливості передбачення та виключення будь-яких інших випадкових факторів, що можуть спотворити результат експерименту. Ще одним із можливих шляхів дослідження такого впливу є проведення низькорівневих симуляцій процесів розробки для різних типів архітектури за допомогою дискретно-подійного та агентно-орієнтованого моделювання, що не потребує великих фінансових та часових витрат, а також абсолютно виключає вплив людського фактору на результати вимірювань.

Аналіз останніх досліджень і публікацій. Моделювання різних бізнес-процесів вже давно широко використовується як на практиці, так і в дослідницькій діяльності. Важливість ролі моделювання бізнес-процесів підприємств для їх подальшого вдосконалення та впровадження інновацій підкреслюється в працях Т. Шматковської, Т. Коробчук [2, с. 157], а також А. Сараванос (Saravanos, A.), М. Курінґа (Curinga, M.X.) [3, с. 15]. Аналіз наукових досліджень щодо моделювання процесу розробки програмного забезпечення [4, с. 16] виявив тенденції до спаду застосування парадигми системної динаміки та зростання популярності серед науковців дискретно-подійних та агентно-орієнтованих симуляцій, які також виявились найпоширенішими парадигмами.

Різні підходи до моделювання процесу розробки програмного забезпечення висвітлювались в працях вітчизняних науковців, серед яких Ю.С. Кордунова, М. Фелтіновскі, О.В. Придатко, О.О. Смотр [5, с. 29], С.Б. Приходько, Н.В. Приходько, К.О. Книрік [6, с. 50], а також іноземними дослідниками, такими як А. Сараванос (Saravanos, A.), М. Курінґа (Curinga, M.X.) [3, с. 15]. Однак в даних роботах процес розробки програмного забезпечення моделюється на рівні етапів життєвого циклу програмного проєкту, де рівень деталізації ходу виконання робіт недостатній для дослідження впливу програмної архітектури системи на ефективність його розробки.

Теоретичні аспекти низькорівневого моделювання процесу розробки програмного забезпечення, включно з аналізом законів розподілу розміру вхідних задач, були досліджені різними вітчизняними та іноземними науковцями, серед яких І.В. Ярош, Є.В. Павловський, І.А. Назарова [7, с. 80], Ан Джен Чіанг (An Jen Chiang) і Ангус Джіанг (Angus Jeang) [8, с. 2], П.О. Бошар (Bochard, P.O.) і Т. Шварц (Schwarz, T.) [9, с. 404], Д.О. Москалик і Д.С. Антонюк [10, с. 32], М. Лунесу (Maria Ilaria Lunesu), Р. Тонеллі (Roberto Tonelli), Л. Маркезі (Lodovica Marchesi), М. Маркезі (Michele Marchesi) [11, с. 134–245]. В різних працях використовуються різні закони розподілу складності задач, такі як нормальний, випадковий, бета-розподіл, ступенева та дрібно-раціональна функції та інші. Проте останнє дослідження історичних записів проєктів з відкритим вихідним кодом [10, с. 32] виявило, що складність задач підпорядковується логнормальному розподілу, але дана гіпотеза ще вимагає підтвердження, а також числові параметри такого розподілу поки не були досліджені. Крім того, логнормальний розподіл не має верхньої межі, що вимагає його адаптації для можливості застосування в симуляції процесу розробки.

Постановка завдання. Метою статті є дослідження імовірнісного розподілу складності вхідних задач для дискретно-подійного та агентно-орієнтованого моделювання процесу розробки програмного забезпечення шляхом аналізу фактичного розподілу розміру задач на основі історичних даних проєктів з відкритим вихідним кодом.

Виклад основного матеріалу. Для дослідження імовірнісного розподілу складності задач при розробці програмного забезпечення доцільно проаналізувати той самий набір даних “Apache Jira Issue Tracking Dataset” [12], що використовувався для виявлення самого закону логнормального розподілу [10, с. 32]. Даний набір містить історичну інформацію про виконані задачі при розробці програмних проєктів з відкритим вихідним кодом, що належать до фонду “Apache Software Foundation”. Із загальної кількості у більш ніж мільйон різних типів записів, для аналізу розподілу складності практичних задач були обрані лише повністю завершені задачі з інформацією про фактично витрачений час, які мають тип дефекту (“Bug”), покращення існуючої функції (“Improvement”) або створення нової (“New Feature”). Типи записів більш високого (абстрактного) рівня, задачі по тестуванню та створенню документації та інші були виключені з вибірки, оскільки вони або включають в себе, або вже є включеними в задачі обраних типів.

Для більш детального аналізу закону розподілу складності задач розробки є доцільним дослідити розподіл як всіх задач разом, так і окремо кожного з обраних типів задач. На рисунку 1 зображені відповідні гістограми розподілу часу, витраченого на різні типи задач.

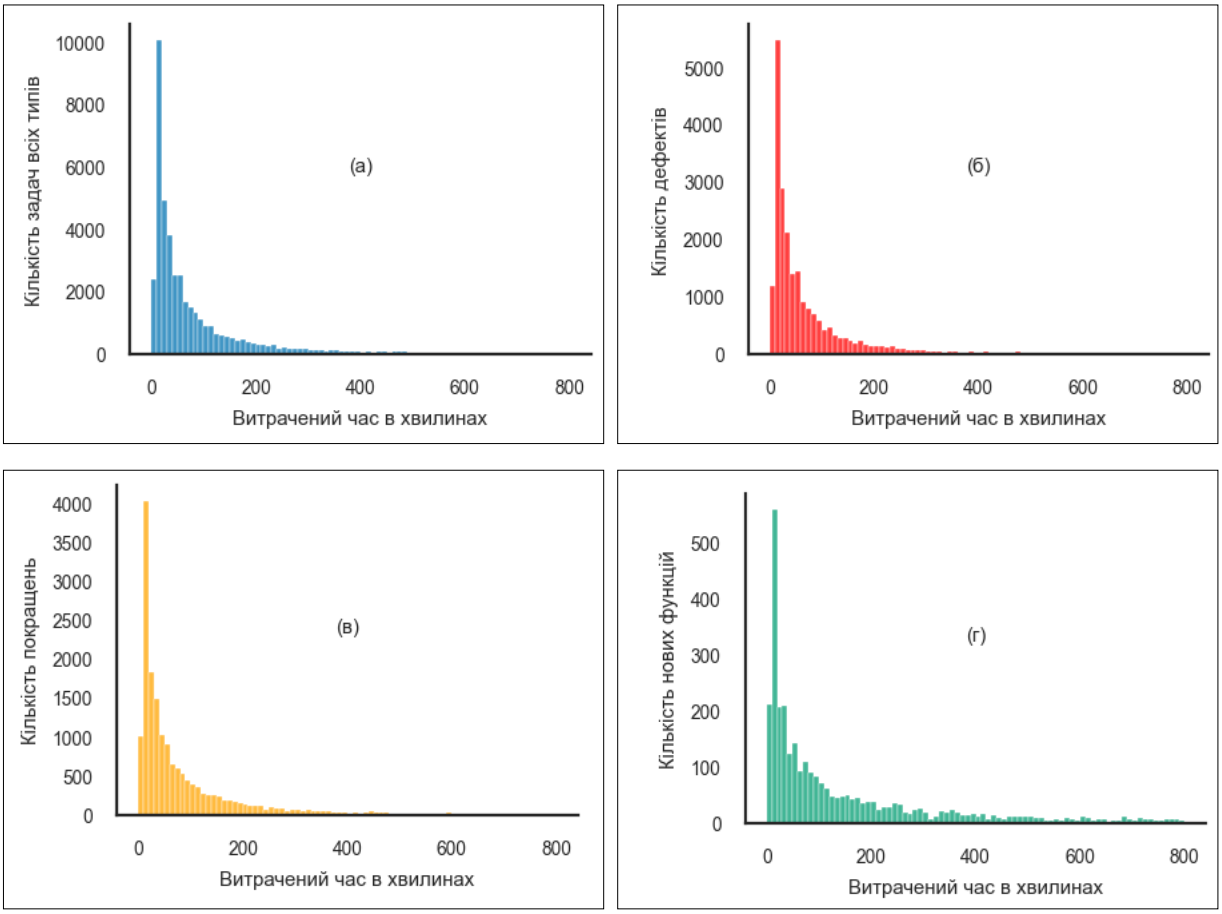

Рис. 1. Гістограми розподілу витраченого часу для (а) всіх типів задач, (б) лише дефектів, (в) покращень та (г) нових функцій

Вищенаведені гістограми мають однакову форму функції розподілу імовірностей незалежно від типу задач, проте для доведення гіпотези саме логнормального розподілу необхідно визначити числові параметри даного розподілу та порівняти графік отриманої функції густини імовірності з фактичними історичними даними.

Логнормальний розподіл випадкової величини X задається функцією:

$$X = e^{\mu+\sigma Z}, \qquad (1)$$

де μ та σ – дійсні числа, при чому $\sigma > 0$, а Z – випадкова величина, задана стандартним нормальним розподілом. Параметри розподілу μ та σ можна знайти за допомогою методу максимальної правдоподібності [13, с. 332], використовуючи наступні формули:

$$\hat{\mu} = \frac{\sum_k \ln x_k}{n}, \quad \hat{\sigma} = \sqrt{\frac{\sum_k \left(\ln x_k - \hat{\mu}\right)^2}{n}}, \qquad (2)$$

де x_k – значення витраченого часу на k-ту задачу, n – загальна кількість задач.

Після визначення параметрів розподілу можливий розрахунок таких його числових характеристик, як середнє значення витраченого часу, медіана, мода та дисперсія. Результати розрахунків з точністю до 4 десяткових знаків наведені в таблиці 1.

Виходячи з гістограм на рисунку 1 та обчислених характеристик розподілу з таблиці 1 можна зробити висновок, що розподіли складності дефектів та покращень наближені до загального розподілу всіх типів задач разом, в той час як для задач на розробку нових функцій програмної системи спостерігається більше середнє значення витраченого часу та підвищена імовірність його відхилення в сторону збільшення. З точки зору практичного процесу розробки програмного забезпечення, специфіка роботи над різними типами задач не має суттєвих відмінностей одна від одної, тому для генерації потоку вхідних задач при моделюванні процесу розробки програмного забезпечення достатньо використати параметри імовірнісного розподілу складності для всіх типів задач.

Таблиця 1

Параметри та характеристики логнормального розподілу складності задач

	Параметри розподілу		Характеристики розподілу			
	μ	**σ**	**Середнє**	**Медіана**	**Мода**	**Дисперсія**
Всі типи задач	3,9687	1,0512	91,9440	52,9136	17,5249	17070,9490
Дефекти	3,8718	0,9600	76,1424	48,0269	19,1073	8774,9768
Покращення	4,0020	1,0685	96,8140	54,7048	17,4662	19983,3344
Нові функції	4,4581	1,3645	218,9741	86,3193	13,4134	260621,6962

Відповідно до формули 1, окрім обрахованих параметрів логнормального розподілу, для генерації задач різної складності необхідно також мати можливість генерувати значення випадкової величини за стандартним нормальним розподілом. Для цього доцільно використати перетворення Бокса-Мюллера, так як даний підхід є більш обчислювально ефективним за метод зворотного перетворення та алгоритм Зіккурата. Полярна форма перетворення Бокса-Мюллера для моделювання стандартних нормально розподілених випадкових величин [14, с. 281] виглядає наступним чином:

$$R = x^2 + y^2,\ z_0 = x \cdot \sqrt{\frac{-2\ln R}{R}},\ z_1 = y \cdot \sqrt{\frac{-2\ln R}{R}}, \quad (3)$$

де x та y – незалежні випадкові величини, рівномірно розподілені на інтервалі $[-1, 1]$, z_0 та z_1 – отримані незалежні випадкові величини, розподілені нормально з математичним сподіванням 0 і дисперсією 1, що задовольняє стандартному нормальному розподілу. Якщо при обрахунку значення R виявиться, що $R > 1$ чи $R = 0$, то значення випадкових величин x та y мають бути згенеровані повторно, поки дана умова щодо R не буде виконана.

На рисунку 2 наведено оригінальну гістограму, побудовану на фактичних історичних даних з вибірки, та гістограму згенерованих випадкових чисел відповідно до логнормального розподілом з обчисленими параметрами.

З вищенаведених графіків видно, що отриманий розподіл імовірностей має більш пологий спуск, ніж фактичний закон розподілу, що спростовує гіпотезу наявності логнормального розподілу складності задач згідно історичних даних. Враховуючи переважаючу імовірність найчастішого значення витраченого часу на задачу і суттєво менші імовірності сусідніх значень, є доцільним застосувати інший закон розподілу імовірностей, що буде краще відображати розподіл фактичних даних. Серед загальновідомих законів розподілу найбільш наближену форму функції густини імовірностей має зворотний розподіл Гауса [15, с. 3], тому варто визначити його параметри і так само порівняти результат з фактичним розподілом складності задач.

Параметри зворотного розподілу Гауса μ та λ можуть бути розраховані на основі вибірки даних методом максимальної правдоподібності [16, с. 265] за допомогою наступних формул:

$$\hat{\mu} = \sum_{i=1}^{n} x_i / n,\ \hat{\lambda} = n / \sum_{i=1}^{n} \left(\frac{1}{x_i} - \frac{1}{\hat{\mu}} \right), \quad (4)$$

де x_i – значення витраченого часу на i-ту задачу, n – загальна кількість задач.

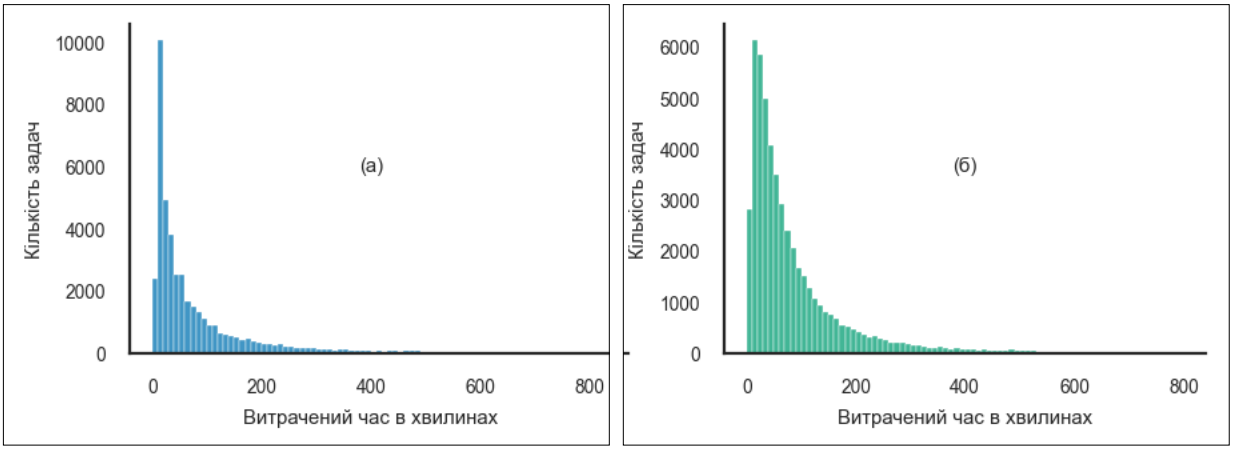

Рис. 2. Гістограми розподілу витраченого часу для (а) фактичних історичних даних, (б) згенерованих випадкових значень

В таблиці 2 наведено розраховані параметри та властивості зворотного розподілу Гауса на основі фактичних даних вибірки з точністю до 4 десяткових знаків.

Таблиця 2

Параметри та характеристики зворотного розподілу Гауса

μ	λ	Середнє	Медіана	Мода	Дисперсія
112.8197	49.1561	113.7829	55.1802	14.8830	29213.0746

Для генерації випадкових чисел, що відповідають зворотному розподілу Гауса, можна скористатись наступним алгоритмом [17, с. 89]:

$$x = \mu + \frac{\mu^2 Z^2}{2\lambda} - \frac{\mu}{2\lambda}\sqrt{4\mu\lambda Z^2 + \mu^2 Z^4},$$

$$R = \begin{cases} x, якщо U \le \dfrac{\mu}{\mu + x} \\ \dfrac{\mu^2}{x}, якщо U > \dfrac{\mu}{\mu + x} \end{cases}, \qquad (5)$$

де μ та λ – параметри розподілу, Z – випадкова величина, задана стандартним нормальним розподілом, що може бути обчислена за формулою 3, U – випадкова величина, рівномірно розподілена на інтервалі [0, 1], R – випадкова величина за зворотним розподілом Гауса.

На рисунку 3 зображені функції густини імовірності логнормального та зворотного розподілу Гауса, накладені на гістограму фактичних даних.

Як видно з рисунку 3, зворотний розподіл Гауса краще повторює форму фактичного розподілу даних, ніж логнормальний, за виключенням аномально високої частоти найпоширенішого значення витраченого часу на одну задачу згідно фактичних даних.

Під час аналізу значень витраченого часу, що найчастіше зустрічаються у поточній вибірці задач із набору історичних даних, було виявлено, що найчастіше значення витраченого часу дорівнює 20 хвилинам, а більше 99 % значень є кратними 10. В таблиці 3 відображені 10 найчастіших значень витрат часу із відповідною кількістю задач із вибірки.

Виходячи з виявленого факту кратності складності задач та властивості неперервності зворотного розподілу Гауса, алгоритм генерації складності можна адаптувати шляхом округлення всіх вихідних значень до найближчого цілого числа, кратного десяти, що є більшим за вхідне дійсне число. Округлення вгору до більшого числа є необхідним, щоб виключити можливість отримання на виході нульової складності задачі та не спотворити розподіл імовірностей.

Враховуючи аномально високу частоту значення витраченого часу у 20 хвилин, що не підпадає під зворотний закон розподілу Гауса, можна скоригувати алгоритм генерації складності задач при моделюванні процесу розробки програмного забезпечення для компенсації даної аномалії. Відповідно до вибірки даних, імовірність найпоширенішого значення дорівнює 23,685 %. Шляхом інтегрування функції розподілу імовірностей зворотного розподілу Гауса з використанням обчислених параметрів на інтервалі (10, 20] було отримано сумарну імовірність даного діапазону

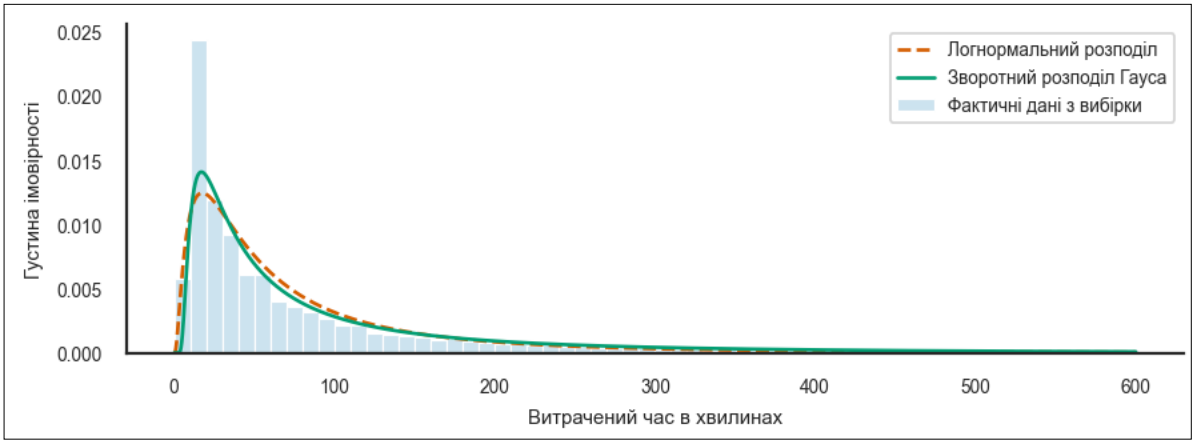

Рис. 3. Відповідність різних розподілів імовірностей фактичному розподілу даних

Таблиця 3

Найчастіші значення витрат часу

Витрати часу (хв)	20	30	40	50	60	10	70	80	90	100
Кількість задач	9974	4913	3792	2494	2492	2304	1639	1486	1295	1085

значень 13,2938 %. Таким чином, реальна імовірність даного значення на 10,3912 % більша за імовірність згідно з отриманим розподілом, проте компенсаційну імовірність необхідно обчислити із врахуванням загальної частоти найпоширенішого значення. Отже, для компенсації аномально високої імовірності найпоширенішого значення в 20 хвилин можна застосувати наступні формули:

$$P = \frac{P_D - P_G}{1 - P_G} = 11{,}9844\%,$$

$$E = \begin{cases} 20, якщо\, U < P \\ R, якщо\, U \geq P \end{cases}, \qquad (6)$$

де P_D – імовірність найпоширенішого значення згідно фактичних даних, P_G – імовірність найпоширенішого значення згідно отриманого зворотного розподілу Гауса, U – випадкова величина, рівномірно розподілена на інтервалі [0, 1), P – компенсаційна імовірність для коригування розподілу, R – випадкова величина, розподілена за отриманий зворотним розподілом Гауса, що обраховується за формулою 5, E – випадкова величина згідно адаптованого розподілу складності задач. Гістограма адаптованого розподілу зображена на рисунку 4.

Ще однією особливістю отриманого розподілу, яка заважає моделюванню процесу розробки програмного забезпечення при генерації потоку вхідних задач є те, що графік його функції не обмежений в правій частині. Це може призвести до генерації надмірно великих значень складності задач розробки, що в свою чергу може призвести до спотворення результатів симуляцій, тому доцільно обмежувати максимальне значення складності задач шляхом повторної генерації при виході згенерованого значення за граничну верхню межу. Найбільше значення витраченого часу на одну задачу у поточній вибірці даних дорівнює 1200 годин, що еквівалентно 150 людино-дням при 8-годинному робочому дні. Задачі такого розміру дійсно зустрічаються на практиці, проте при моделюванні процесу розробки краще задавати граничне значення складності задач в якості параметру моделі, щоб час, необхідний для виконання задачі, не перевищував загальну тривалість періоду часу, що симулюється.

Висновки. У даному дослідженні було проаналізовано розподіл складності задач розробки програмного забезпечення шляхом аналізу історичних даних проєктів з відкритим вихідним кодом, що входять до фонду “Apache Software Foundation”. Було спростовано гіпотезу щодо відповідності історичних даних логнормальному розподілу імовірностей та запропоновано альтернативу у вигляді зворотного розподілу Гауса, що краще описує розподіл витраченого часу на задачі з набору історичних даних, за виключенням аномально високої фактичної імовірності найпоширенішого значення в 20 хвилин. Для компенсації такої аномалії було запропоновано адаптацію алгоритму імовірнісного розподілу та обчислено відповідні числові параметри, а також запропоновано алгоритм обмеження максимального значення імовірнісного розподілу без спотворення його форми. Отриманий алгоритм розподілу імовірностей для генерації потоку вхідних задач різної складності є наближеним до фактичного розподілу витраченого часу з історичних даних та може бути використаний в низькорівневих дискретно-подійному чи агентно-орієнтованому моделюванні процесу розробки програмного забезпечення для подальшого його дослідження.

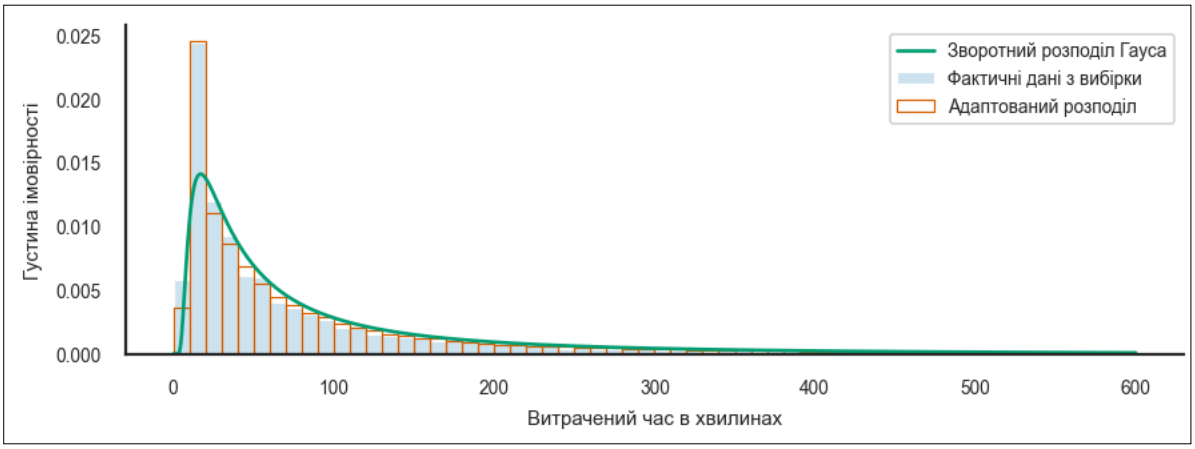

Рис. 4. Порівняння адаптованого розподілу імовірностей з фактичними даними

Список літератури:

1. Bloch, M., Blumberg, S., Laaro, J. Delivering Large-Scale IT Projects on Time, on Budget, and on Value. *Harvard Business Review*. 2012. № 5. P. 2–7.

2. Шматковська Т., Коробчук Т. Сучасні інформаційні та комунікаційні технології в моделюванні бізнес-процесів. *Економічний форум*. 2023. № 1 (3). С. 156–161. DOI: 10.36910/6775-2308-8559-2023-3-20

3. Saravanos, A., Curinga, M.X. Simulating the Software Development Lifecycle: The Waterfall Model. *Applied System Innovation*. 2023. № 6. P. 108. DOI: 10.3390/asi6060108

4. J.A. García-García, J.G. Enríquez, M. Ruiz, C. Arévalo, A. Jiménez-Ramírez Software Process Simulation Modeling: Systematic literature review. *Computer Standards & Interfaces*. 2020. Volume 70. P. 103425. DOI: 10.1016/j.csi.2020.103425

5. Кордунова Ю.С., Фелтіновскі М., Придатко О.В., Смотр О.О. Математичне моделювання процесу розробки спеціалізованих програмних систем безпеко-орієнтованого спрямування. *Вісник Львівського державного університету безпеки життєдіяльності*. 2023. № 27. С. 23–31. DOI: 10.32447/20784643.27.2023.03

6. Приходько, С. Б., Приходько, Н. В., & Книрік, К. О. Математичне моделювання трудомісткості розробки мобільних застосунків у фазі планування із врахуванням викидів. *Комп'ютерне моделювання та оптимізація складних систем* : матеріали V Міжнародної науково-технічної конференції. м. Дніпро, 6–8 листопада 2019 р. / Український державний хіміко-технологічний університет. Дніпро, 2019. С. 50.

7. Ярош, І.В, Павловський Є.В., Назарова І.А. Математичне моделювання процесу прогнозування витрат часу на розв'язання типової задачі з розробки програмного забезпечення. *Наукові праці ДонНТУ. Серія: Інформатика, кібернетика та обчислювальна техніка.* 2023. № 35-36. С. 79–84.

8. An Jen Chiang & Angus Jeang. Stochastic project management via computer simulation and optimisation method. *International Journal of Systems Science: Operations & Logistics*. 2015. Vol 2. Issue 4. P. 211–230. DOI: 10.1080/23302674.2015.1025889

9. Bochard, P. O., & Schwarz, T. Evaluation of mean and variance approximations in three point estimation of task completion times using the beta and the Kumaraswamy distribution. *International Journal of Information Technology and Management*. 2019. Vol. 18. Issue 4. P. 389–406. Inderscience Publishers. DOI: 10.1504/ijitm.2019.103053

10. Москалик Д.О., Антонюк Д.С. Аналіз розподілу складності задач при розробці програмного забезпечення з відкритим вихідним кодом. *Сучасні комп'ютерні системи та мережі в управлінні* : матеріали VI Всеукраїнської наук.-практ. Інтернет-конф. здобувачів вищої освіти та молодих вчених, м. Хмельницький, м. Херсон, 30 листопада 2023 р. / Херсонський національний технічний університет. Херсон, 2023. С. 31–32.

11. M. I. Lunesu, R. Tonelli, L. Marchesi and M. Marchesi. Assessing the Risk of Software Development in Agile Methodologies Using Simulation. *IEEE Access*. 2021. Vol. 9. P. 134240–134258. DOI: 10.1109/ACCESS.2021.3115941

12. Diamantopoulos Themistoklis, Dimitrios-Nikitas Nastosand Symeonidis Andreas. Apache Jira Issue Tracking Dataset. Zenodo. Mar. 17, 2023. DOI: 10.5281/zenodo.7740379

13. D. E. Kline, D. A. Bender. MAXIMUM LIKELIHOOD ESTIMATION FOR SHIFTED WEIBULL AND LOGNORMAL DISTRIBUTIONS. *Transactions of the ASAE*. 1990. № 33 (1). P. 330–335. DOI: 10.13031/2013.31334

14. Knop, R. Remark on algorithm 334 [G5]: normal random deviates. *Communications of the ACM*. 1969. № 12 (5). P. 281. DOI: 10.1145/362946.362996

15. Giner, G. and Smyth, G., K. Statmod: Probability calculations for the inverse gaussian distribution. *The R Journal*. 2016. № 8 (1). P. 339–351. DOI: 10.32614/rj-2016-024

16. J. L. Folks, R. S. Chhikara. The Inverse Gaussian Distribution and its Statistical Application – A Review. *Journal of the Royal Statistical Society: Series B (Methodological)*. 1978. Volume 40, Issue 3. P. 263–275. DOI: 10.1111/j.2517-6161.1978.tb01039.x

17. Michael, J. R., Schucany, W. R., Haas, R. W. Generating Random Variates Using Transformations with Multiple Roots. *The American Statistician*. 1976. № 30 (2). P. 88–90. DOI: 10.1080/00031305.1976.10479147

Moskalyk D.O., Antoniuk D.S. PROBABILITY DISTRIBUTION OF INPUT TASKS COMPLEXITY FOR DISCRETE-EVENT AND AGENT-BASED SOFTWARE DEVELOPMENT PROCESS MODELING

This paper explores the probability distribution of the software development tasks complexity by analyzing a set of historical data on the development of open-source software projects included in the Apache Software Foundation. Based on the histogram of the probability distribution of the time spent on the task of correcting defects, improving existing functionality, and creating new features, the hypothesis that the actual probability

distribution conforms to the form of a lognormal distribution was checked and rejected. Instead, the inverse Gaussian distribution was proposed, proved that it better reflects the real probability distribution and its numerical parameters have been calculated. For practical application in low-level modeling of the software development process, an algorithm for generating random values from the given distribution has been provided. During a comparison of the inverse Gaussian distribution probability density function with the histogram of the actual time spent, an anomalous graphs divergence was noticed at the point of the most common value equal to 20 minutes. To compensate this anomaly, an adaptation of the algorithm for probabilistic generation of the software development tasks complexity has been developed, which preserves the original shape of the inverse Gaussian distribution and replicates the abnormally high frequency of the most common value. Additionally, during an analysis of the historical data, it was encountered that more than 99 % of the values of the time spent on one task are multiples of 10, which allows modifying the current algorithm for generating discrete values by rounding the generated values up to the nearest multiple of 10, which eliminates the possibility of generating zero values and allows to optimize calculations when using this algorithm in the corresponding simulations. Since the inverse Gaussian distribution is not limited at the right side, to prevent distortion of the simulation results, it is advisable to discard all generated values greater than a certain maximum threshold, which should be determined according to the requirements of the corresponding simulations. In this way, the general form of the probability distribution is preserved, and the distribution itself is limited to a finite range of values.

Key words: *modeling, simulation, task complexity, software development, inverse Gaussian distribution, lognormal distribution, algorithm.*